US010826502B1

(12) **United States Patent**
Badizadegan

(10) **Patent No.: US 10,826,502 B1**
(45) **Date of Patent: Nov. 3, 2020**

(54) **FIELD PROGRAMMABLE GATE ARRAY WITH EXTERNAL PHASE-LOCKED LOOP**

(71) Applicant: **Nima Badizadegan**, New York, NY (US)

(72) Inventor: **Nima Badizadegan**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **16/672,076**

(22) Filed: **Nov. 1, 2019**

Related U.S. Application Data

(60) Provisional application No. 62/803,017, filed on Feb. 8, 2019, provisional application No. 62/755,804, filed on Nov. 5, 2018.

(51) **Int. Cl.**

H03L 7/085	(2006.01)
G06F 1/08	(2006.01)
G06F 30/34	(2020.01)
H04J 3/06	(2006.01)

(52) **U.S. Cl.**
CPC ***H03L 7/085*** (2013.01); ***G06F 1/08*** (2013.01); ***G06F 30/34*** (2020.01); ***H04J 3/0658*** (2013.01); *H03L 2207/50* (2013.01)

(58) **Field of Classification Search**
CPC H03L 7/085; H03L 7/087; H03L 7/089; H03L 7/0891; H03L 7/0893; H03L 7/0895; H03L 7/0896; H03L 7/0898; H03L 7/091; H03L 7/093; H03L 7/095; H03L 2207/50; G06F 30/30; G06F 30/34; G06F 30/343; G06F 30/347; G06F 1/04; G06F 1/08; H04J 3/0658
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,068,188	A	1/1978	Yokoyama	
4,745,372	A	5/1988	Miwa	
6,675,306	B1	1/2004	Baxter	
7,080,345	B1 *	7/2006	Iotov	G06F 30/30 716/113
7,333,570	B2 *	2/2008	Aung	G11C 7/22 375/326
8,467,418	B2	6/2013	Aweya et al.	
9,001,951	B1	4/2015	Kumpulainen	
9,330,740	B1	5/2016	Baeckler et al.	
10,037,568	B2	7/2018	Taylor et al.	
10,169,814	B2	1/2019	Parsons et al.	
10,411,720	B2	9/2019	Lenzen et al.	
2019/0025413	A1	1/2019	Liu et al.	

OTHER PUBLICATIONS

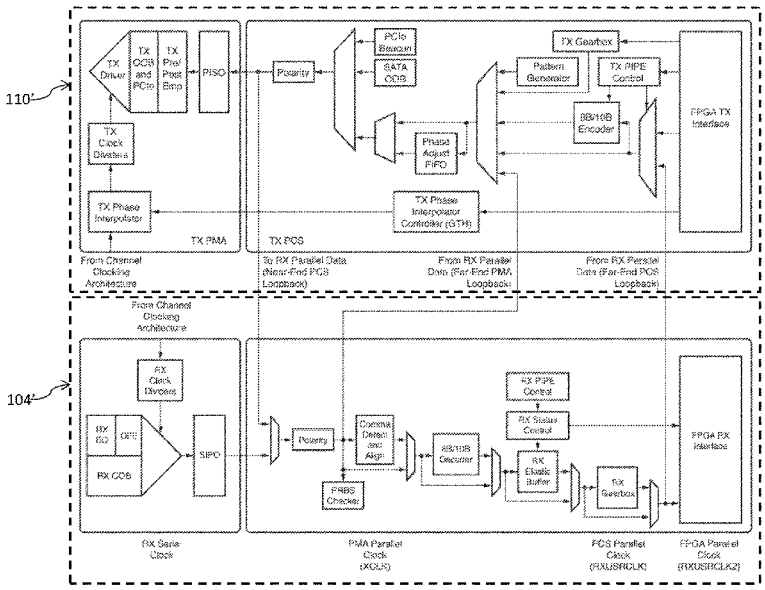
Xilinx, "7 Series FPGAs GTX/GTH Transceivers—User Guide," p. 26, available at https://www.xilinx.com/support/documentation/user_guides/ug476_7Series_Transceivers.pdf (Aug. 14, 2018).
Sarah Harris, "Timing & Synchronization," Harvey Mudd College Engineering Department, p. 24, available at https://cseweb.ucsd.edu/classes/wi06/cse291-b/slide/let4/timingint.pdf (Jan. 31, 2006).
(Continued)

Primary Examiner — Diana J. Cheng
(74) *Attorney, Agent, or Firm* — Amster, Rothstein & Ebenstein LLP

(57) **ABSTRACT**

The present invention relates to a field programmable gate array system that provides phase control with minimal latency.

29 Claims, 26 Drawing Sheets



(56) **References Cited**

OTHER PUBLICATIONS

Sam Palermo, "ECEN 720 High-Speed Links: Circuits and Systems—Lab 5," Texas A&M University, p. 5, available at: http://ece.tamu.edu/~spalermo/ecen689/ECEN720_lab5_2017.pdf (2017).

Altera, "Stratix V Device Handbook," vol. 2: Transceivers, Intel, p. 13, available at: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/stratix-v/stx5_xcvr.pdf (Feb. 15, 2017).

Stackexchange, "Phase Frequency Detector in PLL," StackExchange, available at: https://electronics.stackexchange.com/questions/301402/phase-frequency-detector-in-pll (Apr. 24, 2017).

IEEE Standards Association, "IEEE Standards for Ethernet," IEEE 802.3-2018, Section 4, Clauses 49 and 51, Section 6, Clauses 82 and 83, available at: https://standards.ieee.org/standard/802_3-2018.html (Aug. 31, 2018).

Slobodan Milijevic, "The basics of synchronized Ethernet," EE Times-Asia, available at: https://www.microsemi.com/document-portal/doc_view/126475-the-basics-of-synchronized-ethernet-synce.

Maciej Lipinski, et al., "White Rabbit: a PTP Application for Robust Sub-nanosecond Synchronization," 2011 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, Munich, pp. 25-30, available at: http://white-rabbit.web.cem.ch/documents/White_Rabbit-a_PTP_application_for_robust_sub-nanosecond_synchronization.pdf (2011).

Floyd M. Gardner, "Phaselock Techniques," Third Edition, Wiley-Interscience, Chapters 1, 4, 9, 10, 11, 13, and 17 (2005).

Jens Twiefel, et al., "Digital signal processing for an adaptive phase-locked loop controller," Proceedings of SPIE—The International Society for Optical Engineering, vol. 6926 (Apr. 2008).

Chen Yao, "Time to Digital Converter used in ALL digital PLL," Master of Science Thesis In System-on-Chip Design, Stockholm (Aug. 2011).

Behzad Razavi, "A Circuit for All Seasons: The StrongARM Latch," IEEE Solid-State Circuits Magazine (Spring 2015).

Ping Lu, et al., "A 2-D GRO Vernier Time-to-Digital Converter with Large Input Range and Small Latency," Lund University—EIT Department, University of Toronto, IEEE Radio Frequency Integrated Circuits Symposium (2013).

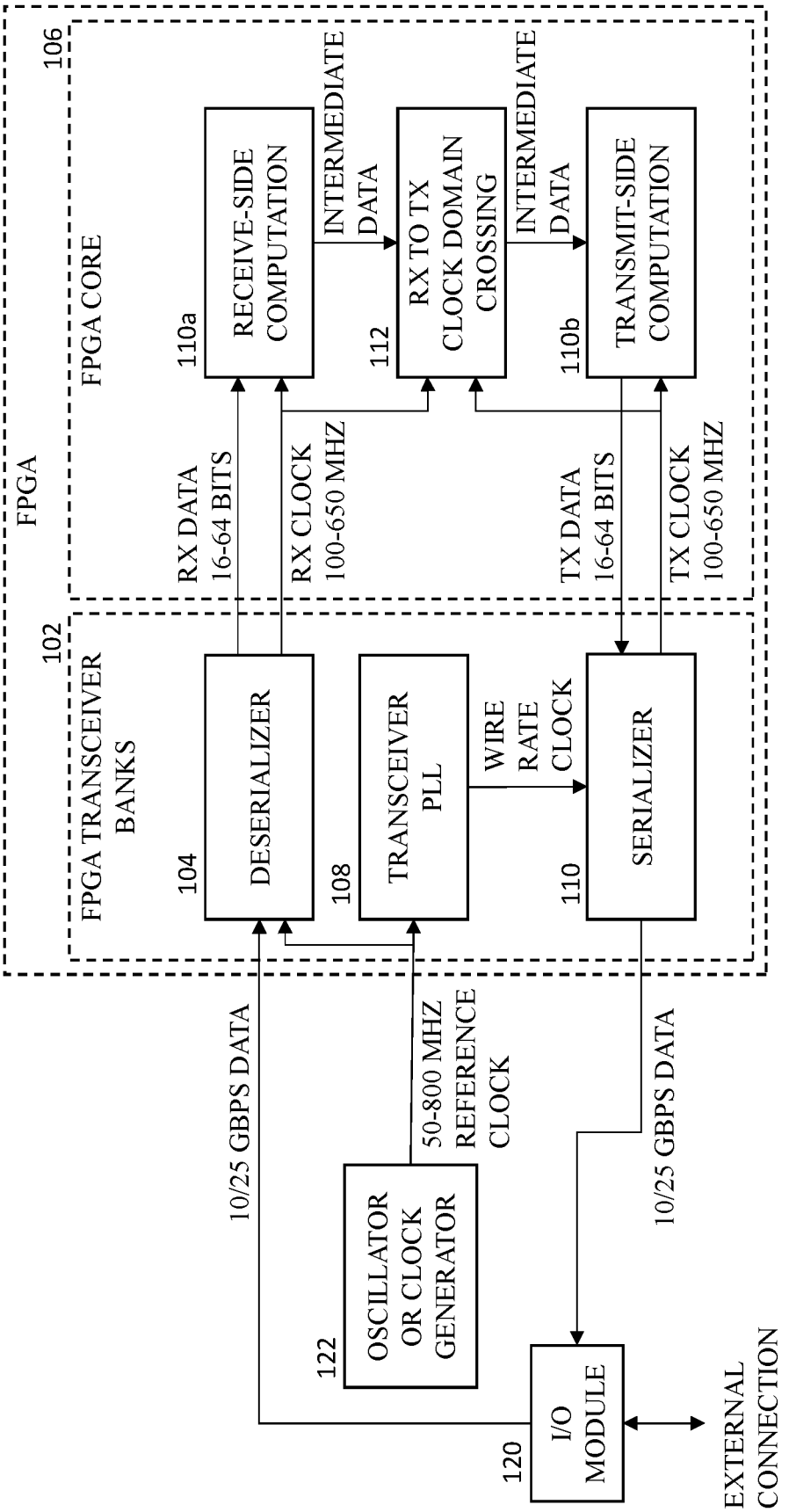
Christian Leber, et al. "High Frequency Trading Acceleration using FPGAs," University of Heidelberg, International Conference on Field Programmable Logic and Applications, Germany (2011).

International Telecommunication Union, "Timing and synchronization aspects in packet networks," ITU-T, G.8261/Y.1361, International Telecommunication Union (May 2006).

International Telecommunication Union, "Distribution of timing information through packet networks," ITU-T, G.8264/Y.1364, International Telecommunication Union (Aug. 2017).

IEEE Standards Association, "IEEE Standards for Ethernet," IEEE 802.3-2018, Section 4, Figure 52-4, p. 634 (2018).

* cited by examiner



PRIOR ART

FIG. 1

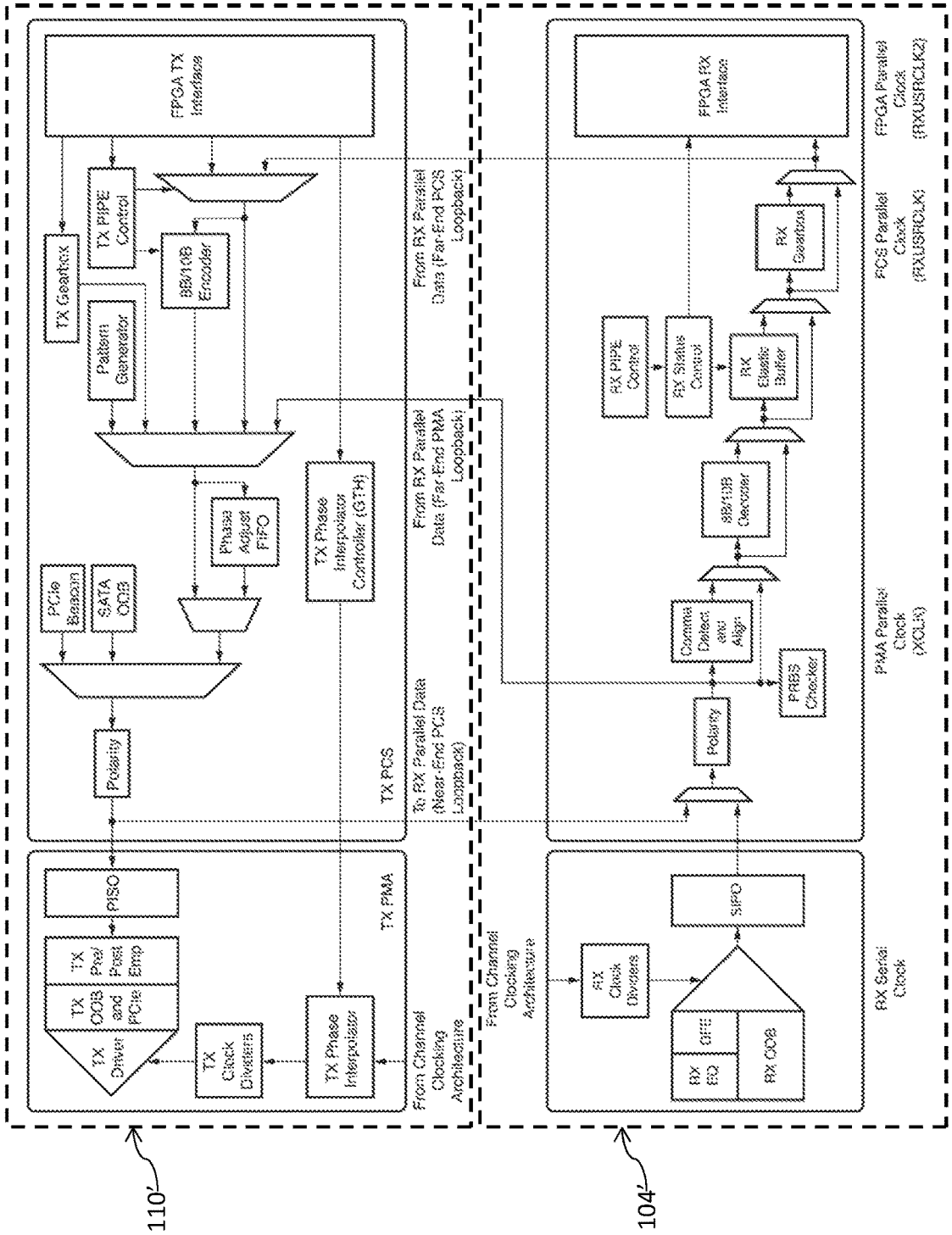

110'
104'
TX PCS
TX PMA
RX PCS
RX PMA
FPGA TX Interface
FPGA RX Interface
TX PIPE Control
TX Gearbox
Pattern Generator
8B/10B Encoder
Phase Adjust FIFO
TX Phase Interpolator Controller (GTH)
PCIe Beacon
SATA OOB
Polarity
PISO
TX Pre Post Emp
TX OOB and PCIe
TX Driver
TX Clock Dividers
TX Phase Interpolator
From Channel Clocking Architecture
To RX Parallel Data (Near-End PCS Loopback)
From RX Parallel Data (Far-End PCS Loopback)
From RX Parallel Data (Far-End PMA Loopback)
RX PIPE Control
RX Status Control
RX Elastic Buffer
RX Gearbox
8B/10B Decoder
Comma Detect and Align
PRBS Checker
SIPO
RX Clock Dividers
RX CDR
RX EQ
RX Serial Clock
PMA Parallel Clock (XCLK)
PCS Parallel Clock (RXUSRCLK)
FPGA Parallel Clock (RXUSRCLK2)

FIG. 1A

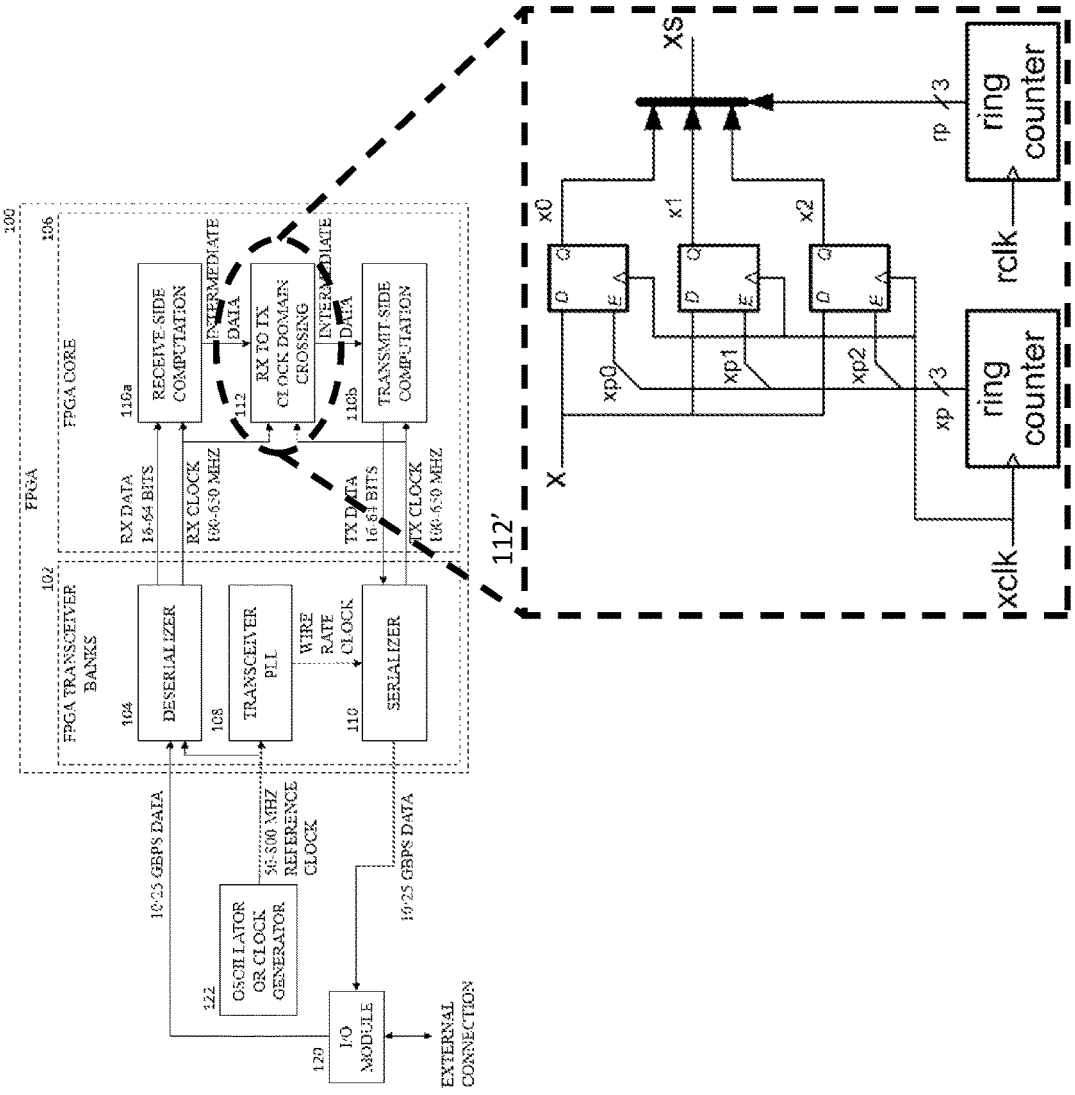

100
FPGA
102
FPGA TRANSCEIVER BANKS
104
DESERIALIZER
108
TRANSCEIVER PLL
WIRE RATE CLOCK
110
SERIALIZER
106
FPGA CORE
RX DATA 16-64 BITS
RX CLOCK (86-650 MHZ)
116a
RECEIVE SIDE COMPUTATION
INTERMEDIATE DATA
112
RX TO TX CLOCK DOMAIN CROSSING
INTERMEDIATE DATA
116b
TRANSMIT SIDE COMPUTATION
TX DATA 16-64 BITS
TX CLOCK (86-650 MHZ)
10-25 GBPS DATA
50-800 MHZ REFERENCE CLOCK
122
OSCILLATOR OR CLOCK GENERATOR
120
I/O MODULE
EXTERNAL CONNECTION
10-25 GBPS DATA
112'
x
xs
x0
x1
x2
xp0
xp1
xp2
xp
xclk
rclk
ring counter
ring counter
3
3

FIG. 1B

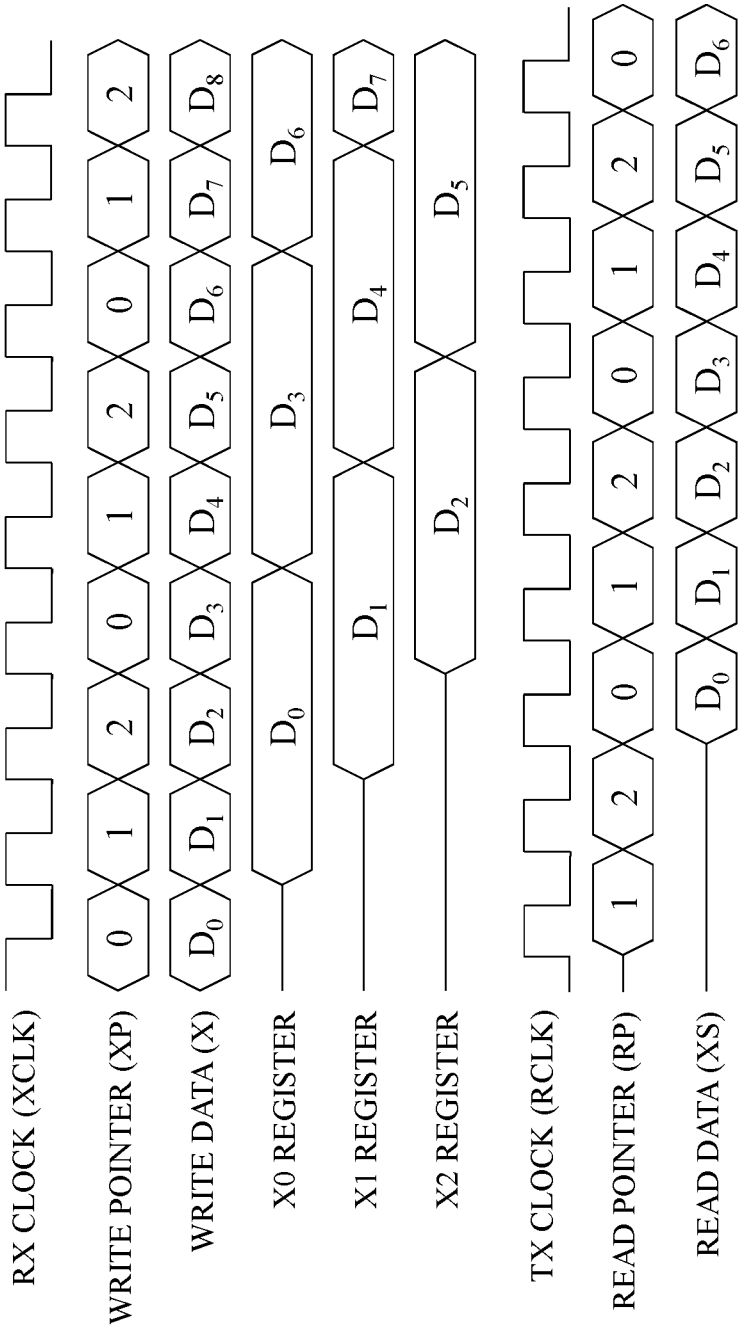

RX CLOCK (XCLK)
WRITE POINTER (XP)
0 1 2 0 1 2 0 1 2
WRITE DATA (X)
D0 D1 D2 D3 D4 D5 D6 D7 D8
X0 REGISTER
D0 D3 D6
X1 REGISTER
D1 D4 D7
X2 REGISTER
D2 D5
TX CLOCK (RCLK)
READ POINTER (RP)
1 2 0 1 2 0 1 2 0
READ DATA (XS)
D0 D1 D2 D3 D4 D5 D6

FIG. 1C

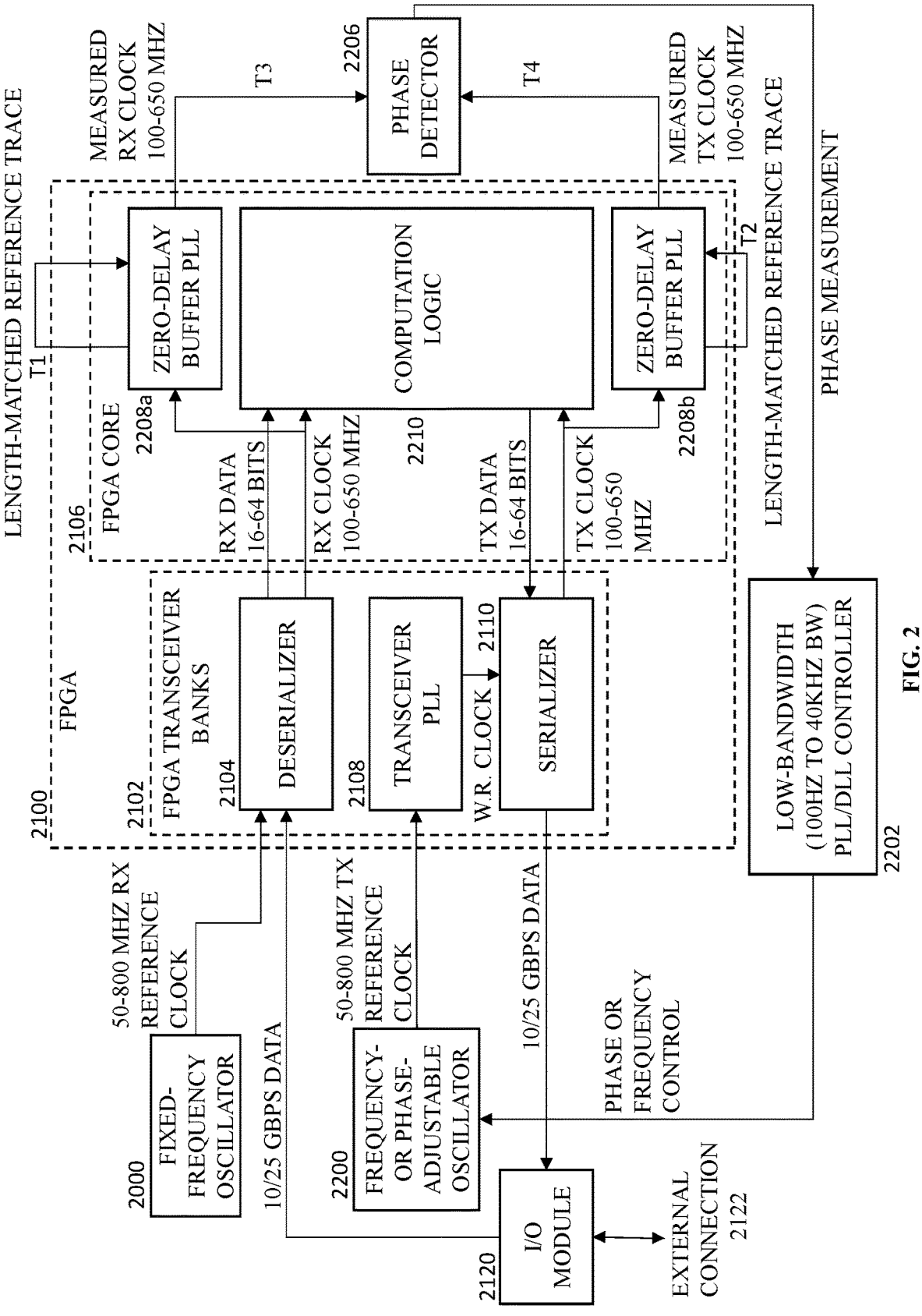

LENGTH-MATCHED REFERENCE TRACE
T1
MEASURED RX CLOCK 100-650 MHZ
T3
2206
PHASE DETECTOR
T4
MEASURED TX CLOCK 100-650 MHZ
T2
LENGTH-MATCHED REFERENCE TRACE
PHASE MEASUREMENT
2106
FPGA CORE
2208a
ZERO-DELAY BUFFER PLL
COMPUTATION LOGIC
2210
ZERO-DELAY BUFFER PLL
2208b
RX DATA 16-64 BITS
RX CLOCK 100-650 MHZ
TX DATA 16-64 BITS
TX CLOCK 100-650 MHZ
2100
FPGA
2102
FPGA TRANSCEIVER BANKS
2104
DESERIALIZER
2108
TRANSCEIVER PLL
W.R. CLOCK
2110
SERIALIZER
LOW-BANDWIDTH (100HZ TO 40KHZ BW) PLL/DLL CONTROLLER
2202
50-800 MHZ RX REFERENCE CLOCK
50-800 MHZ TX REFERENCE CLOCK
10/25 GBPS DATA
10/25 GBPS DATA
PHASE OR FREQUENCY CONTROL
2000
FIXED-FREQUENCY OSCILLATOR
2200
FREQUENCY-OR PHASE-ADJUSTABLE OSCILLATOR
2120
I/O MODULE
EXTERNAL CONNECTION 2122

FIG. 2

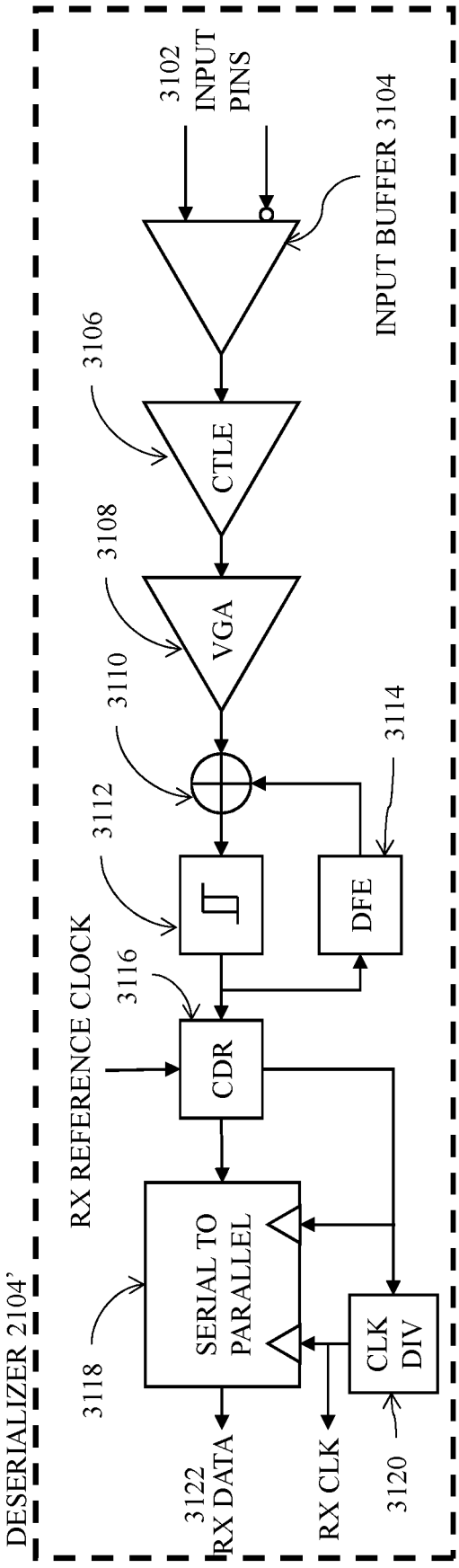

DESERIALIZER 2104'
3102 INPUT PINS
INPUT BUFFER 3104
3106 CTLE
3108 VGA
3110
3112
3114 DFE
RX REFERENCE CLOCK
3116 CDR
3118 SERIAL TO PARALLEL
CLK DIV
3120
3122 RX DATA
RX CLK

FIG. 3A

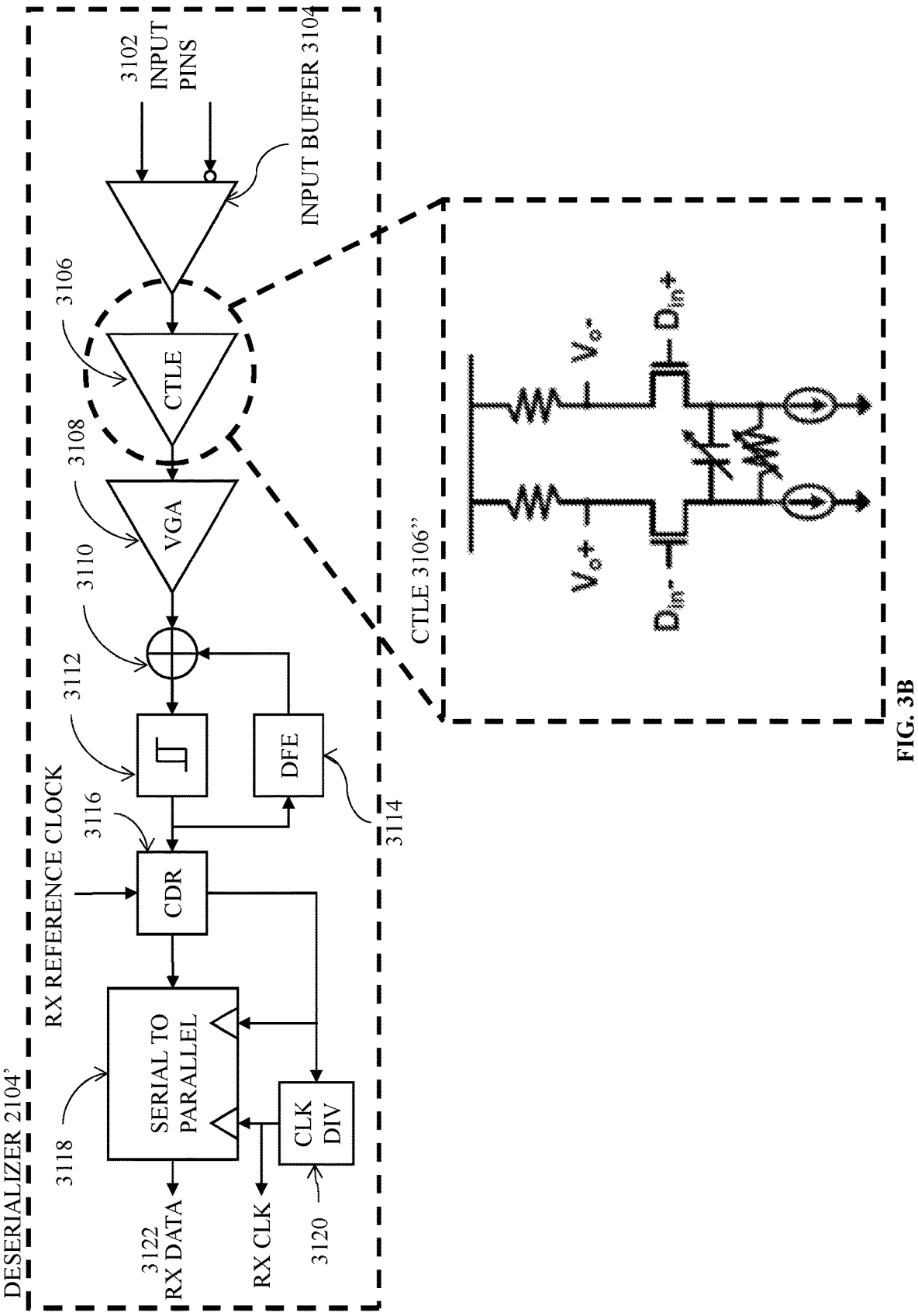

DESERIALIZER 2104'
RX REFERENCE CLOCK
3102 INPUT PINS
INPUT BUFFER 3104
3106
CTLE
3108
VGA
3110
3112
3114
DFE
3116
CDR
3118
SERIAL TO PARALLEL
CLK DIV
3120
3122 RX DATA
RX CLK
CTLE 3106"
V_o^-
D_{in}^+
V_o^+
D_{in}^-

FIG. 3B

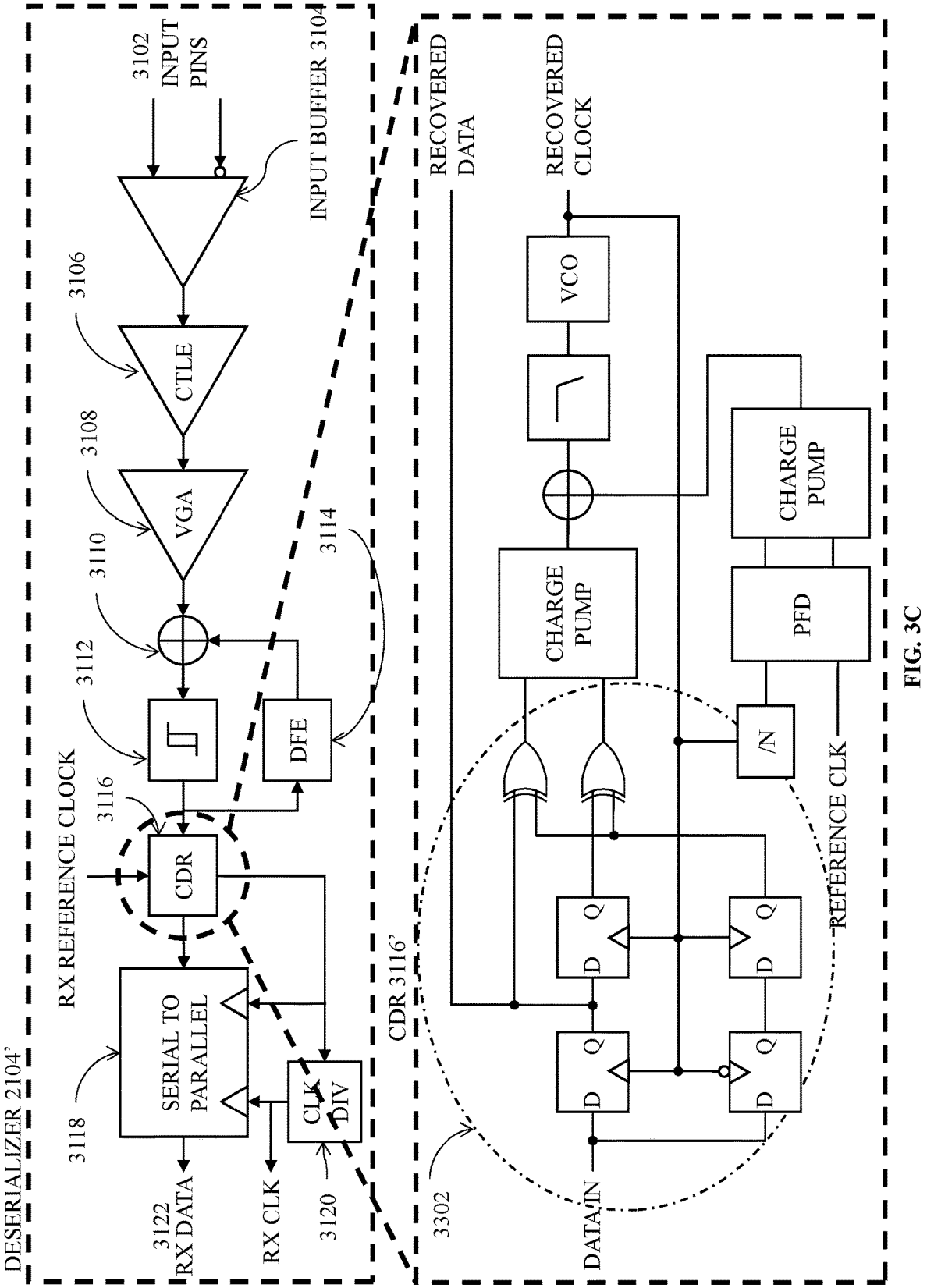

DESERIALIZER 2104'
3102
INPUT PINS
INPUT BUFFER 3104
3106
CTLE
3108
VGA
3110
3112
3114
DFE
RX REFERENCE CLOCK
3116
CDR
3118
SERIAL TO PARALLEL
CLK DIV
3120
3122
RX DATA
RX CLK
CDR 3116'
RECOVERED DATA
RECOVERED CLOCK
VCO
CHARGE PUMP
CHARGE PUMP
PFD
/N
Q
D
3302
DATA IN
REFERENCE CLK

FIG. 3C

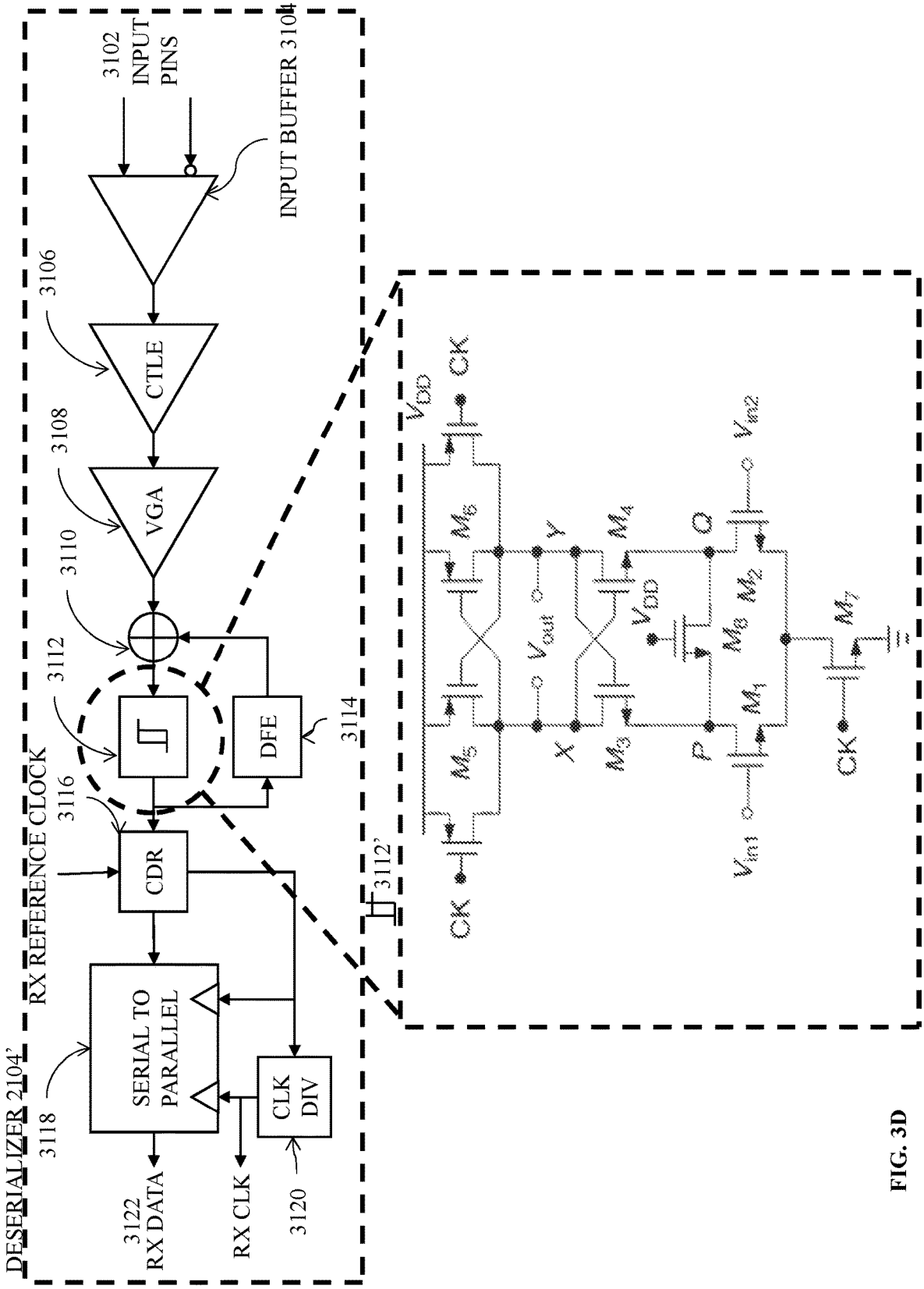

DESERIALIZER 2104'
RX REFERENCE CLOCK
3102 INPUT PINS
INPUT BUFFER 3104
3106 CTLE
3108 VGA
3110
3112
3112'
3114 DFE
3116 CDR
3118 SERIAL TO PARALLEL
CLK DIV
3120
3122 RX DATA
RX CLK
V_{DD}
CK
M_6
M_5
M_4
M_3
M_8
M_2
M_1
M_7
V_{out}
X
Y
P
Q
V_{in2}
V_{in1}

FIG. 3D

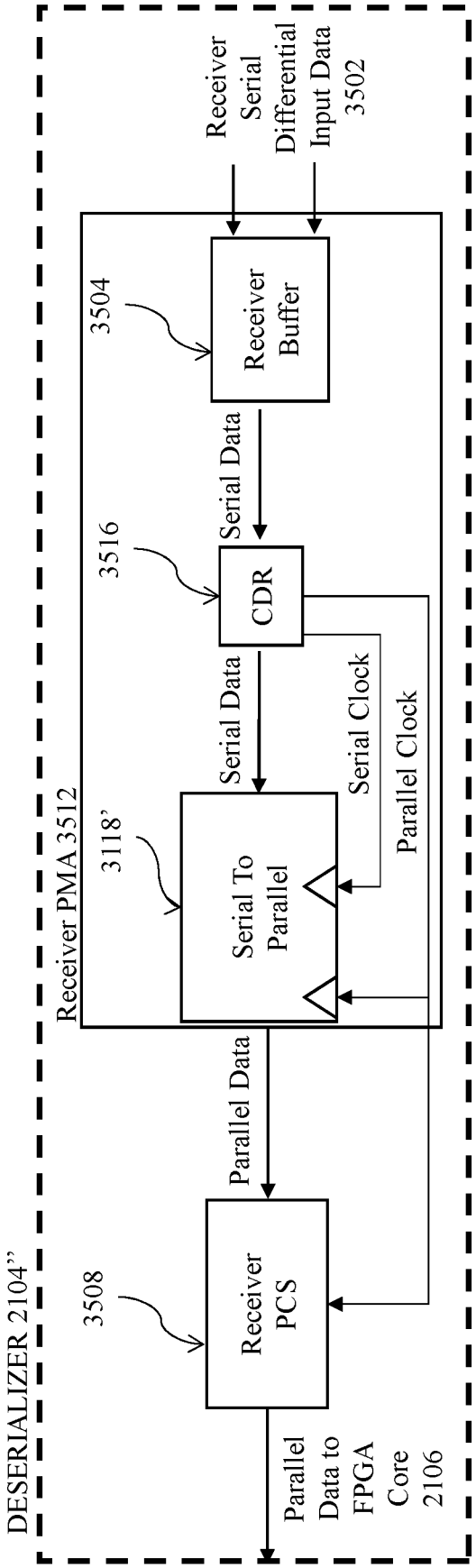

DESERIALIZER 2104"
Receiver PMA 3512
Receiver Serial Differential Input Data 3502
3504
Receiver Buffer
Serial Data
3516
CDR
Serial Data
Serial Clock
Parallel Clock
3118'
Serial To Parallel
Parallel Data
3508
Receiver PCS
Parallel Data to FPGA Core 2106

FIG. 3E

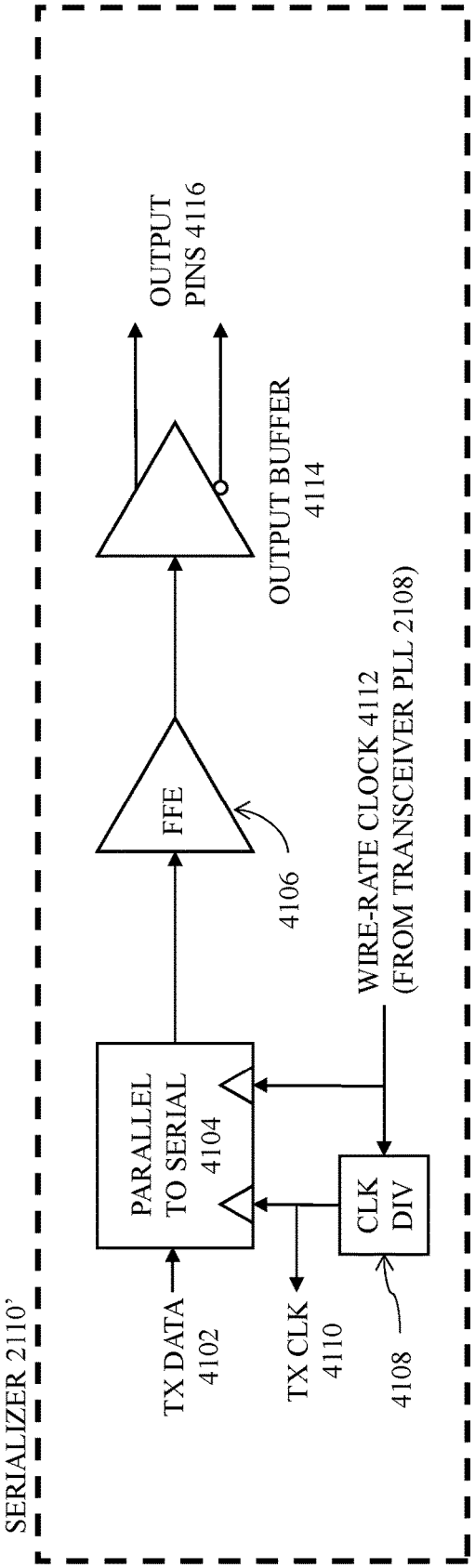

SERIALIZER 2110'
TX DATA 4102
TX CLK 4110
4108
PARALLEL TO SERIAL 4104
CLK DIV
WIRE-RATE CLOCK 4112 (FROM TRANSCEIVER PLL 2108)
FFE
4106
OUTPUT BUFFER 4114
OUTPUT PINS 4116

FIG. 4A

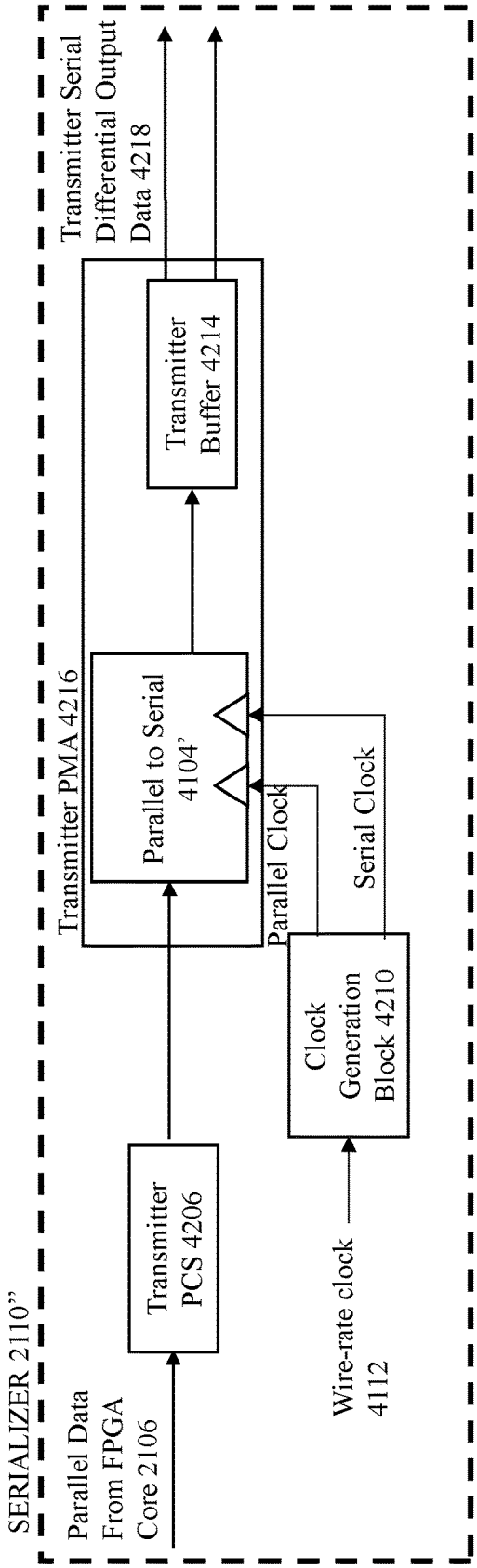

SERIALIZER 2110''
Parallel Data From FPGA Core 2106
Transmitter PCS 4206
Wire-rate clock 4112
Clock Generation Block 4210
Parallel Clock
Serial Clock
Transmitter PMA 4216
Parallel to Serial 4104'
Transmitter Buffer 4214
Transmitter Serial Differential Output Data 4218

FIG. 4B

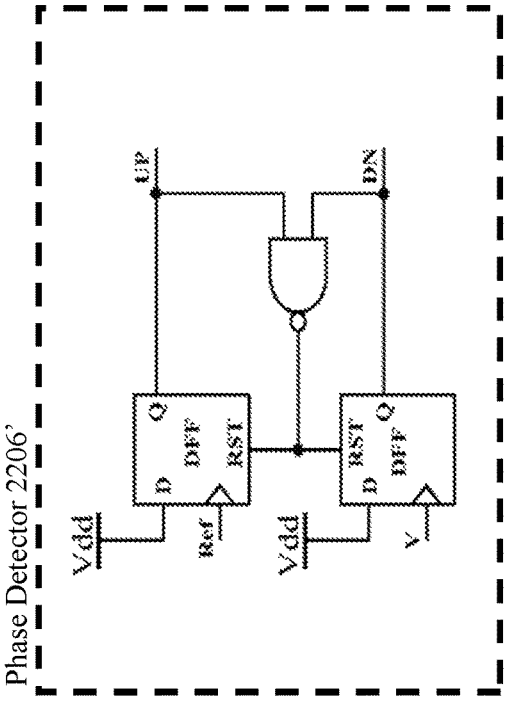

Phase Detector 2206'
UP
DN
Q
D
DFF
RST
Vdd
Ref
Vdd
V

FIG. 5A

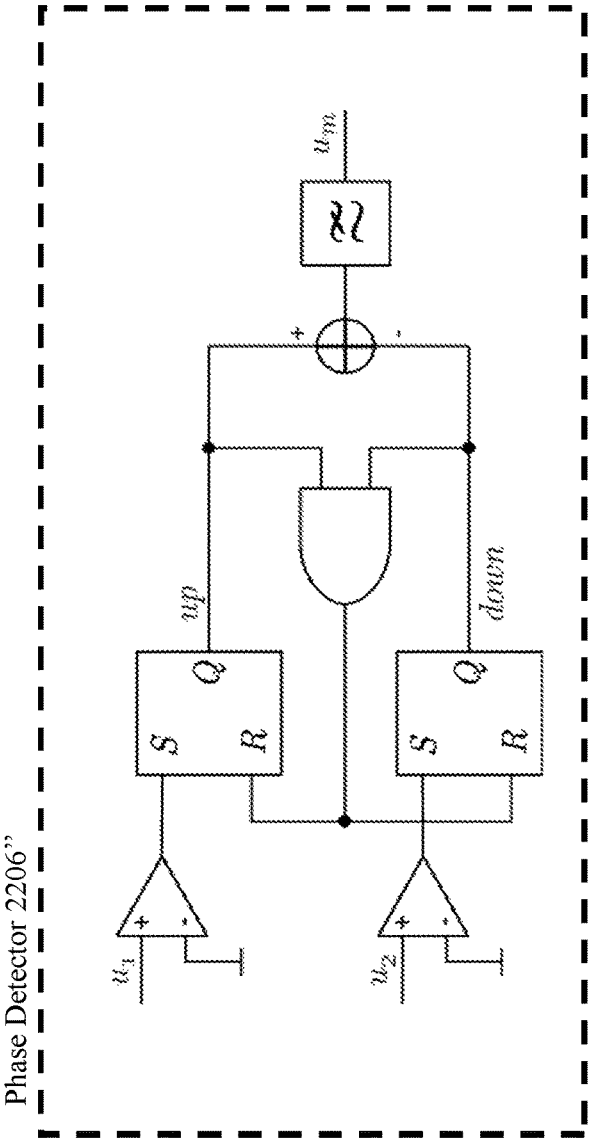

Phase Detector 2206''
u_m
$\mathcal{R}$
up
down
Q
S
R
u_1
u_2

FIG. 5B

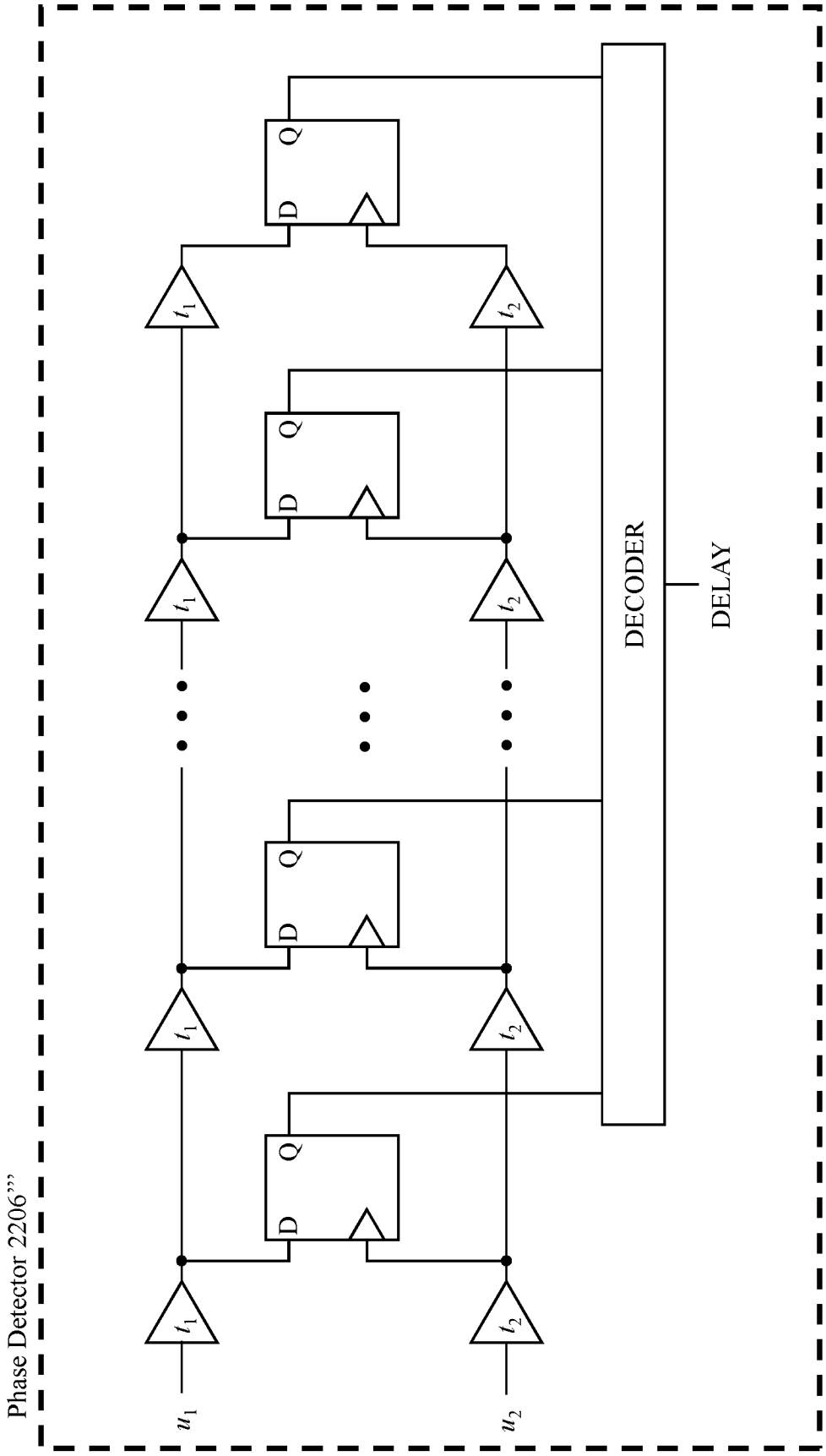


FIG. 5C

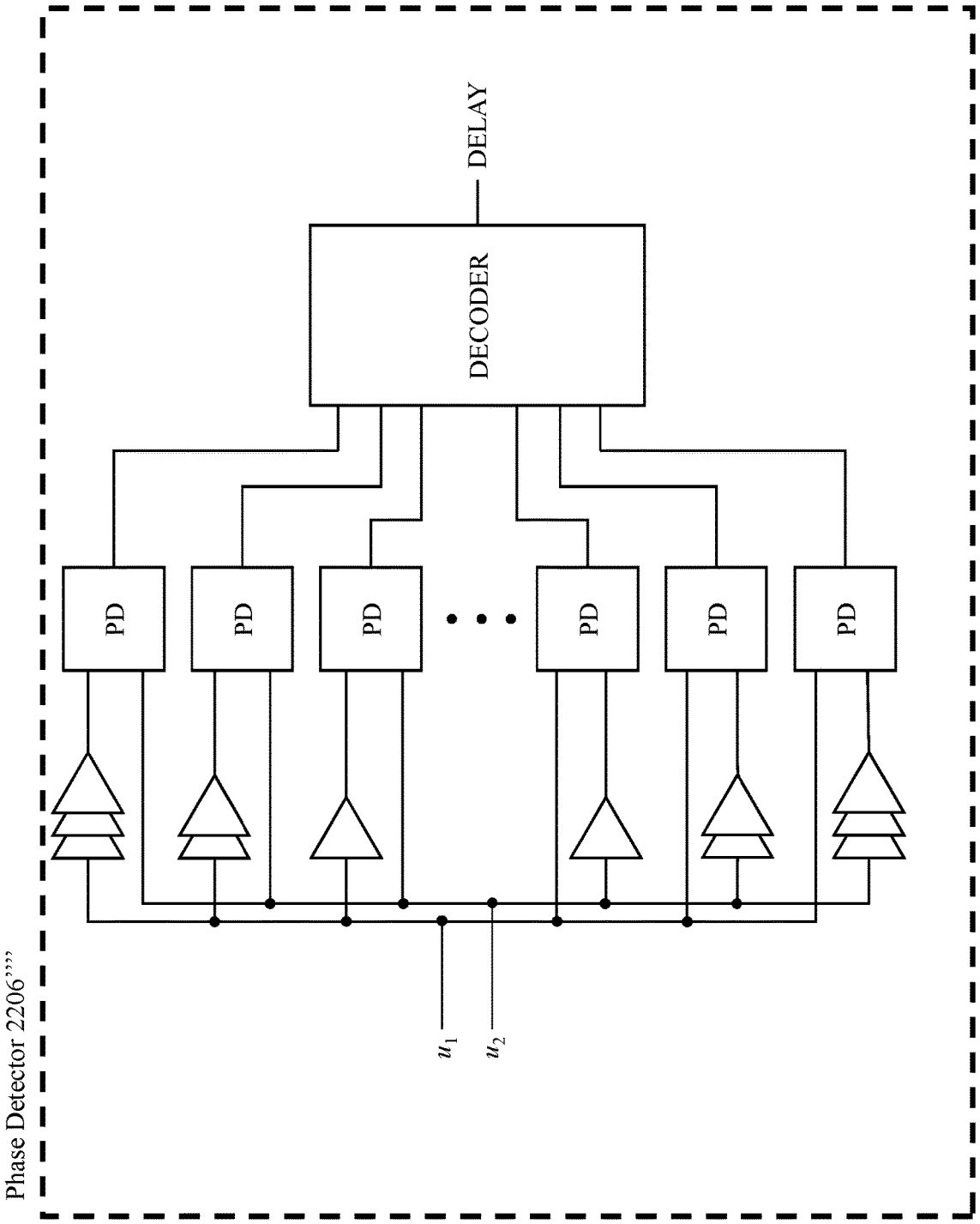


FIG. 5D

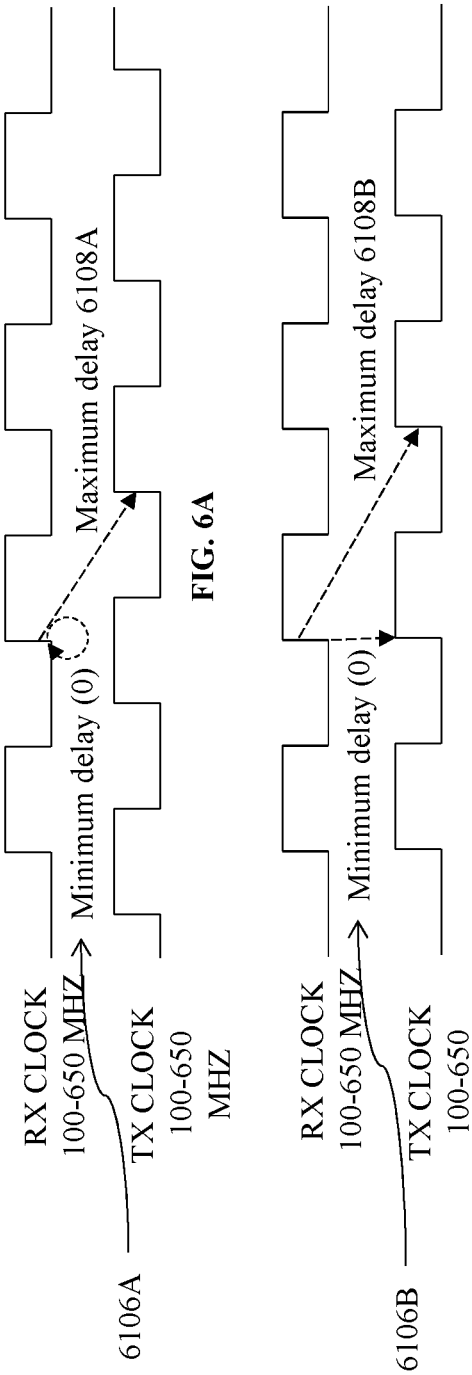


FIG. 6A

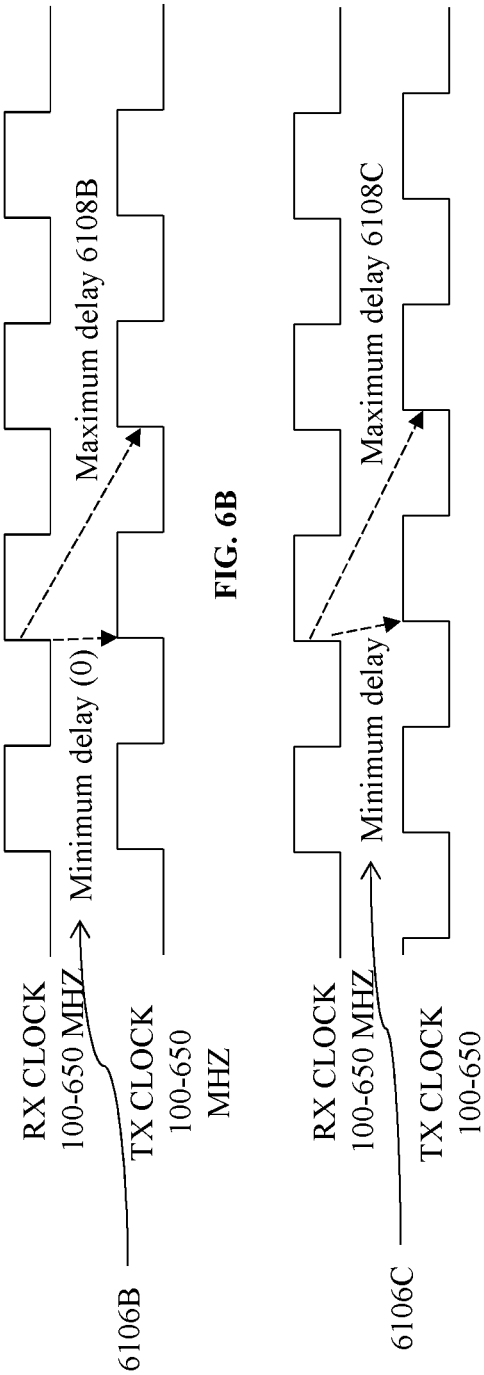


FIG. 6B

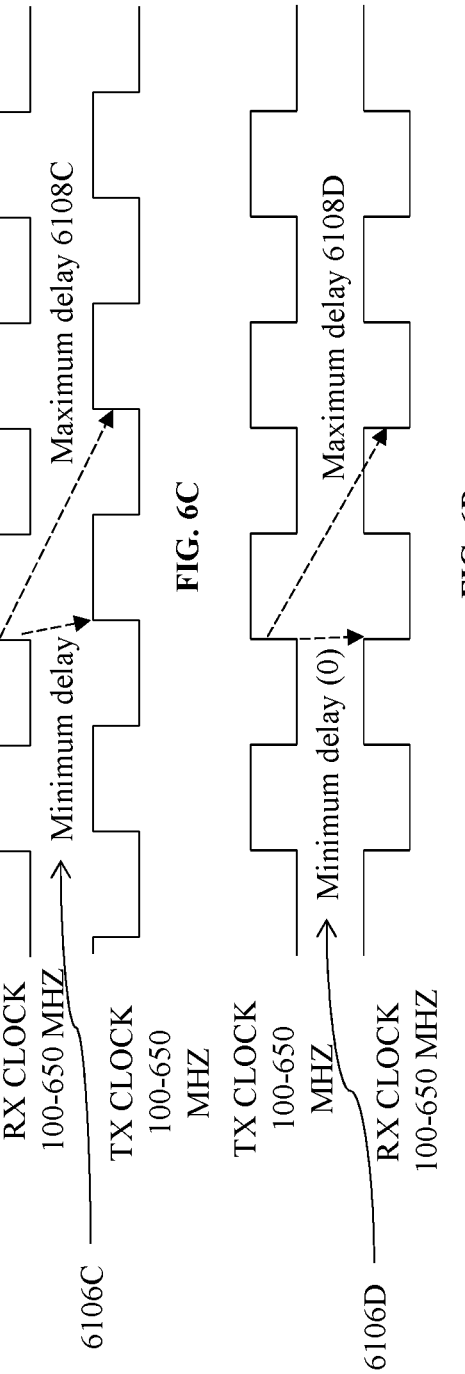


FIG. 6C

FIG. 6D

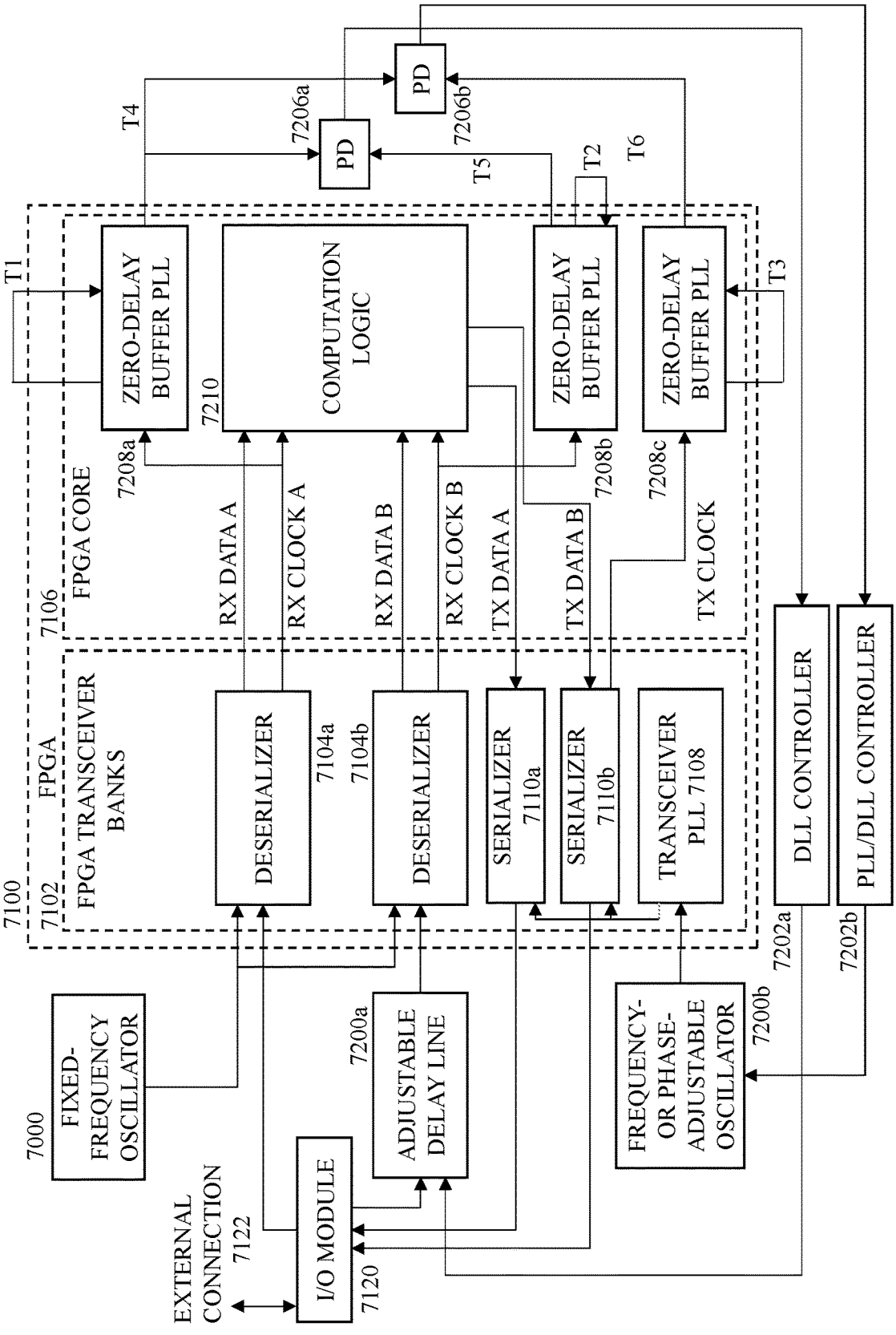

7100
FPGA
7102
FPGA TRANSCEIVER BANKS
7106
FPGA CORE
DESERIALIZER
7104a
DESERIALIZER
7104b
SERIALIZER 7110a
SERIALIZER 7110b
TRANSCEIVER PLL 7108
ZERO-DELAY BUFFER PLL
ZERO-DELAY BUFFER PLL
ZERO-DELAY BUFFER PLL
7210
COMPUTATION LOGIC
7208a
7208b
7208c
RX DATA A
RX CLOCK A
RX DATA B
RX CLOCK B
TX DATA A
TX DATA B
TX CLOCK
T1
T2
T3
T4
T5
T6
PD
PD
7206a
7206b
DLL CONTROLLER
PLL/DLL CONTROLLER
7202a
7202b
FIXED-FREQUENCY OSCILLATOR
7000
ADJUSTABLE DELAY LINE
7200a
FREQUENCY-OR PHASE-ADJUSTABLE OSCILLATOR
7200b
I/O MODULE
7120
EXTERNAL CONNECTION
7122

FIG. 7A

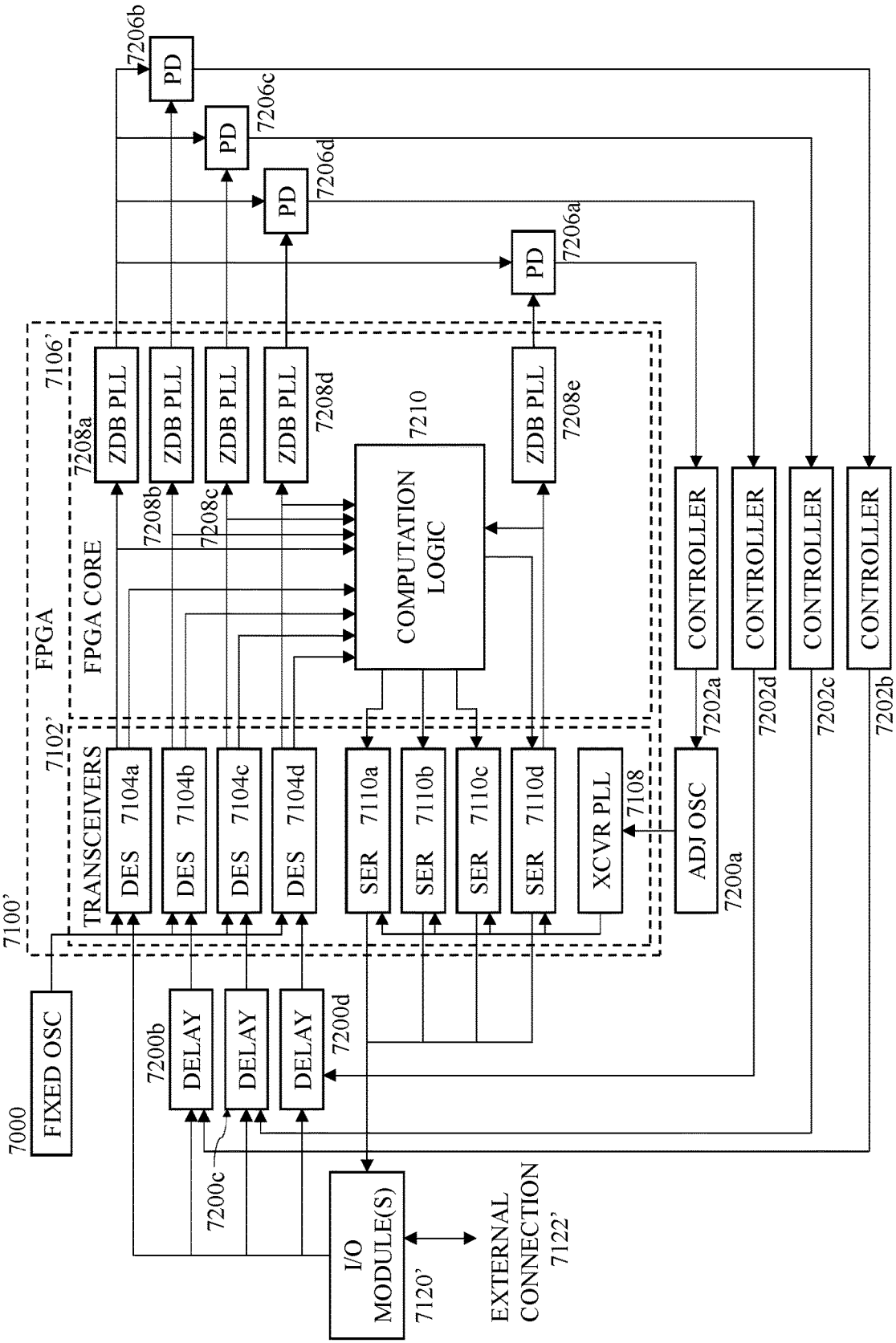

7100'
FPGA
7102'
TRANSCEIVERS
7106'
FPGA CORE
DES 7104a
DES 7104b
DES 7104c
DES 7104d
SER 7110a
SER 7110b
SER 7110c
SER 7110d
XCVR PLL
7108
ZDB PLL
7208a
ZDB PLL
7208b
ZDB PLL
7208c
ZDB PLL
7208d
ZDB PLL
7208e
COMPUTATION LOGIC
7210
PD
7206b
PD
7206c
PD
7206d
PD
7206a
CONTROLLER
CONTROLLER
CONTROLLER
CONTROLLER
7202a
7202d
7202c
7202b
ADJ OSC
7200a
FIXED OSC
7000
DELAY
DELAY
DELAY
7200b
7200d
7200c
I/O MODULE(S)
7120'
EXTERNAL CONNECTION
7122'

FIG. 7B

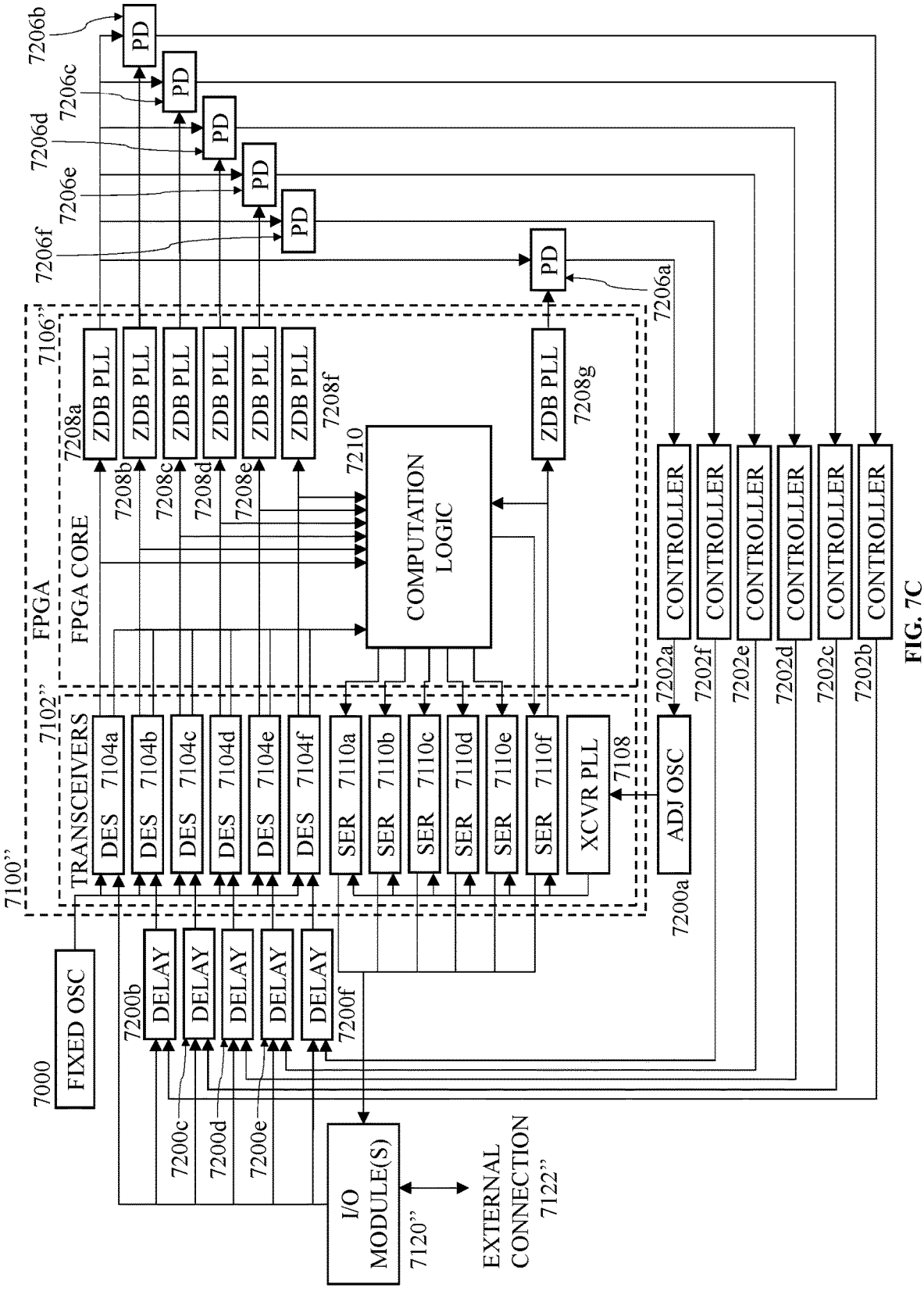

7100''
FPGA
7102''
TRANSCEIVERS
DES 7104a
DES 7104b
DES 7104c
DES 7104d
DES 7104e
DES 7104f
SER 7110a
SER 7110b
SER 7110c
SER 7110d
SER 7110e
SER 7110f
XCVR PLL 7108
7106''
FPGA CORE
ZDB PLL 7208a
ZDB PLL 7208b
ZDB PLL 7208c
ZDB PLL 7208d
ZDB PLL 7208e
ZDB PLL 7208f
ZDB PLL 7208g
COMPUTATION LOGIC 7210
PD 7206a
PD 7206b
PD 7206c
PD 7206d
PD 7206e
PD 7206f
CONTROLLER 7202a
CONTROLLER 7202f
CONTROLLER 7202e
CONTROLLER 7202d
CONTROLLER 7202c
CONTROLLER 7202b
ADJ OSC 7200a
DELAY 7200b
DELAY 7200c
DELAY 7200d
DELAY 7200e
DELAY 7200f
FIXED OSC 7000
I/O MODULE(S) 7120''
EXTERNAL CONNECTION 7122''

FIG. 7C

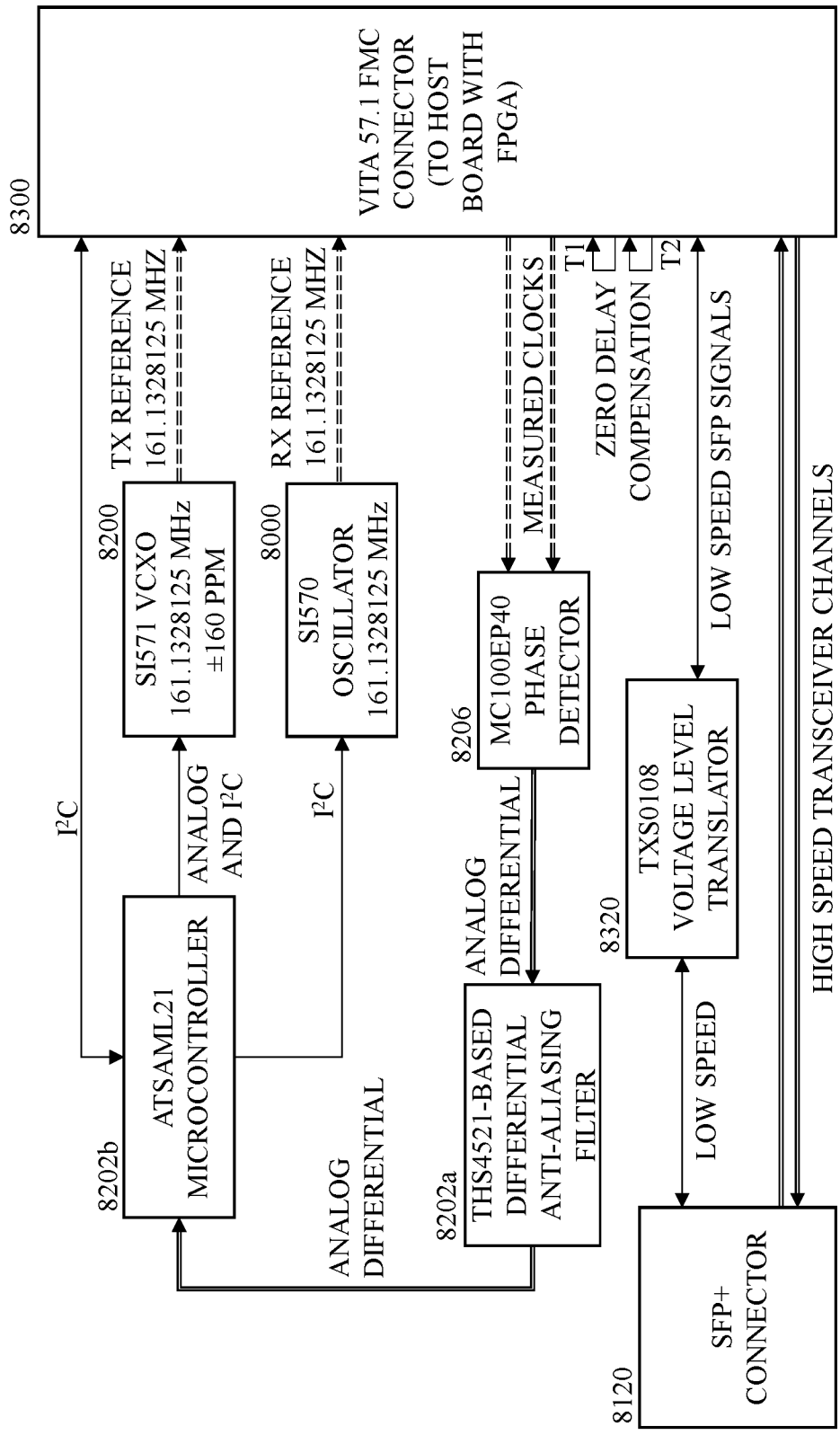

8300
VITA 57.1 FMC CONNECTOR (TO HOST BOARD WITH FPGA)
TX REFERENCE 161.1328125 MHZ
RX REFERENCE 161.1328125 MHZ
8200
SI571 VCXO 161.1328125 MHz ±160 PPM
8000
SI570 OSCILLATOR 161.1328125 MHz
MEASURED CLOCKS
T1
T2
ZERO DELAY COMPENSATION
LOW SPEED SFP SIGNALS
I²C
ANALOG AND I²C
I²C
8206
MC100EP40 PHASE DETECTOR
ANALOG DIFFERENTIAL
8320
TXS0108 VOLTAGE LEVEL TRANSLATOR
8202b
ATSAML21 MICROCONTROLLER
8202a
THS4521-BASED DIFFERENTIAL ANTI-ALIASING FILTER
ANALOG DIFFERENTIAL
LOW SPEED
HIGH SPEED TRANSCEIVER CHANNELS
8120
SFP+ CONNECTOR

FIG. 8A

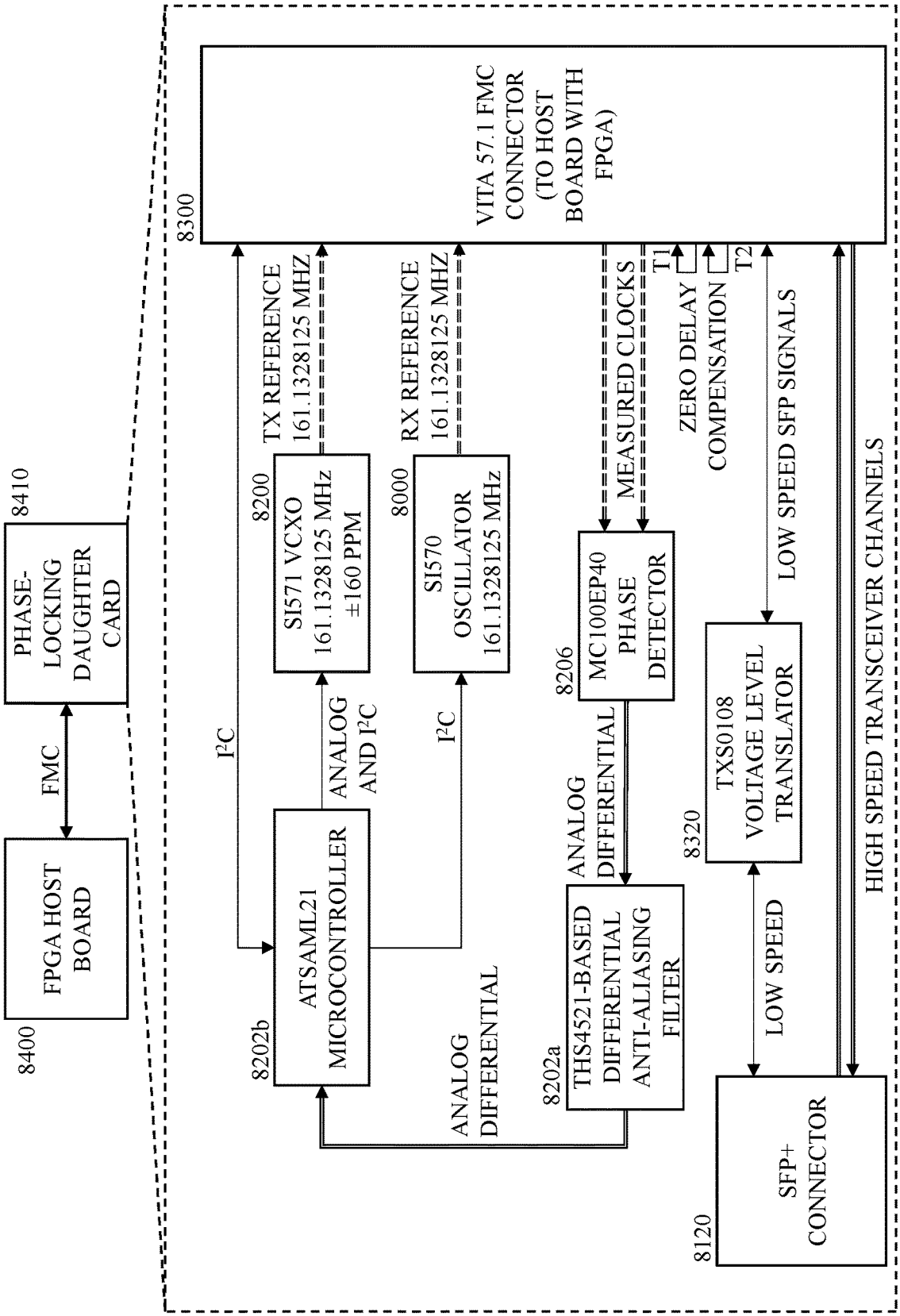

8400
FPGA HOST BOARD
FMC
8410
PHASE-LOCKING DAUGHTER CARD
8300
VITA 57.1 FMC CONNECTOR (TO HOST BOARD WITH FPGA)
8200
SI571 VCXO 161.1328125 MHz ±160 PPM
TX REFERENCE 161.1328125 MHZ
8000
SI570 OSCILLATOR 161.1328125 MHz
RX REFERENCE 161.1328125 MHZ
I²C
ANALOG AND I²C
I²C
8202b
ATSAML21 MICROCONTROLLER
ANALOG DIFFERENTIAL
8202a
THS4521-BASED DIFFERENTIAL ANTI-ALIASING FILTER
8206
MC100EP40 PHASE DETECTOR
ANALOG DIFFERENTIAL
MEASURED CLOCKS
T1
ZERO DELAY COMPENSATION
T2
8320
TXS0108 VOLTAGE LEVEL TRANSLATOR
LOW SPEED SFP SIGNALS
LOW SPEED
HIGH SPEED TRANSCEIVER CHANNELS
8120
SFP+ CONNECTOR

FIG. 8B

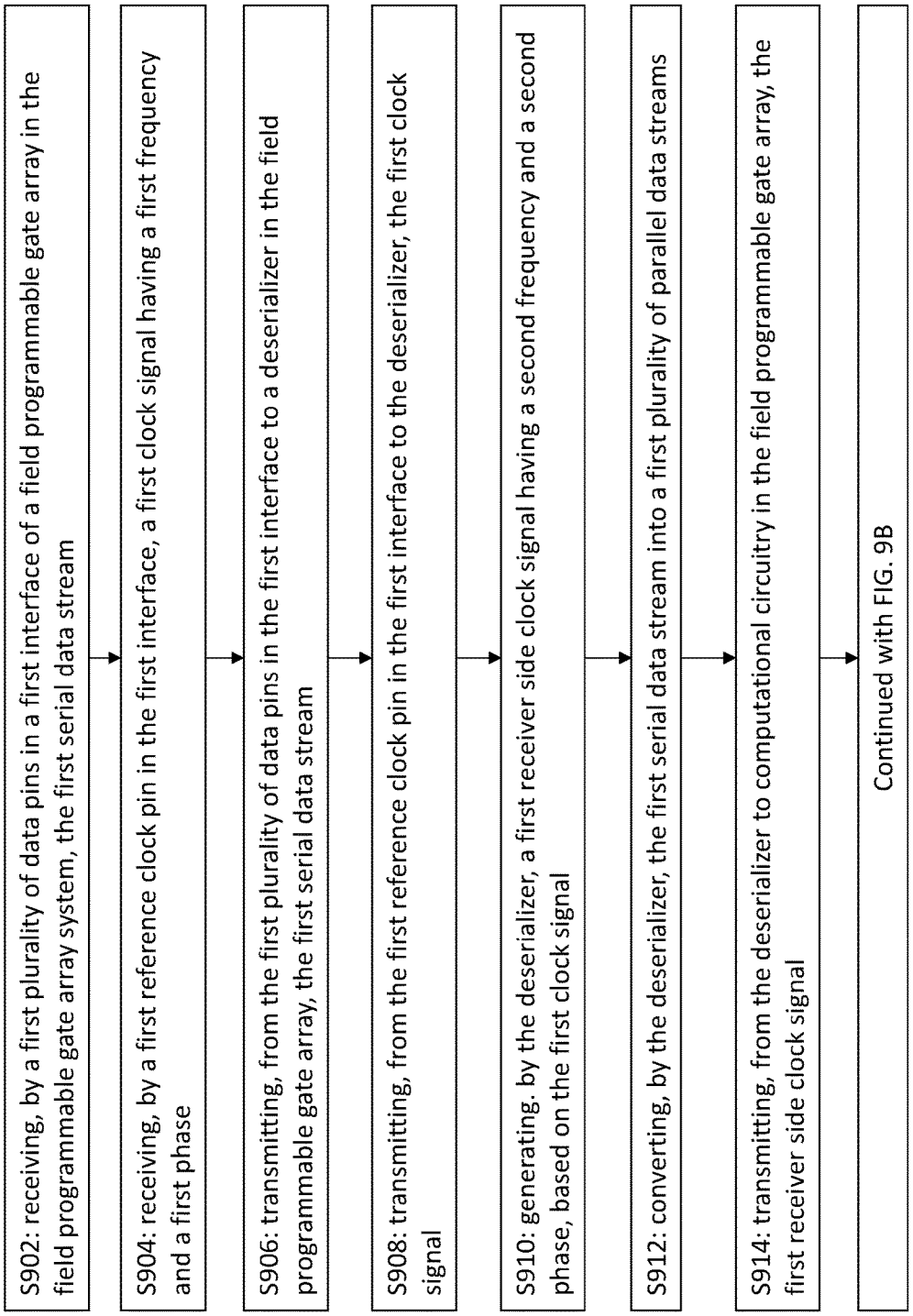

S902: receiving, by a first plurality of data pins in a first interface of a field programmable gate array in the field programmable gate array system, the first serial data stream
S904: receiving, by a first reference clock pin in the first interface, a first clock signal having a first frequency and a first phase
S906: transmitting, from the first plurality of data pins in the first interface to a deserializer in the field programmable gate array, the first serial data stream
S908: transmitting, from the first reference clock pin in the first interface to the deserializer, the first clock signal
S910: generating. by the deserializer, a first receiver side clock signal having a second frequency and a second phase, based on the first clock signal
S912: converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams
S914: transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first receiver side clock signal
Continued with FIG. 9B

FIG. 9A

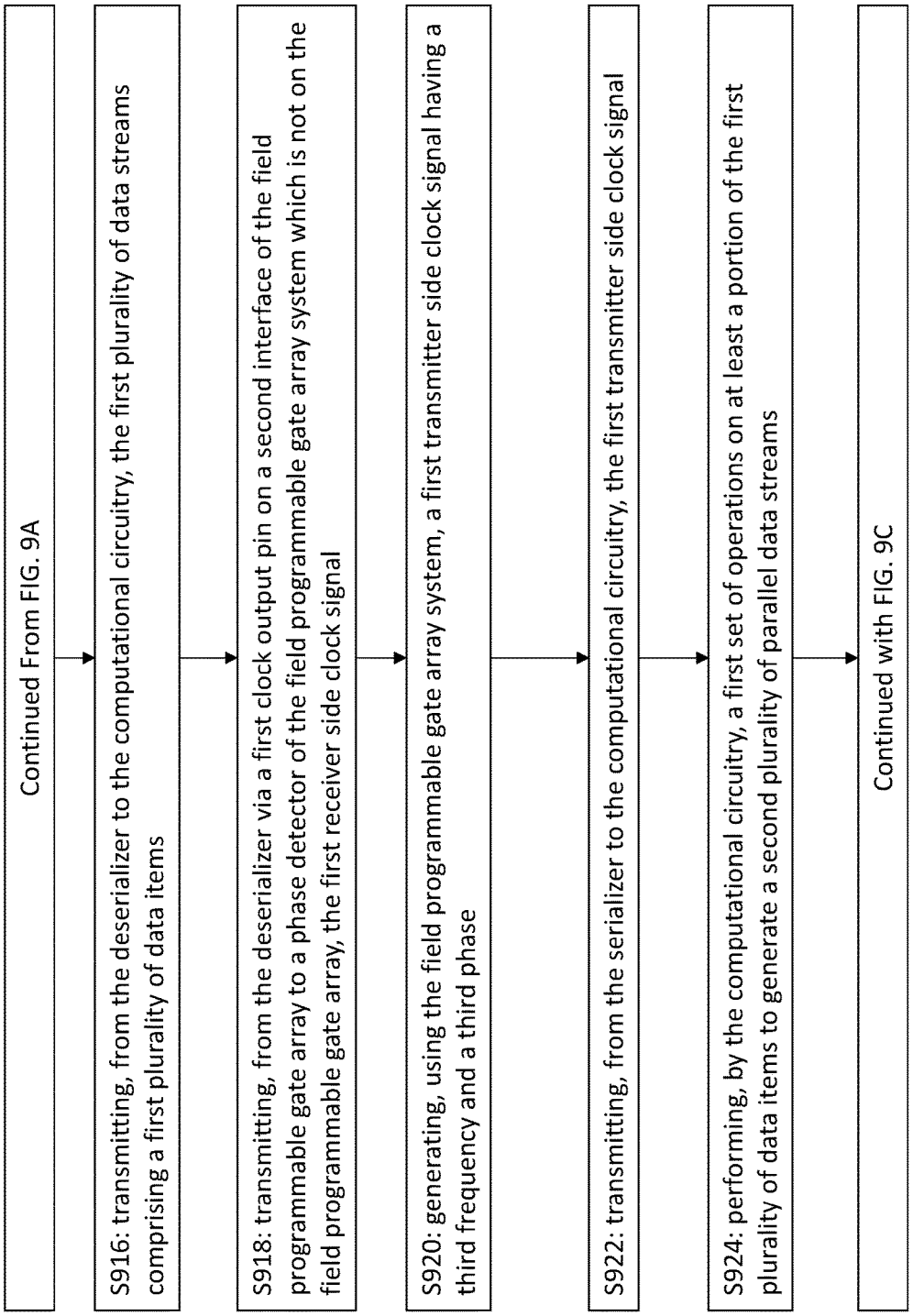

Continued From FIG. 9A
S916: transmitting, from the deserializer to the computational circuitry, the first plurality of data streams comprising a first plurality of data items
S918: transmitting, from the deserializer via a first clock output pin on a second interface of the field programmable gate array to a phase detector of the field programmable gate array system which is not on the field programmable gate array, the first receiver side clock signal
S920: generating, using the field programmable gate array system, a first transmitter side clock signal having a third frequency and a third phase
S922: transmitting, from the serializer to the computational circuitry, the first transmitter side clock signal
S924: performing, by the computational circuitry, a first set of operations on at least a portion of the first plurality of data items to generate a second plurality of parallel data streams
Continued with FIG. 9C

FIG. 9B

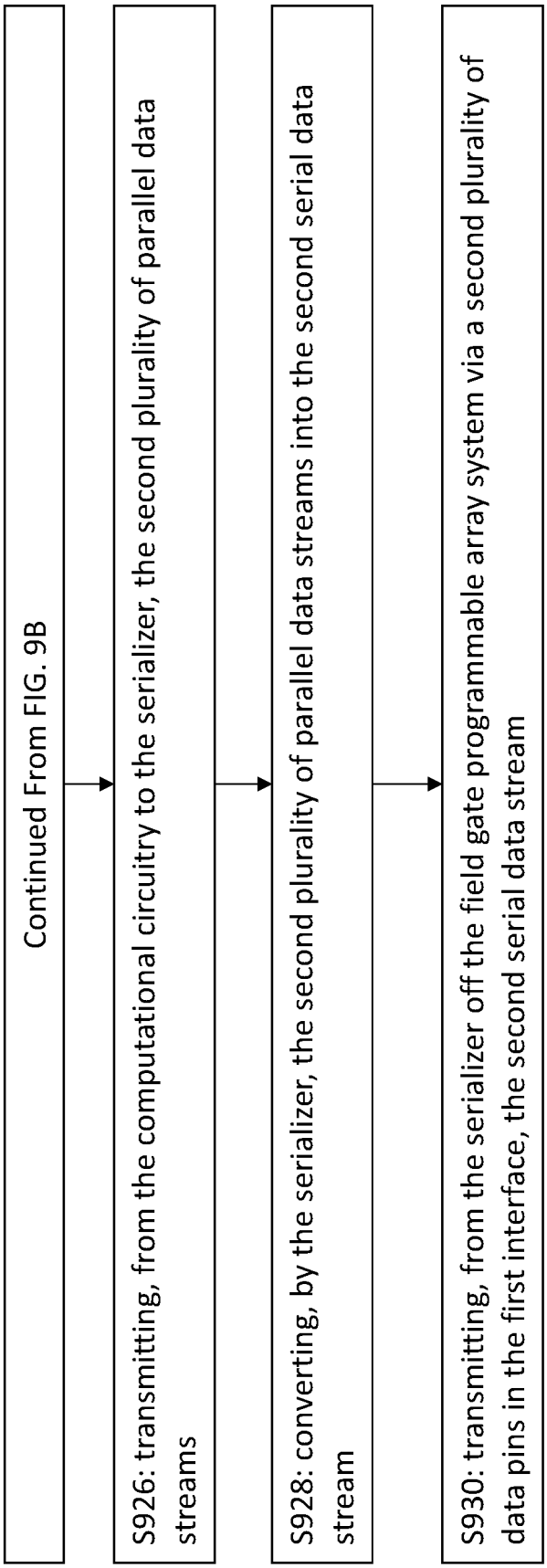

Continued From FIG. 9B
S926: transmitting, from the computational circuitry to the serializer, the second plurality of parallel data streams
S928: converting, by the serializer, the second plurality of parallel data streams into the second serial data stream
S930: transmitting, from the serializer off the field gate programmable array system via a second plurality of data pins in the first interface, the second serial data stream

FIG. 9C

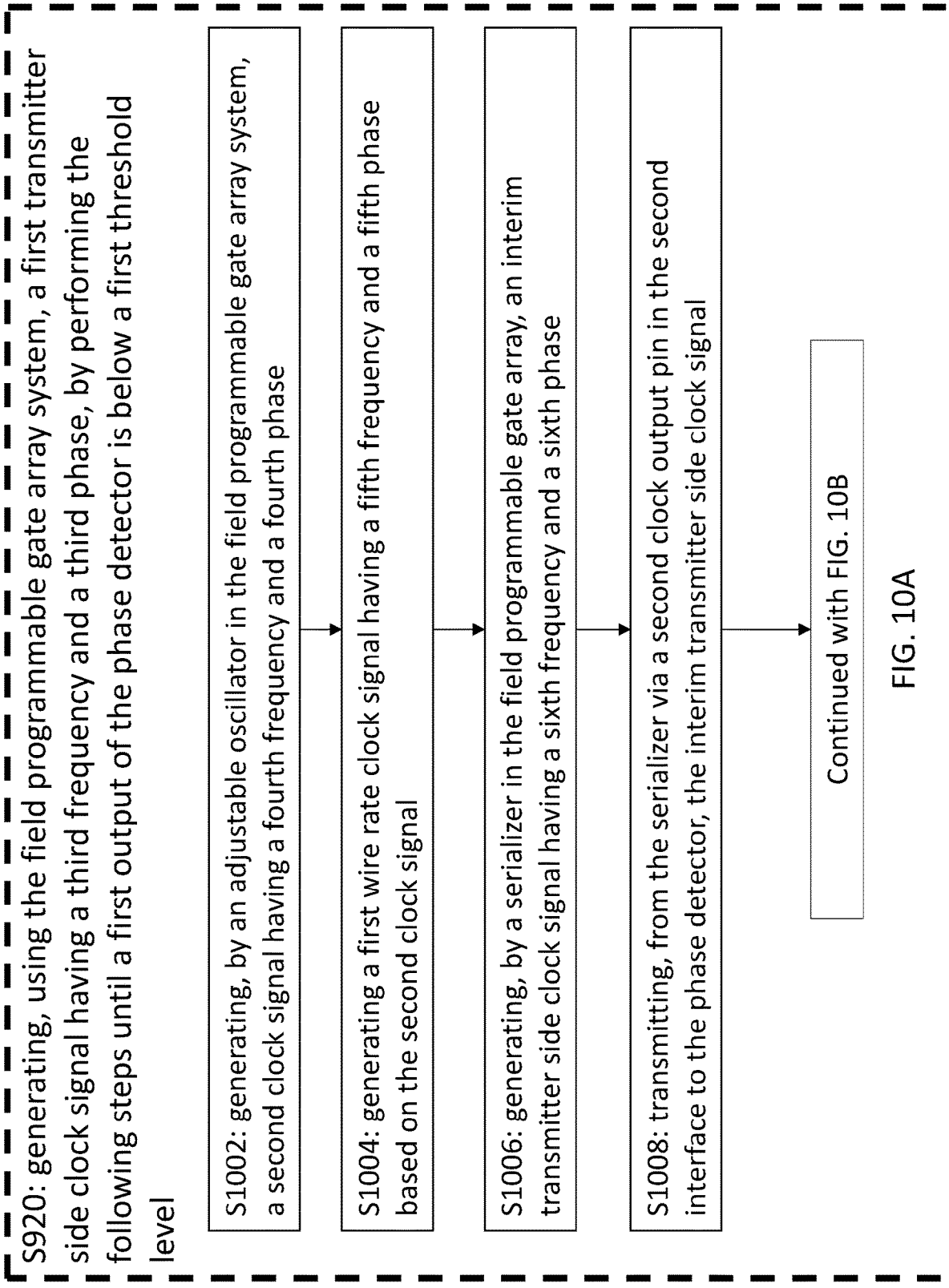

S920: generating, using the field programmable gate array system, a first transmitter side clock signal having a third frequency and a third phase, by performing the following steps until a first output of the phase detector is below a first threshold level
S1002: generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal having a fourth frequency and a fourth phase
S1004: generating a first wire rate clock signal having a fifth frequency and a fifth phase based on the second clock signal
S1006: generating, by a serializer in the field programmable gate array, an interim transmitter side clock signal having a sixth frequency and a sixth phase
S1008: transmitting, from the serializer via a second clock output pin in the second interface to the phase detector, the interim transmitter side clock signal
Continued with FIG. 10B

FIG. 10A

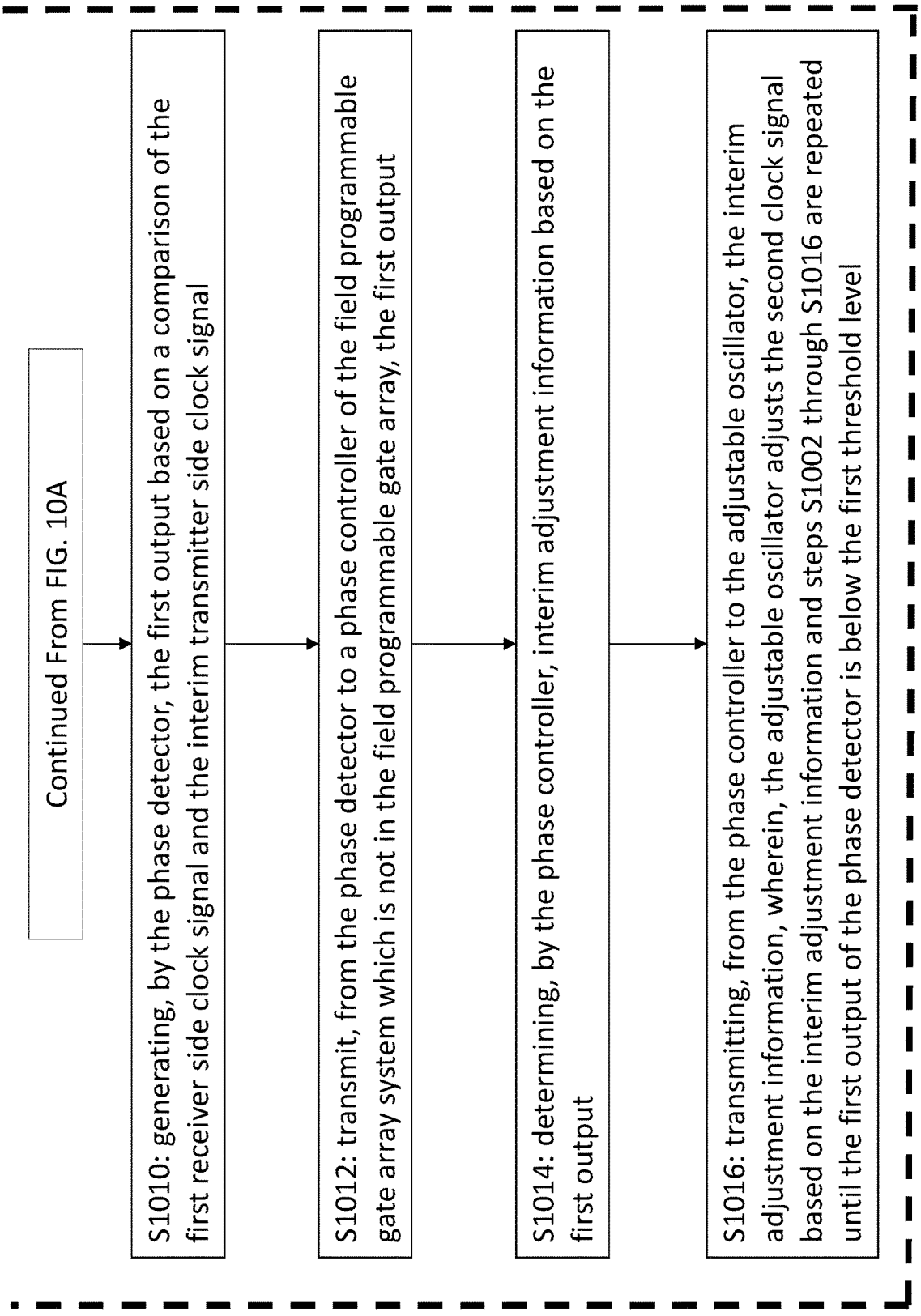

Continued From FIG. 10A
S1010: generating, by the phase detector, the first output based on a comparison of the first receiver side clock signal and the interim transmitter side clock signal
S1012: transmit, from the phase detector to a phase controller of the field programmable gate array system which is not in the field programmable gate array, the first output
S1014: determining, by the phase controller, interim adjustment information based on the first output
S1016: transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information, wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information and steps S1002 through S1016 are repeated until the first output of the phase detector is below the first threshold level

FIG. 10B

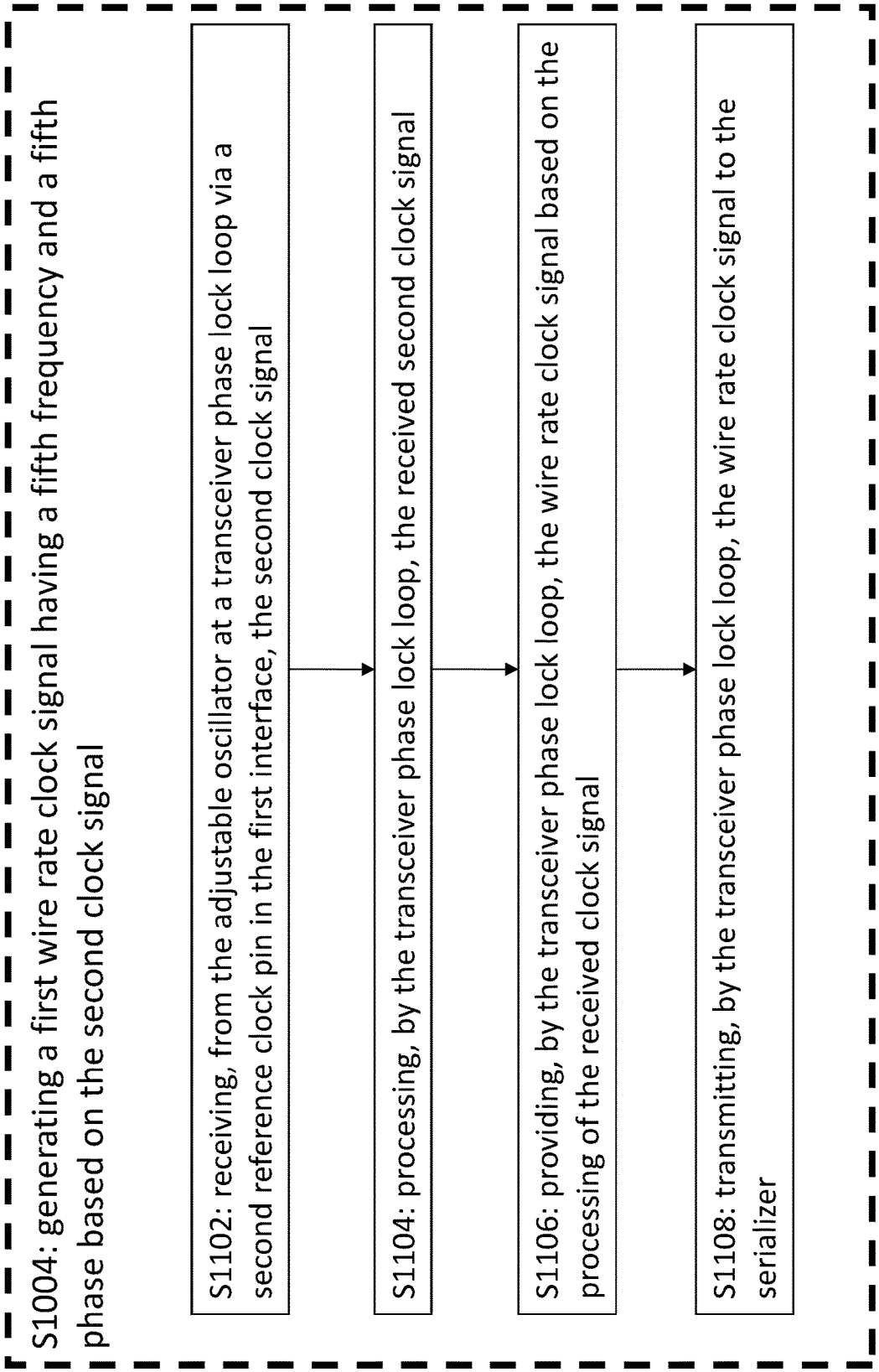

S1004: generating a first wire rate clock signal having a fifth frequency and a fifth phase based on the second clock signal
S1102: receiving, from the adjustable oscillator at a transceiver phase lock loop via a second reference clock pin in the first interface, the second clock signal
S1104: processing, by the transceiver phase lock loop, the received second clock signal
S1106: providing, by the transceiver phase lock loop, the wire rate clock signal based on the processing of the received clock signal
S1108: transmitting, by the transceiver phase lock loop, the wire rate clock signal to the serializer

FIG. 11

FIELD PROGRAMMABLE GATE ARRAY WITH EXTERNAL PHASE-LOCKED LOOP

REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of and priority to U.S. Provisional Patent Application No. 62/755,804, filed on Nov. 5, 2018 and entitled FIELD PROGRAMMABLE GATE ARRAY WITH EXTERNAL PHASE-LOCKED LOOP, and U.S. Provisional Patent Application No. 62/803,017, filed on Feb. 8, 2019 and entitled FIELD PROGRAMMABLE GATE ARRAY WITH EXTERNAL PHASE-LOCKED LOOP, the entire content of each of which is hereby incorporated by reference herein.

FIELD

The present invention generally relates to a field programmable gate array system. In embodiments, the present invention generally relates to a field programmable gate array and an external phase controller providing phase matching between a receiver clock and a transmitter clock used in the field programmable gate array.

BACKGROUND

Field programmable gate arrays (FPGAs) may be used in applications that require fast processing since FPGAs allow for all computations to occur on a single chip that has massive fine-grained parallelism. For example, FPGAs are used in the financial industry in high frequency trading where the rapid processing of the FPGA is desired. One technological problem with FPGAs is that there is a need to synchronize receiving side and transmitting side clock signals within the FPGA. The prior art sought to address this problem by including a clock domain crossing circuit in the FPGA, however, these circuits inherently add a delay to the processing that takes place in the FPGA, which is not desirable since high frequency trading may include time-stamps that are accurate to the microsecond such that even small delays may present a large problem.

Accordingly, a technical problem is presented in FPGAs in that phase synchronization between the receiver side clock and the transmitter side clock will introduce unwanted latency that results in delay of processing. Accordingly, it would be beneficial to provide an FPGA system that avoids these problems and provides sub-microsecond processing with throughput of at least 10 Gbps. In embodiments, the sub-microsecond processing throughput may be: 10-25 Gbps, 22-33 Gbps, 33-45 Gbps, 45-60 Gbps, 60-80 Gbps, 80-120 Gbps, to name a few.

SUMMARY

An object of the present invention is to address technological challenges that currently exist in phase matching receiver side and transmitter side clocks of a FPGA without introducing unnecessary delay in processing.

This and other objects shall be addressed by embodiments of the present invention as set forth herein.

The present invention generally relates to a field programmable gate array system. In embodiments, the present invention generally relates to a field programmable gate array and an external phase controller providing phase matching between a receiver clock and a transmitter clock used in the field programmable gate array.

In embodiments, a field programmable gate array system includes (a) a field programmable gate array comprising (1) a first interface including: (A) a first reference clock pin, wherein said first reference clock pin is configured to receive a first clock signal having a first frequency and a first phase; (B) a second reference clock pin, wherein said second reference clock pin is configured to receive a second clock signal having a second frequency and a second phase; (C) a first plurality of data pins, wherein said first plurality of data pins is configured to receive a first serial data stream; (D) a second plurality of data pins, wherein said second plurality of data pins is configured to transmit a second serial data stream; (2) a deserializer operationally connected to: (x) the first reference clock pin to receive as a first input the first clock signal and (y) the first plurality of data pins to receive as a second input the first serial data stream, and wherein the deserializer is configured to: (A) convert the first serial data stream into a first plurality of parallel data streams having a first amount of data streams, and (B) generate a first receiver side clock signal based on the first clock signal, wherein the first receiver side clock signal has a third frequency and a third phase; and (C) transmit the first plurality of parallel data streams and the first receiver side clock signal within the field programmable gate array; (3) computational circuitry operationally connected to the deserializer to receive the first plurality of parallel data streams and the first receiver side clock signal, wherein the computational circuitry is configured to perform a first set of operations on the first plurality of parallel data streams to generate a second plurality of parallel processed data streams having a second amount of data streams; (4) a serializer operationally connected to: (x) the second reference clock pin to receive as a third input a first wire rate clock signal based on the second clock signal, wherein the first wire rate clock signal has a fourth frequency and a fourth phase; (y) the second plurality of data pins to transmit as a first output the second serial data stream; (z) the computational circuitry, wherein the serializer receives the second plurality of parallel processed data streams from the computational circuitry and the serializer transmits to the computational circuitry a first transmitter side clock signal including a fifth frequency and a fifth phase; and wherein the serializer is configured to: (A) convert the second plurality of parallel processed data streams into the second serial data stream; (B) generate the first transmitter side clock signal based on the first wire rate signal, wherein the first transmitter side clock signal has the fifth frequency and the fifth phase, wherein the fifth frequency is different than and less than the fourth frequency; and (C) transmit the second serial data stream to the second plurality of data pins for transmission off the field programmable gate array; (5) a second interface including: (A) a first clock output pin configured to transmit the first receiver side clock signal, wherein the first clock output pin is operationally connected to the deserializer; and (B) a second clock output pin configured to transmit the first transmitter side clock signal, wherein the second clock output pin is operationally connected to the serializer; and (b) a phase control circuit, provided outside of the field programmable gate array, wherein the phase control circuit includes: (1) a phase detector operationally connected to the first clock output pin and the second clock output pin of the second interface of the field programmable gate array, and wherein the phase detector is configured to compare the third phase of the receiver side clock signal to the fifth phase of the transmitter side clock signal and to generate a phase difference indicator signal based on a difference between the third phase of the receiver side clock signal and the fifth phase of the trans-

mitter side clock signal; (2) a phase controller operationally connected to the phase detector and configured to receive the phase difference indicator signal, and wherein the phase controller is configured to determine adjustment information based on the phase difference indicator signal; and (3) an adjustable oscillator operationally connected to the phase controller and configured to receive the adjustment information as well as operationally connected to the second reference clock pin of the first interface of the field programmable gate array, wherein the adjustable oscillator is configured to generate the second clock signal including the second frequency and the second phase based on the adjustment information and transmit the second clock signal to the second reference clock pin of the first interface of the field programmable gate array; wherein the transmitter side clock signal and the receiver side clock signal are phase aligned so that there is a fixed phase difference between the third phase and the fifth phase.

In embodiments, the third frequency corresponds to the first frequency.

In embodiments, the third phase is not aligned with the first phase.

In embodiments, the third frequency is different than the first frequency.

In embodiments, the first set of operations does not include clock domain crossing operations that delays processing of the first set of parallel data streams. In embodiments, the adjustment information is used to do at least one of the following (i) set a bias to an oscillator; (ii) to set a divider ratio; and (iii) to set a delay.

In embodiments, the first set of operations includes an arithmetic operation. In embodiments, the first set of operations includes a logical operation. In embodiments, the first set of operations includes a pipeline operation. In embodiments, the first set of operations includes a memory access operation.

In embodiments, the first amount of data streams is the same as the second amount of data streams.

In embodiments, the first amount of data streams and the second amount of data streams are equal to eight (8) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to ten (10) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to sixteen (16) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to twenty (20) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to thirty-two (32) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to forty (40) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to sixty-four (64) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to eighty (80) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to one hundred twenty-eight (128) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to one hundred sixty (160) data streams.

In embodiments, the first amount of data streams and the second amount of data streams are equal to two (2) to the power of N, where N is an integer. In embodiments, the first amount of data streams and the second amount of data streams are equal to ten (10) times two (2) to the power of N, where N is an integer. In embodiments the first amount of data streams and the second amount of data streams are equal to thirty-three (33) times two (2) to the power of N, where N is an integer.

In embodiments, the field programmable gate array further includes a phase lock loop configured to: receive the second clock signal, as a fourth input, to generate the first wire rate clock signal; and (B) transmit the first wire rate clock signal as the third input, to the serializer.

In embodiments, the fourth frequency is equal to x times the fifth frequency, where x is the second amount of data streams.

In embodiments, the fourth frequency is equal to x times the fifth frequency, where x is half of the second amount of data streams.

In embodiments, the fourth frequency is a rational multiple of the second frequency.

In embodiments, the phase difference indicator signal is a pulse signal.

In embodiments, the adjustment information indicates a desired phase. In embodiments, the adjustment information indicates a desired frequency. In embodiments, the adjustment information indicates a change in phase. In embodiments, the adjustment information indicates a change in frequency. In embodiments, adjustment information comprises a voltage. In embodiments, adjustment information comprises a digital transmission. In embodiments adjustment information further comprises a digital transmission.

In some embodiments, the first serial data stream comprises market data, the second serial data stream comprises order entry data, and the first operation comprises a trading algorithm.

In embodiments, the trading algorithm includes the steps of: (a) parsing market data; (b) performing mathematical operations at a portion of the market data; and (c) generating order packets using at least an output of (b).

In embodiments, the first serial data stream includes market data and the second serial stream includes trading data.

In embodiments, the field programmable gate array includes a transceiver phase locked loop operatively connected between the to the serializer and the first reference clock pin.

In embodiments, the transceiver phase locked loop operatively connected between the serializer and the first reference clock pin includes a second adjustable oscillator to provide the first wire rate signal.

In embodiments, a second adjustable oscillator is provided outside of the transceiver phase locked loop and operatively connected to the serializer.

In embodiments, the second adjustable oscillator is programmable with a desired delay.

In embodiments, the desired delay is a constant value

In embodiments, the constant value is zero.

In embodiments, the adjustment information is provided to the second adjustable oscillator to provide the first wire rate signal.

A method for processing a first serial data stream comprising market data, using a field programmable gate array system, to generate a second serial data stream comprising order entry data, wherein the method includes the steps of: (A) receiving, by a first plurality of data pins in a first interface of a field programmable gate array in the field programmable gate array system, the first serial data stream; (B) receiving, by a first reference clock pin in the first interface, a first clock signal having a first frequency and a first phase; (C) transmitting, from the first plurality of data pins in the first interface to a deserializer in the field

programmable gate array, the first serial data stream; (D) transmitting, from the first reference clock pin in the first interface to the deserializer, the first clock signal; (E) generating, by the deserializer, a first receiver side clock signal having a second frequency and a second phase, based on the first clock signal; (F) converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams; (G) transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first receiver side clock signal; (H) transmitting, from the deserializer to the computational circuitry, the first plurality of data streams comprising a first plurality of data items and a first amount of data streams; (I) transmitting, from the deserializer via a first clock output pin on a second interface of the field programmable gate array to a phase detector that is part of the field programmable gate array system and not part of the field programmable gate array, the first receiver side clock signal; (J) generating, using the field programmable gate array system, a first transmitter side clock signal having a third frequency and a third phase, by performing the following steps until a first output of the phase detector is below a first threshold level: (i) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal having a fourth frequency and a fourth phase; (ii) generating a first wire rate clock signal having a fifth frequency and a fifth phase based on the second clock signal; (iii) generating, by a serializer in the field programmable gate array, an interim transmitter side clock signal having a sixth frequency and a sixth phase; (iv) transmitting, from the serializer via a second clock output pin in the second interface to the phase detector, the interim transmitter side clock signal; (v) generating, by the phase detector, the first output based on a comparison of the first receiver side clock signal and the interim transmitter side clock signal; (vi) transmitting, from the phase detector to a phase controller of the field programmable gate array system which is not in the field programmable gate array, the first output; (vii) determining, by the phase controller, interim adjustment information based on the first output; and (viii) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information; wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information and steps (i) through (viii) are repeated until the first output of the phase detector is below the first threshold level; (K) transmitting, from the serializer to the computational circuitry, first transmitter side clock signal; (L) performing, by the computational circuitry, a first set of operations on at least a portion of the first plurality of data items to generate a second plurality of parallel data streams comprising a second amount of data streams; (M) transmitting, from the computational circuitry to the serializer, the second plurality of parallel data streams; (N) converting, by the serializer, the second plurality of parallel data streams into the second serial data stream; and (O) transmitting, from the serializer off the field gate programmable array system via a second plurality of data pins in the first interface, the second serial data stream.

In embodiments, step (a) is performed before step (b).

In embodiments, step (b) is performed before step (a).

In embodiments, step (a) and step (b) are performed contemporaneously.

In embodiments, step (c) is performed before step (d).

In embodiments, step (d) is performed before step (c).

In embodiments, step (c) and step (d) are performed contemporaneously.

In embodiments, step (e) is performed before step (f).

In embodiments, step (f) is performed before step (e).

In embodiments, step (e) and step (f) are performed contemporaneously.

In embodiments, the second frequency corresponds to the first frequency.

In embodiments, the second phase is not aligned with the first phase.

In embodiments, the second frequency is different from the first frequency.

In embodiments, step (g) is performed before step (h).

In embodiments, step (h) is performed before step (g).

In embodiments, step (g) and step (h) are performed contemporaneously.

In embodiments, the transmission of the first receiver side clock signal from the deserializer to the phase detector is also via a first zero delay buffer phase lock loop of the field programmable gate array before being transmitted via the first clock output pin of the second interface.

In embodiments, the third frequency corresponds to the second frequency.

In embodiments, the third phase is aligned with the second phase.

In embodiments, a difference between the third phase and the second phase is less than a second threshold level.

In embodiments, the sixth frequency corresponds to the third frequency.

In embodiments, the sixth phase is not aligned with the third phase prior to the first output of the phase detector being below the first threshold level.

In embodiments, the transmission of the interim transmitter side clock signal from the serializer to the phase detector is also via a second zero delay buffer phase lock loop of the field programmable gate array before being transmitted via the second clock output pin of the second interface.

In embodiments, the first set of operations does not include clock domain crossing operations that delays processing of the first set of parallel data streams.

In embodiments, the first amount of data streams is the same as the second amount of data streams.

In embodiments, the first amount of data streams and the second amount of data streams is eight (8) data streams. In embodiments, the first amount of data streams and the second amount of data streams is ten (10) data streams. In embodiments, the first amount of data streams and the second amount of data streams is sixteen (16) data streams. In embodiments, the first amount of data streams and the second amount of data streams is twenty (20) data streams. In embodiments, the first amount of data streams and the second amount of data streams is thirty-two (32) data streams. In embodiments, the first amount of data streams and the second amount of data streams is forty (40) data streams. In embodiments, the first amount of data streams and the second amount of data streams is sixty-four (64) data streams. In embodiments, the first amount of data streams and the second amount of data streams is eighty (80) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to one hundred twenty-eight (128) data streams. In embodiments, the first amount of data streams and the second amount of data streams are equal to one hundred sixty (160) data streams.

In embodiments, the first amount of data streams and the second amount of data streams are equal to two (2) to the power of N, where N is an integer. In embodiments, the first amount of data streams and the second amount of data streams are equal to ten (10) times two (2) to the power of N, where N is an integer. In embodiments the first amount

of data streams and the second amount of data streams are equal to thirty-three (33) times two (2) to the power of N, where N is an integer.

In embodiments, at least a portion of the first set of operations is performed prior to step (k).

In embodiments, at least a portion of the first set of operations is performed after step (k).

In embodiments, all of the first set of operations is performed after step (k).

In embodiments, the transmission of the second serial data stream from the serializer off the field programmable gate array is transmitted to an input/output module off the field programmable gate array.

In embodiments, the first serial data stream includes market data and the second serial data stream includes trading data.

In embodiments, the step of generating the first wire rate clock signal further includes: (a) receiving, from the adjustable oscillator at a transceiver phase lock loop via a second reference clock pin in the first interface, the second clock signal; (b) processing, by the transceiver phase lock loop, the received second clock signal; (c) providing, by the transceiver phase lock loop, the wire rate clock signal based on the processing of the received clock signal; and (d) transmitting, by the transceiver phase lock loop, the wire rate clock signal to the serializer.

In embodiments, a phase lock loop generates the first wire rate clock signal. In embodiments, the phase lock loop is on the field programmable gate array. In embodiments, the phase lock loop that is part of the field programmable gate array system and not part of the field programmable gate array.

In embodiments, the step of generating the first wire rate clock signal further includes: (a) processing the second clock signal; (b) providing the wire rate clock signal based on the processing of the second clock signal; and (c) transmitting, to the serializer, the wire rate clock signal.

In embodiments, the sixth frequency corresponds to the third frequency. In embodiments, the sixth phase is not aligned with the third phase.

BRIEF DESCRIPTION OF THE DRAWINGS

Exemplary embodiments of the present invention will be described with references to the accompanying figures, wherein:

FIG. **1** is a block diagram of a conventional field programmable gate array;

FIG. **1**A is a block diagram of an exemplary transceiver suitable for use in the field programmable gate array of FIG. **1**;

FIG. **1**B is an exemplary block diagram of a Clock Domain Crossing Circuit suitable for use in the field programmable gate array of FIG. **1**;

FIG. **1**C illustrates exemplary input output waveforms of the Clock Domain Crossing Circuit of the field programmable gate array of FIG. **1**;

FIG. **2** is a block diagram of a field programmable gate array system in accordance with an embodiment of the present invention;

FIG. **3**A is an exemplary block diagram of a deserializer suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **3**B is an exemplary circuit diagram of a continuous-time linear equalizer (CTLE) suitable for use in the deserializer of FIG. **3**A in accordance with an embodiment of the present invention;

FIG. **3**C is an exemplary block diagram of a clock and data recovery (CDR) circuit suitable for use in the deserializer of FIG. **3**A in accordance with an embodiment of the present invention;

FIG. **3**D is an exemplary circuit diagram of a decision circuit suitable for use in the deserializer of FIG. **3**A in accordance with an embodiment of the present invention;

FIG. **3**E is an exemplary block diagram of a deserializer suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **4**A is an exemplary block diagram of a serializer suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **4**B is an exemplary block diagram of a serializer suitable for use in field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **5**A is an exemplary block diagram of a phase detector suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **5**B is another exemplary block diagram of a phase detector suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **5**C is another exemplary block diagram of a phase detector suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **5**D is another exemplary block diagram of a phase detector suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention;

FIG. **6**A illustrates an example of a phase difference between the receiver side clock and the transmitter side clock of an FPGA in accordance with an embodiment of the present invention;

FIG. **6**B illustrates an example of a phase difference between receiver clock and a transmitter clock of an FPGA in accordance with an embodiment of the present invention;

FIG. **6**C illustrates an example of a phase difference between receiver clock and a transmitter clock of an FPGA in accordance with an embodiment of the present invention;

FIG. **6**D illustrates an example of an inverted phase difference between receiver clock and a transmitter clock of an FPGA in accordance with an embodiment of the present invention;

FIG. **7**A is a block diagram of a field programmable gate array system in accordance with another embodiment of the present invention;

FIG. **7**B is a block diagram of another field programmable gate array system in accordance with another embodiment of the present invention;

FIG. **7**C is a block diagram of another field programmable gate array system in accordance with another embodiment of the present invention;

FIG. **8**A is a block diagram of another field programmable gate array system in accordance with an embodiment of the present invention; and

FIG. **8**B is another block diagram of another field programmable gate array system in accordance with an embodiment of the present invention;

FIGS. **9**A-C are exemplary flow charts of a process implementing the field programmable gate array system in accordance with an embodiment of the present invention;

FIGS. **10**A-**10**B are exemplary flow charts of a process for generating a transmitter side clock signal in accordance with an embodiment of the present invention; and

FIG. **11** is an exemplary flow chart of a process for generating a wire clock signal in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

The present invention generally relates to a field programmable gate array system. In embodiments, the present invention generally relates to a field programmable gate array and an external phase controller providing phase matching between a receiver clock and a transmitter clock used in the field programmable gate array.

FIG. **1** illustrates an exemplary schematic of a FPGA **100**. Examples of currently available FPGAs include the XCVU3P-2FFVC1517E, Intel 1SG280LH3F55E3VG, Xilinx XCVU9P-2FLGA2104E the Vitrex® Ultrascale, the Vitrex® Ultrascale Plus, the Stratix® V, the Stratix® 10, XILINX XC3042, the Intel EP4CE6E22C8LN and the Lattice ICE40LP384-SG32, to name a few.

In embodiments, FPGA **100** may include an FPGA Core **106** and a number of peripheral systems. In embodiments, the FPGA Core **106** includes 4 main components: logic elements, digital signal processor blocks ("DSP blocks"), clock distribution components, and memories, to name a few, and may generally be referred to as the FPGA core, or simply logic core, **106**. In embodiments, the peripheral systems may include CPU cores, reconfiguration controllers, security features, hardened logic functions, FPGA transceiver banks **102**, hardened high-speed interface accelerators (e.g. PCIe or 100 GbE controllers, to name a few), general-purpose I/O pins, memory interface controllers, larger memories, analog components (e.g. ADCs or DACs to name a few), and **3**D-stacked memories (e.g. HBM), to name a few. The FPGA Transceiver banks **102** receive serial data to be processed from outside the FPGA and then transmit serial data out of the FBGA after being processed by internal circuitry of the FPGA **100**. Some FPGA's may not include FPGA Transceiver banks.

In embodiments, data may be received by a receiver side of the transceiver banks **102** in the FPGA **100** at rates such as 10 or 25 Gbps, to name a few. In embodiments, rates may be between a range of 9-25 Gbps, 10-25 Gbps, 22-33 Gbps, 33-45 Gbps, 45-60 Gbps, 60-80 Gbps, or 80-120 Gbps, to name a few. An exemplary transceiver (including deserializer **104'** and serializer **110'**) suitable for use in FPGA **100** is shown in FIG. **1**A. In embodiments, the input data may be received from I/O module **120**. The input data may then be deserialized, using deserializer **104**, for example, and parallelized into parallel data streams, such as 16-64 bits wide at $1/16^{th}$ to $1/64^{th}$ of the frequency of the incoming data stream, to name a few. In embodiments, FPGA transceivers may support parallel streams with widths of, for example, 8 bits, 10 bits, 16 bits, 20 bits, 32 bits, 40 bits, 64 bits, 80 bits, 128 bits, and 160 bits (with corresponding clock division), to name a few. In some FPGAs, a physical coding sublayer (PCS) may be used in the transceiver **102**. In some FPGAs, the PCS of the transceiver **102** may be bypassed and so that the corresponding PCS functions may be done in the FPGA core **106**. After deserialization, data is available for computation in the FPGA logic core **106** where computation is completed. In embodiments, the computation of the data may be computed by receive-side computation **110***a* of the FPGA core **106**. Similarly, data may be computed by the transmit-side computation **110***b*.

In embodiments, a REFERENCE CLOCK signal is provided, by Oscillator or Clock Generator **122**, to both the deserializer **104** and the serializer **110**. The REFERENCE CLOCK signal is received by the serializer **110** via the transceiver PLL **108** (Phase-Locked Loop). The receiver side clock signal RXCLOCK, however, is different in both frequency and phase from the REFERENCE CLOCK based on the deserialization process that occurs in the deserializer **104**. Similarly, the transmitter clock signal TXCLOCK, while based on the REFERNCE CLOCK signal varies in frequency and phase based on processing that occurs in the serializer **110**. As a result, the RXCLOCK signal and TXCLOCK signal will be out of phase. A known solution to this issue is the inclusion of the RX to TX clock domain crossing circuit **112**. An exemplary Clock Domain Crossing circuit is suitable for use in FPGA **100** is shown as clock domain crossing circuit **112'** in connection with FIG. **1**B. As is noted above, the inclusion of the RX to TX clock domain crossing circuit **112** creates a technical problem, introducing an inherent delay in the FPGA **100**, which is undesirable. The technical problem of an inherent delay is illustrated in FIG. **1**C. FIG. **1**C illustrates an example of a phase difference between the receiver side clock and the transmitter side clock of the conventional field programmable gate array of FIG. **1**.

Thereafter, the data it transmitted out of the FPGA **100**. After all computation (computation by receive-side computation **110***a* and/or transmit-side computation **110***b*) and clock domain crossing (by RX to TX clock domain crossing **112**) is completed, the data goes back to the transmit side of the FPGA transceiver **102** to be serialized by the serializer **110** out on an output wire. The serializer **110** (e.g. the transmitter) typically runs off of a fast clock generated (e.g. 10 Gbps, 25 Gbps, or a range from 10 Gbps to 25 Gbps, or a range from 1 Gbps to 100 Gbps, to name a few) by the transceiver phase-locked loop **108** within the transceiver bank **102** from a reference clock signal received by oscillator or clock generator **122**.

The receiver (deserializer **104**) and transmitter (serializer **110**) of the transceiver **102** operate on different clocks that are independently generated within the transceiver **102**. The receiver clock domain is typically generated by a clock and data recovery (CDR) circuit from the incoming data stream. The transmitter clock is typically generated by the transceiver phase-locked loop **108**. In conventional FPGAs, phase matching or synchronizing is provided using the clock domain crossing circuit **112** that adjusts the phases of the two clock domains. The clock domain crossing circuit **112** may be an asynchronous FIFO or an asynchronous gearbox, to name a few. In embodiments, domain crossing circuit **112** may be a mesochronous clock crossing circuit. In embodiments, clock domain crossing circuit **112** may be instantiated within transceiver **102** of FPGA **100**, although this structure has a higher latency cost than implementing it in logic fabric **106** of FPGA **100**. As discussed above, a significant drawback of the clock domain crossing circuit **112** is that it adds latency related to the phase difference between the clocks plus the latency of the synchronizers used, and does not perform any computation, such that it slows the effective processing speed of FPGA **100**.

FIG. **2** illustrates a block diagram of a field programmable gate array (FPGA) system in accordance with an embodiment of the present disclosure that provides phase control while minimizing latency. In embodiments, the system generally includes FPGA **2100** which preferably includes FPGA transceiver banks **2102**. In embodiments, the FPGA transceiver banks **2102** includes deserializer **2104** on the receiver end of FPGA **2100**, transceiver PLL **2108**, and serializer **2110** on the transmitter end of FPGA **2100**. Deserializer **2104** may be similar to the deserializer **2014'** and deserializer **2104"** discussed below in connection with FIGS. **3**A-**3**E, the descriptions of which applying herein. Serializer **2110** may be similar to serializer **2110'** and serializer **2110"** discussed below in connection with FIGS. **4**A and **4**B respectively, the descriptions of which applying herein. In embodiments, deserializer **2104** receives a serial data stream via a first plurality of data pins in a first interface from the I/O module **2120** and provides deserialized data to the FPGA core **2106**. More specifically, the deserializer **2104** will provide parallel data streams to the FPGA core **2106** for processing. As illustrated, this parallel data is provided to receiver side computational element **2210***a* where processing and computation takes place. The resulting data may then be provided to transmitting side computational element **2210***b* where additional computation and processing may take place. FIG. **2** illustrates computational logic **2210** as a single element. In embodiments, the computational logic **2210** may be implemented as a plurality of logic elements. In embodiments, computational logic **2210** may also include one or more of the following: RAM blocks or dedicated hardened accelerators. In embodiments, when all computation is complete, the resulting data may be provided to serializer **2110** and then transmitted out of FPGA **2100**, preferably via I/O module **2120**, for example. In embodiments, the I/O module **2120** may be a direct soldered cable, on-chip optics, or an on-board optical transceiver, to name a few. I/O Module **2120**, may be, for example, a Samtec® I/O Module Firefly® I/O Module, to name a few. Additionally, the I/O module may be designed to meet one or more of the following standards: SFP, a QSFP, a MicroQSFP, QSFPDD to QSFP-DD, or a QSFPDD, to name a few.

FPGA **2100**, in embodiments may include one or more interfaces, such as pins. By way of example, a first interface of the one or more interfaces, may include a first plurality of pins of FPGA **2100**. The first plurality of pins may be used to transmit and/or receive data and/or signals. In embodiments, the first plurality of pins may include a first reference clock pin. The first reference clock pin may be operationally connected to deserializer **2104**. In embodiments, the first reference clock pin may also be operationally connected to fixed-frequency oscillator **2000** such that a first clock signal having a first phase and a first phase may be transmitted from fixed-frequency oscillator **2000** to deserializer **2104** via the first reference clock pin. In embodiments, the first clock signal may have a frequency between 50 and 800 MHz.

In embodiments, the first plurality of pins may also include a second reference clock pin. The second reference clock pin may, in embodiments, be operationally connected to transceiver PLL **2108**. The second reference clock pin, in embodiments, may be operationally connected to frequency or phase adjustable oscillator **2200** such that a second clock signal having a second phase and a second frequency may be transmitted from the frequency or phase-adjustable oscillator **2200** to transceiver PLL **2108** via the second reference clock pin. In embodiments, the second reference clock pin may be operationally connected to serializer **2110** such that the second clock signal may be transmitted from the frequency or phase-adjustable oscillator **2200** to serializer **2110** via the second reference clock pin. In embodiments, the second reference clock pin may also receive a first wire rate clock signal.

In embodiments, the first plurality of pins may also include a first plurality of data pins. The first plurality of data pins may be operationally connected deserializer **2104**. In embodiments, the first plurality of data pins may also be operationally connected to I/O module **2120** such that a first serial data stream may be transmitted from external connection **2122** to deserializer via the I/O module **2120** and the first plurality of data pins. In embodiments, the first serial data stream may range between 10 to 25 Gbps.

In embodiments, the first plurality of pins may also include a second plurality of data pins. The second plurality of data pins may be operationally connected to serializer **2110**. In embodiments, the second plurality of data pins may also be operationally connected to I/O module **2120** such that a second serial data stream may be transmitted from serializer **2110** to the external connection **2122** via the I/O module **2120** and the second plurality of data pins. In embodiments, the second serial data stream may range between 10 to 25 Gbps.

Although reference is made to separate first reference clock pin, second reference clock pin, first plurality of data pins and second plurality of data pins, in embodiments, each pin may be a fixed-function, fixed-location pin, connected internally through a wire. In embodiments, each pin may be a multiplexed fixed-function pin, connected internally through a multiplexer or crossbar. In embodiments, each pin may be a general purpose I/O pin connected through the FPGA core **2106**. In embodiments, one or more pin may be fixed function, fixed location pines, while other pins may me multiplexed fixed function pins and/or a general purpose I/O pin.

In embodiments, the one or more interfaces may also include at least a second interface. The second interface of the one or more interfaces, may include a second plurality of pins of FPGA **2100**. The second plurality of pins may be used to transmit and/or receive data and/or signals. In embodiments, the second plurality of pins may include a first clock output pin. The first clock output pin may be operationally connected to deserializer **2104**. In embodiments, the first clock output pin may also be operationally connected to computation logic **2210** such that a receiver side clock signal is transmitted from the deserializer **2104** to computation logic **2210** via the first clock output pin. Additionally, in some embodiments, the first clock output pin may also be operationally connected to zero-delay buffer PLL **2208***a* such that the receiver side clock is transmitted from the deserializer **2104** to the zero-delay buffer PLL **2208***a* via the first clock output pin. In embodiments the receiver side clock signal may have a frequency ranging between 100-650 MHz.

In embodiments, the second plurality of pins may also include a second clock output pin. The second clock output pin may be operationally connected to serializer **2110**. In embodiments, the second clock output pin may also be operationally connected to computation logic **2210** such that a transmitter side clock signal is transmitted from the serializer **2110** to computation logic **2210** via the second clock output pin. Additionally, in some embodiments, the second clock output pin may also be operationally connected to zero-delay buffer PLL **2208***b* such that the transmitter side clock is transmitted from the serializer **2110** to the zero-delay buffer PLL **2208***b* via the second clock output pin. In

embodiments the transmitter side clock signal may have a frequency ranging between 100-650 MHz.

In embodiments, the serializer **2110** and deserializer **2104** (SERDES) circuit(s) may contain two major sections: (a) an analog side whose purpose is signal cleaning, and (2) a digital side which turns the analog signal into bits and converts between parallel and serial data streams. In embodiments, the analog side may include a few different types of amplifiers to provide signal cleaning. In embodiments, the adjustable oscillator **2200** may be implemented in a variety of ways. In embodiments, the adjustable oscillator **2200** may be implemented as a voltage controlled oscillator. A voltage controller oscillator may be implemented using a variety of architectures. In embodiments, other types of oscillators may be used including negative-resistance oscillators, Clapp oscillators, Colpitts oscillators, ring oscillators, and varactor-tuned oscillators, to name a few.

In embodiments, a voltage controller crystal oscillator may be used as the adjustable oscillator **2200**, for example, the Si550 from Silicon Labs. In embodiments, a numerically/digitally-controlled oscillator may be used as the adjustable oscillator **2200**, which is a digital version of an analog VCO, and may use switched circuit elements or a fixed frequency oscillator and a digital PLL to adjust the frequency.

In embodiments, the adjustable oscillator **2200** may be implemented with a digital delay line, e.g. the SY89295U from Micrel.

In embodiments, the adjustable oscillator **2200** may be implemented as a voltage-controlled delay element, e.g. the HMC910 provided by Analog devices.

As illustrated in FIG. **2**, in embodiments, the receiver side clock signal RXCLOCK is based on, but has a different frequency and phase than the REFERENCE CLOCK signal provided by oscillator **2200** based on the operation of the deserializer **2104**, as is discussed above with respect to the serializer **110**. In embodiments, the transmitter side clock signal TXCLOCK provided from the deserializer **2104** may be provided by transceiver phase locked loop **2108**, which is provided inside FPGA **2100**, and may be controlled by frequency or phase adjustable oscillator **2200**, external to FPGA **2100**. In this manner, the frequency or phase of the transmitter side clock signal TXCLOCK may be adjusted.

In embodiments, an external phase controller **2202** is connected to the frequency or phase adjustable oscillator **2200** and provides control signals to allow for adjustment of the phase of at least the transmitter side clock signal TXCLOCK. The phase of the incoming data, in embodiments, may be adjusted by adjusting the data stream received by I/O module **2120**. In embodiments, the external phase controller **2202** provides control signals based on the phase difference between the receiver clock signal RXCLOCK and the transmitter clock signal TXCLOCK. In embodiments, the phase difference may be determined using phase detector **2206** and zero-delay buffers, **2208***a*, **2208***b*. In embodiments, the zero-delay buffers **2208***a*, **2208***b* may be incorporated into the FPGA core **2106** of the FPGA **2100**. In embodiments, FPGA core **2106** may not include the zero-delay buffers **2208***a*, **2208***b*. In embodiments, zero-delay buffer **2208***a* may have a reference trace T**1** whose length is matched to a length of wire T**3** between the I/O pin of FPGA **2100** and external phase detector **2206**. Similarly, zero delay buffer **2208***b* may have a reference trace T**2** whose length is matched to a second length of wire T**4** between the I/O pin of the FPGA **2100** and the external phase detector **2206**. Any length mismatch between reference traces T**1** and wire T**3** between the I/O pin of the FPGA **2100** and the external phase detector **2206** may introduce a deterministic phase error proportional to the mismatch of the length of the wires. Any length mismatch between reference traces T**2** and wire T**4** between the I/O pin of the FPGA **2100** and the external phase detector **2206** may introduce a deterministic phase error proportional to the mismatch of the length of the wires.

In embodiments, the transceiver PLL **2108** may include a second adjustable oscillator. In embodiments, the second adjustable oscillator of the transceiver PLL **2108** may provide the wire rate signal to the serializer **2110** which may be used to provide the transmitter clock signal TX CLOCK. In embodiments, the adjustable oscillator of the transceiver PLL **2108** may be provided outside of the transceiver PLL **2108** and operatively connected to the serializer **2110**. In embodiments, where a second adjustable oscillator is provided in the FPGA **2100**, the adjustable oscillator **2200** may not be necessary and the output of the controller **2202** may be provided to and used by the second adjustable oscillator to adjust the wire rate clock signal W.R. CLOCK provided to the serializer **2110** based on the information from the controller. In embodiments, where the second adjustable oscillator is provided on the FPGA **2100** and the adjustable oscillator **2200** is also provided off the FPGA, the delay of the second adjustable oscillator may be set to a constant value and the controller **2202** may provide instructions to the second adjustable oscillator based on this constant value. In embodiments, the constant value may be zero.

Using the configuration of FIG. **2**, the receiver clock signal RXCLOCK and the transmitter clock signal TXCLOCK are sufficiently aligned in phase such that there is no need for the clock domain crossing circuit discussed above, eliminating a technical problem. The technical solution to a technical problem, in embodiments, is shown in the system of FIG. **2**, which provides a wire or a logic path via the controller **2202** and adjustable oscillator **2200** to adjust the phase of the transmitter side clock TMCLOCK to correspond to that of the receiver side clock signal RXCLOCK. Some delays may be introduced in the path including min delay and max_delay constraints of the wire and components, however, such delays are insignificant compared to the larger delays that are necessarily present when an asynchronous clock domain crossing circuit is used.

In embodiments, zero delay buffers **2208***a*, **2208***b* may be configured to add a phase offset such that the receiver clock signal RXCLOCK and the transmitter clock signal TXCLOCK may be provided with any desired fixed phase offset with respect to each other. In such embodiments, any phase difference may be used, depending on the length of the logic path between the clocks and the corresponding delay constraints. In embodiments, with a phase difference between about 45° and 360°, the min delay should be set at 0 and the max_delay should be set equal to the time between rising edges on the receive clock signal and transmit clock signal (as shown in FIGS. **6**A and **6**B). The difference between 45° and 360° approximates the minimum amount of time required to safely capture data. In embodiments, the formula for the time is as follows:

$$t_{min}=t_{setup}+t_{wire}+t_{clock\text{-}out}+\Delta t_{jitter}+t_{skew} \quad \text{Equation 1.}$$

Referencing the above equation, t_{setup} is the setup time of a flip-flop (i.e. a latch) on FPGA **2100**. The t_{wire} is the delay of the, for example, shortest wire (as compared to other wires on the FPGA) on FPGA **2100**. In embodiments, the shortest wire is the shortest possible wire. In embodiments, $t_{clock\text{-}out}$ is the clock-to-output time of a flip-flop on the

FPGA **2100**. In embodiments, Δt_{jitter} is a safety factor for jitter on both clocks. Jitter, for example, is the deviation from true periodicity of a periodic signal from a reference clock. In embodiments, t_{skew} is the skew between clock paths to the two flip-flops on FPGA **2100**.

In embodiments where there is a phase difference with the transmitter clock running less than 45° ahead of the receiver clock, metastability may occur unless the clocks are treated as being greater than 360 degrees apart (as seen in FIG. **6**C). In such embodiments, the min and max delay are set with the min delay greater than the time difference between rising edges and the max delay less than one full clock cycle plus the min delay. In embodiments, the transmit clock signal may also sample on the opposite edge as the receiving clock signal launch edge, but cases are symmetric. In embodiments, a change from positive to negative edge sampling adds an approximate 180 degrees offset. In embodiments, manufacturer analysis tools both may add a reasonable jitter margin to account for times when jitter causes temporary misalignment.

In embodiments, to avoid instability, both the transceiver PLL **2108** and the zero-delay buffer PLL **2208***b* preferably run at the highest bandwidth available, while the overall phase locking system including the phase detectors **2206**, controller **2202** and oscillator **2200** runs at a low bandwidth. In such embodiments, the poles in the on-chip PLLs on the FPGA **2100** are prevented from affecting the gain or phase margin of the off-chip PLL which may be included in the controller **2202**. In embodiments, any loop filter order may be used in the controller **2202** as long as the bandwidth is low enough to avoid interaction of poles in the control system. In embodiments, a low bandwidth may be, for example, 100 Hz-40 kHz, 400 Hz and 4 khz, 4 kHz and 10 kHz, 100 hz to 40 kHz, 0 Hz and 100 Hz, to name a few. In embodiments, a second-order loop filter may be used in an effort to improve lock time and phase locking error. In embodiments a third-order filter may be used if the second-order filter is stable. In embodiments, a first-order filter is also an option when the second-order filter has stability problems. In embodiments, other order filters may be used, depending on the internal bandwidth of the FPGA **2100**.

In embodiments, where the bandwidth of the controller **2202** is too low, the receiving clock and the transmitter clock may not track each other closely enough to consider them locked. In embodiment, such tracking issues may be solved by adjusting the min and max delay constraints appropriately to provide a cushion for the clocks being away from the expected phase. In embodiments, a PD control system may be provided after the loop filter to aid in resolving tracking issues. In embodiments, tracking issues may be addressed by adding some nonlinearity in the transfer function of the filter. Other solutions may also be possible.

In the embodiment of FIG. **2**, the controller **2202** utilizes a phase-locked loop. In embodiments, a derivative component may be provided in the loop filter to slow down the control loop when the output value has changed recently and reduces the chance that the PLLs on the FPGA will lose lock. In embodiments, the derivative component's coefficient will be negative (i.e. the derivative component will act against the loop filter, however, some references refer to this as being a positive direction) and less than 1 (so the path straight from the filter dominates). The loop filter consists of a standard PLL loop filter with bandwidth between 400 Hz and 40 kHz of up to 4th order plus a derivative component that slows down the loop based on recent changes in the filter value. The bandwidth of the standard PLL loop filter, in some embodiments, may range between 400 Hz and 4 khz, 4 kHz and 10 kHz, 100 hz to 40 kHz, 0 Hz and 100 Hz, to name a few. The extra derivative component helps to keep the other components in the control system in their linear operating regions. The loop filter is represented by DigitalFilter( ) in the pseudocode provided.

FIG. **3**A is an exemplary block diagram of a deserializer suitable for use in the field programmable gate array system of FIG. **2** in accordance with an embodiment of the present invention. Deserializer **2104'**, in embodiments, may include an input buffer **3104**, a continuous time linear equalizer (CTLE) **3106**, a variable gain amplifier (VGA) **3108**; voltage adder **3110**; decision feedback equalizer (DFE) **3110**, clock and data recovery (CDR) **3116**, Serial to parallel box **3118**, decision circuit **3112**, and clock divider **3120**. In embodiments, deserializer **2104'** may not include one or more of the following components: CTLE **3106**, DFE **3114**, and/or VGA **3108**, to name a few.

A deserializer circuit accepts a data stream from an input buffer and converts it into a parallel format at a lower bit rate. First, data comes into the chip from the input pins **3102** into an input buffer **3104**. In embodiments, the input buffer receives a transmission from an I/O module, such as I/O module **2120**. A continuous time linear equalizer (CTLE) **3106** may be used after the input buffer **3104** to improve signal quality by placing a zero near the dominant pole of the transmission medium to cancel the first pole of the system. A variable-gain amplifier (VGA) **3108** may then be used to scale the signal up for further processing. This output is then turned into a binary 0-1 decision by the decision circuit **3112**. In embodiments, the decision circuit **3112** includes adder **3110**. A decision feedback amplifier may be included to further improve signal integrity by taking the results of the decision circuit, applying a weighting filter, and adding the result to the output from the VGA **3108**. When the signal is in the digital domain, a clock and data recovery circuit (CDR) **3116** recovers the phase of the data and create a wire-rate clock that is phase-aligned to the received data. The CDR **3116** may use a reference clock and a priori knowledge of the nominal frequency of the data stream to set the frequency of the recovered clock. With a clock aligned to the data stream, which in some embodiments may be a high speed data stream (e.g. 10 Gbps, 25 Gbps, a range from 9-25 Gbps, 10-25 Gbps, 22-33 Gbps, 33-45 Gbps, 45-60 Gbps, 60-80 Gbps, 80-120 Gbps, or 1 Gbps-100 Gbps, to name a few) the receiver can use a clock divider **3120** and a serial-to-parallel circuit **3118** to convert the serial data stream into a lower-frequency than the data stream (e.g. $\frac{1}{4}^{th}$ of the data rate of the high speed data stream or $\frac{1}{256}^{th}$ of the data rate of the high speed data stream, to name a few) parallel data stream.

In embodiments, a decision feedback equalizer **3114** (DFE) may be provided. In embodiments, the decision feedback equalizer may be a filter that subtracts or adds at least n decided bits to cancel inter-symbol interference on the wire. In embodiments, a clock and data recovery circuit may be provided and includes a data phase detector, which may be any of the circuits discussed above, as well as a phase-locked loop circuit.

FIG. **3**B is an exemplary circuit diagram of a continuous-time linear equalizer (CTLE) suitable for use in the deserializer of FIG. **3**A in accordance with an embodiment of the present invention. CTLE **3104'**, in some embodiments, places a zero to cancel out the dominant pole of the trans mission line for an incoming signal. In embodiments, CTLE **3104'** is a linear filter applied at a receiver, in the case of FIG. **3**B, input buffer **3104**, that attenuates low-frequency signal components and filters off higher frequencies. The CTLE

3104' may be adjusted to optimize the ratio of low frequency attenuation to high frequency amplification. In embodiments, a variable gain amplifier may be provided and used to scale the output of the CTLE to as large of a signal as possible. In embodiments, a decision circuit may also be provided.

FIG. 3C is an exemplary block diagram of a clock and data recovery (CDR) circuit suitable for use in the deserializer of FIG. 3A in accordance with an embodiment of the present invention. In embodiments, CDR 3118' circuit recovers phase information from an incoming data stream. The data stream, in embodiments, may be received by an alexander phase detector 3302. The CDR 3118 may also obtain frequency information from a reference clock, output. The output, may determine, in embodiments, whether the recovered clock have the same phase as the recovered data.

FIG. 3D is an exemplary circuit diagram of a decision circuit suitable 3110' for use in the deserializer of FIG. 3A in accordance with an embodiment of the present invention. FIG. 3D illustrates an exemplary embodiment of such a decision circuit 3110' utilizing a StrongARM sense amplifier. In embodiments, any comparator circuit may be used as a decision circuit.

FIG. 3E illustrates an exemplary embodiment of deserializer 2104". In embodiments, the deserializer 2104" may include a receiver PMA 3512 and receiver PCS 3508. The receiver PMA 3512 may include receiver buffer 3504, CDR 3516, and serial to parallel box 3118'. In embodiments, the receiver buffer 3504 receives serial differential input data 3502. In embodiments, this data is received by an I/O module, such as the I/O module 2120 of FIG. 2. In embodiments, the data is then received by the CDR 3516, from the receiver buffer 3504. The CDR, in embodiments, transmits the inputted serial data, a serial clock signal, and a parallel clock signal to the serial to parallel box 3506. The CDR may also transmits the parallel clock data to the receiver PCS 3508. The serial to parallel box 3118', in embodiments, then converts the serial data into parallel data, transmitting the parallel data to the receiver PCS 3508. The receiver PCS 3508 transmits parallel data to the FPGA Core 2106.

In addition to the components illustrated in FIG. 3E, deserializer 2104" may also include or be electrically coupled to a continuous-time linear equalizer (CTLE) (see FIG. 3B) which is a pure analog circuit that may be used to correct channel issues. In embodiments, the deserializer 2104" may include a shift register or demultiplexer with a counter.

FIG. 4A is an exemplary block diagram of a serializer suitable for use in the field programmable gate array system of FIG. 2 in accordance with an embodiment of the present invention. The serializer circuit accepts a clock running at wire rate 4112 (from transceiver PLL 2108) and accepts parallel data (e.g. TX data 4102) to serialize the data. Data is converted from a parallel format to a serial data stream in the parallel to serial box 4104. A feed-forward equalizer (FFE) 4106, which uses the history of bits serialized on the wire to filter out inter-symbol interference, may be included to improve signal integrity. Finally, the serializer outputs the serial data stream to a transmission medium using an output buffer 4114. The serializer also generates a parallel clock (TX Clk 4110), that operates at a lower speed than the data stream (e.g. 100-400 MHz, 100-600 MHz, to name a few), for use by other components using either a simple clock divider 4108 or a more complex clock generation block which can contain dividers and phase shifting elements for aligning multiple transmitters.

FIG. 4B illustrates an exemplary embodiment of serializer 2110". As discussed above with respect to FIGS. 2 and 4A, serializer 2110" receives parallel data streams from the FPGA core 2106 and serializes them for transmission out of the FPGA 2100. In embodiments, serializer 2110" may include transmitter PCS 4206, clock generation block 4210, and transmitter PMA 4216. In embodiments, transmitter PMA 4216 may include parallel to serial box 4104' and transmitter buffer 4214. The transmitter PCS 4206 may receive parallel data from FPGA core 2106. In embodiments, the parallel data is then transmitted to the parallel to serial box 4104'. The clock generation block 4210 may receive a wire-rate clock signal 4112. The clock generation block 4210, in embodiments, may split the signal in to a parallel clock signal and serial clock signal, both the parallel and serial signals being transmitted to the parallel to serial box 4104'. The parallel to serial box 4104 may covert the data from parallel to serial, transmitting the serial differential output data 4218 to the transmitter buffer 4214. Then, in embodiments, the transmitter buffer 4214 outputs the serial differential output data 4218.

In embodiments the serializer 2110" may be a shift register or a multiplexer with a counter. In embodiments, the serializer 2110" may include additional components, such as a feed forward equalizer (FFE) which uses the last n bits to cancel inter symbol interference on the wire. In embodiments, a buffer amplifier may be included which provides variable gain to allow adjustable voltage swing on the output of the transmitter. In embodiments, a clock divider may be provided. In embodiments, the clock divider may generate a low-speed clock (e.g. 100-400 MHz, 100-600 MHz, to name a few) for the serializer and the PMA/PCS. In embodiments, the clock divider is typically provided using a counter. In embodiments, the clock generation block 4210 might also include a clock phase adjustment circuit to allow multiple transmitter lanes to be phase-aligned.

Referring to FIGS. 5A-5D, in embodiments, the phase detector 2206 may be an analog, single bit phase detector which may be implemented using a variety of circuit components including (1) an XOR gate; (2) S-R flip-flop; (3) D flip-flop; (4) Edge-triggered JK flip-flop; (5) Gilbert cell multiplier and/or (6) Diode ring mixer, to name a few. FIG. 5A illustrates an exemplary embodiment of a phase detector 2206 that utilizes D flip-flops. FIG. 5B illustrates another exemplary embodiment of a phase detector 2206 that utilizes S-R flip-flops. FIG. 5C is another exemplary block diagram of a phase detector suitable for use in the field programmable gate array system of FIG. 2 in accordance with an embodiment of the present invention. FIG. 5D is another exemplary block diagram of a phase detector suitable for use in the field programmable gate array system of FIG. 2 in accordance with an embodiment of the present invention. Other than a basic phase detector, a time-to-digital converter (TDC) can be used to detect the phase offset of the clocks. FIG. 5C shows a Vernier TDC and FIG. 5D shows a parallel delay line TDC as examples. The Vernier TDC uses a pair of unequal delay lines to delay the two signals by varying amounts and uses D-type flip-flops to quantify the phase difference between the two signals at multiple points along the delay line. The parallel delay line TDC uses a set of simple phase detectors with the two input signals delayed by varying amounts to quantify the phase difference between the two signals. In both cases, a decoding circuit is used to translate the results of the individual phase detection elements into a phase offset.

In embodiments, the phase detector 2206 may be a multi-bit phase detector, sometimes referred to as a time-to

digital converter (TDC). In embodiments, such a phase detector may be a counter-based TDC that counts up the time between START and STOP. In embodiments, such a phase detector may be an interpolator TDC in which time measurement may be performed by measuring a voltage difference (e.g. by using a ramp wave and sampling at two points or by charging a capacitor). In embodiments, a phase detector may be a vernier TDC. FIG. **5**C

In embodiments the phase detector **2206** may be implemented using a tapped delay line TDC. In embodiments, the phase detector **2206** may be implemented using a metastability-based phase detector. In embodiments, the metastability-based phase detector is similar to a D flip-flop phase detector but includes an averaging filter. In embodiments, a parallel-delay-line TDC may be used to implement the phase detector **2206**. FIG. **5**D illustrates an example of a parallel-delay-line TDC. In embodiments, the phase detector may be implemented as a pulse-shrinking TDC. Such a pulse-shrinking TDC may generate a pulse and then send it circularly through an engineered delay line to shrink it until it is gone, count the number of times it goes through the pulse-shrinking circuit.

In embodiments, the phase detector may be implemented as a scrambling TDC. Scrambling TDCs may be implemented with added digital noise shaping to suppress errors. In embodiments, the phase detector **2206** may be implemented as a PLL/DLL-based TDC. In such an embodiment, the phase detector may lock a digital PLL to the incoming feedback signal and record the internal control code.

In embodiments, all of the phase detector embodiments discussed above may be combined with or connected to a low-pass filter and an analog to digital converter (ADC). In embodiments, other examples of phase detectors may be used consistent with the teachings of this disclosure.

FIGS. **6**A-**6**D illustrate example phase differences between the receiver side clock and the transmitter side clock. It will be understood that other examples of phase differences may be present consistent with the teachings of this disclosure. Generally, when the clocks are aligned, the minimum and maximum delays may be used to determine the time available for the computation of data.

FIG. **6**A illustrates an example of a phase difference between the receiver side clock and the transmitter side clock of an FPGA in accordance with an embodiment. FIG. **6**A depicts a phase difference when the clocks are under 360 degrees apart. This embodiment results in minimum delay **6106**A and maximum delay **6108**A, for which data may be computed. The minimum delay **6106**A is from the driving clock edge back to itself, as shown in FIG. **6**A.

FIG. **6**B illustrates an example of a phase difference between receiver clock and a transmitter clock of an FPGA in accordance with an embodiment. The phase difference depicted in FIG. **6**B shows a phase difference with minimum delay **6106**B and maximum delay **6108**B. The example shown in FIG. **6**B shows a minimum delay **6106**B of **0**. However, in practice, while designers may design for the phase difference in FIG. **6**B, the phase difference may end up more similar to the phase differences of FIGS. **6**A and/or **6**C.

FIG. **6**C illustrates an example of a phase difference between receiver clock and a transmitter clock of an FPGA in accordance with an embodiment. FIG. **6**C depicts a phase difference with the clocks are over 360 degrees apart. The time between minimum delay **6106**C and maximum delay **6108**C may allow for more data computations within the phase difference.

FIG. **6**D illustrates an example of an inverted phase difference between receiver clock and a transmitter clock of an FPGA in accordance with an embodiment. This example of an inverted phase difference between minimum delay **6106**D and maximum delay **6108**D shows that, even inverted, the phase difference would work similarly to the phase detectors of FIGS. **6**A-**6**C.

FIG. **7**A is a block diagram of another field programmable gate array system in accordance with an embodiment of the present invention. The system of FIG. **7**A is similar to that of FIG. **2**, and common components are referred to using the same reference numbers (with a differing first digit to indicate the Figure—e.g. if the first digit of a reference numeral is "7," the reference numeral is located on FIG. **7**) as in FIG. **2**. One difference between FIG. **2** and FIG. **7**A is FIG. **7**A shows multiple deserializers **7104***a*, **7104***b*, and multiple serializers **7110***a*, **7110***b*. Multiple deserializers and serializers, as opposed to one of each, allows FPGA **7100** to receive more input data and transmit more output data.

The system, in accordance with FIG. **7**, may include FPGA **7100**. FPGA **7100** may include FPGA transceiver banks **7102** and FPGA core **7106**. The FPGA transceiver banks **7102** may include multiple serializers and deserializers, such as deserializers **7104***a*,**7104***b*, and serializers **7110***a*, and **7110***b*. FPGA core **7106** may include zero-delay buffers **7208***a*, **7208***b*, and **7208***c* and computation logic **7210**. The system may further include a fixed-frequency oscillator **7000**, an I/O module **7120** connected to an external connection **7122**, an adjustable delay line **7200***a*, a frequency or phase-adjustable oscillator **7200***b*, a DLL controller **7202***a*, a PLL/DLL controller **7202***b*, and phase detectors **7206***a*, **7206***b*.

In a multi-channel system, phase alignment can be achieved by using multiple serializers in a channel bonded mode, which keeps their parallel and serial clocks running at the same phase, and by using an adjustable delay line **7200***a* (along with an additional controller **7202***a*) on the data streams going to all but one of the deserializers. The adjustable delay line **7200***a* may have adjustment range at least as wide as one period of the parallel RX clock. This can be accomplished by using a long delay line or by using a pair of smaller delay lines with a glitch-free switchover circuit to hide the boundary conditions of the delay lines.

Similar to the system described in connection with FIG. **2**, the system of FIG. **7**A may or may not include zero delay buffers **7208***a*, **7208***b*, and **7208***c*. In embodiments, zero-delay buffer **2208***a* may have a reference trace T**1** whose length is matched to a length of wire T**4** between the I/O pin of FPGA **2100** and external phase detectors **7206***a*, **7206***b*. Similarly, zero delay buffer **2208***b* may have a reference trace T**2** whose length is matched to a second length of wire T**5** between the I/O pin of the FPGA **7100** and the external phase detector **7206***a*. Also, similarly, zero delay buffer **2208***c* may have a reference trace T**3** whose length is matched to a second length of wire T**6** between the I/O pin of the FPGA **7100** and the external phase detector **7206***b*. Any length mismatch between reference traces T**1** and wire T**4** between the I/O pin of the FPGA **7100** and the external phase detectors **7206***a*, **7206***b* may introduce a deterministic phase error proportional to the mismatch of the length of the wires. Any length mismatch between reference traces T**2** and wire T**5** between the I/O pin of the FPGA **7100** and the external phase detector **7206***a* may introduce a deterministic phase error proportional to the mismatch of the length of the wires. Any length mismatch between reference traces T**3** and wire T**6** between the I/O pin of the FPGA **7100** and the external phase detector **7206***b* may introduce a deterministic phase error proportional to the mismatch of the length of the wires.

The FPGA **7100**, and all the components within the system shown in FIG. **7**A, may be similar to FPGA **2100** and all of the components within the system of FIG. **2**, the descriptions of which applying herein.

FIG. **7**B is a block diagram of another field programmable gate array system in accordance with an embodiment of the present invention. The system of FIG. **7**B is similar to that of FIGS. **2** and **7**A, and common components are referred to using the same reference numbers (with a differing first digit to indicate the Figure—e.g. if the first digit of a reference numeral is "7," the reference numeral is located on FIG. **7**) as in FIG. **2**. XCVR PLL **7108** stands for Transceiver PLL **7108**. Similar to the system shown in FIG. **7**A, the system of FIG. **7**B has multiple deserializers **7104***a-d* and multiple serializers **7110***a-d*. As noted above, multiple deserializers and serializers, as opposed to one of each, allows FPGA **7100'** to receive more input data and transmit more output data.

The FPGA **7100'**, and all the components within the system shown in FIG. **7**B, may be similar to FPGA **2100** and all of the components within the system of FIG. **2**, the descriptions of which applying herein. The FPGA **7100'**, and all the components within the system shown in FIG. **7**B, may be similar to FPGA **7100** and all of the components within the system of FIG. **7**A, the descriptions of which applying herein.

FIG. **7**C is a block diagram of another field programmable gate array system in accordance with an embodiment of the present invention. The system of FIG. **7**C is similar to that of FIGS. **2** and **7**A, and common components are referred to using the same reference numbers (with a differing first digit to indicate the Figure—e.g. if the first digit of a reference numeral is "7," the reference numeral is located on FIG. **7**) as in FIG. **2**. XCVR PLL **7108** stands for Transceiver PLL **7108**. Similar to the system shown in FIGS. **7**A and **7**B, the system of FIG. **7**C has multiple deserializers **7104***a-f* and multiple serializers **7110***a-f*. As noted above, multiple deserializers and serializers, as opposed to one of each, allows FPGA **7100"** to receive more input data and transmit more output data.

The FPGA **7100"**, and all the components within the system shown in FIG. **7**C, may be similar to FPGA **2100** and all of the components within the system of FIG. **2**, the descriptions of which applying herein. The FPGA **7100"**, and all the components within the system shown in FIG. **7**C, may be similar to FPGA **7100** and all of the components within the system of FIG. **7**A, the descriptions of which applying herein.

An exemplary prototype for this system is shown in FIGS. **8**A and **8**B. In this embodiment, the prototype is a VITA **57.1** FMC expansion card **8410** with an SFP+ connector for 10 gigabit Ethernet **8120** and the required hardware for a PLL approach to the phase locking problem. In this embodiment, the VITA **57.1** FMC connector **8300** attaches the expansion card to a host FPGA card **8400**, for example. In this embodiment, adjustments to the transmitter side clock signal may be based on frequency or phase adjustment provided via the SI571 VXCO **8200**. Control of these elements may be provided via the ATSAML21 Microcontroller **8202***b*. Phase detection may be provided by the phase detector **8206** which may provide phase difference information to the microcontroller via the filter **8202***a*. A receiving side reference clock may be provided by the **51570** crystal oscillator **8000**. In this embodiment, control loop filtering is done in software using the microcontroller, except for an anti-aliasing filter at half the sample rate of the microcontroller ADC. In embodiments, the high-speed transceiver channels may have a data stream of 8-12 Gbps.

In the embodiment of FIGS. **8**A and **8**B, the following pseudocode may be used to implement an exemplary embodiment of the system discussed above in connection with FIG. **2**.

The pseudo code shown below implements the control loop and communicates with the host. In embodiments, the pseudocode shown is run by the microcontroller (**8202***b*) and uses an interrupt-based system to run the control algorithm for either the PLL or DLL paths on the device. The pseudo-code also includes a lock indicator and a memory-mapped I^2C interface for communication with the host system. The main control loop is triggered on every sample of the internal ADC.

```
Begin:
  Set up peripherals
  Program Si5340 with configuration
  Enable Analog Comparator to measure lock indicator from the phase detector
  Set ADC in free-running mode and enable interrupts on every sample
  Enable I2C interrupt
  Wait
On ADC Interrupt:
  Read ADC value into Window[0]
  ControlCode=DigitalFilter(Window)
  Read MODE pin to Mode
  If Mode=PLL MODE then:
  Output ControlCode on DAC channel 0
  Else:
  Coarse=9 MSBs of ControlCode
  Fine=NonlinearCorrection(ControlCode-Coarse)
  Output Coarse to DELAY_CONTROL pins
  Output Fine to DAC channel 1
  Shift Window array by 1
On I2C Packet Received:
  Address=I2C_packet[0]
  Length=I2C_packet.length
  If I2C READ then:
  I2C write(I2C regs[Address])
  Else:
  for i<Length:
    I2C regs[Address+i]=I2C_packet[1+i]
On Analog Comparator Interrupt:
  Read comparator value to Ind
  Output Ind to LOCK pin
```

FIGS. **9**A-C are exemplary flow charts of a process implementing the field programmable gate array system in accordance with an embodiment of the present invention. The field programmable gate array system, including the field programmable gate array and the components thereof, may be similar to the field programmable gate array system of FIGS. **2**, **7**A-**7**C and **8**A-**8**C, the descriptions of which applying herein.

The process of FIGS. **9**A-**9**C may begin at step S**902**. Referring to FIG. **9**A, at step S**902**, a first serial data stream is received by a first plurality of data pins in a first interface of a field programmable gate array in the field programmable gate array system. In embodiments, the first serial data stream may include market data. In embodiments, market data may be price and trade-related data for a financial instrument reported by a trading venue (i.e. a stock exchange).

At a step S**904**, a first clock signal having a first frequency and a first phase is received by a first reference clock in in the first interface. In embodiments, step S**904** may be

performed before step S902. In embodiments step S904 may be performed contemporaneously with step S902.

The process of FIGS. 9A-9C may continue with a step S906. At step S906, the first plurality of data pins in the first interface may transmit the first serial data stream to a deserializer in the field programmable gate array.

At a step S908, the first reference clock pin in the first interface may transmit the first clock signal to the deserializer. In embodiments, step S908 may be performed before step S906. In embodiments step S908 may be performed contemporaneously with step S906.

At a step S910, a first receiver side clock signal may be generated by the deserializer. In embodiments, the first receiver side clock signal may have a second frequency and a second phase. In embodiments, the second frequency is different from the first frequency. In embodiments, the second frequency corresponds to the first frequency. In embodiments, corresponding may refer to the frequencies being the same frequency. In embodiments corresponding may also refer to frequencies being close (e.g. within an acceptable range in the art) to the same frequencies. In embodiments, the first phase and the second phase may not be aligned. In embodiments the first phase and the second phase are aligned. In embodiments, a difference between the first phase and the second phase is below a threshold level. In embodiments, the deserializer generates the first receiver clock signal based at least in part on the first clock signal.

At a step S912, the first serial data stream received by the deserializer may be converted, by the deserializer, into a first plurality of data streams. In embodiments, the first plurality of data streams may include a first amount of data streams. In embodiments, the first amount of data streams is equal to eight (8) data streams. In embodiments, the first amount of data streams is equal to ten (10) data streams. In embodiments, the first amount of data streams is equal to sixteen (16) data streams. In embodiments, the first amount of data streams is equal to twenty (20) data streams. In embodiments, the first amount of data streams is equal to thirty-two (32) data streams. In embodiments, the first amount of data streams is equal to forty (40) data streams. In embodiments, the first amount of data streams is equal to sixty-four (64) data streams. In embodiments, the first amount of data streams is equal to eighty (80) data streams. In embodiments, the first amount of data streams is equal to one hundred twenty-eight (128) data streams. In embodiments, the first amount of data streams is equal to one hundred sixty (160) data streams. In embodiments, the first amount of data streams is equal to two (2) to the power of N, where N is an integer. In embodiments, the first amount of data streams is equal to ten (10) times two (2) to the power of N, where N is an integer. In embodiments the first amount of data streams is equal to thirty-three (33) times two (2) to the power of N, where N is an integer.

At a step S914, the first receiver side clock signal is transmitted from the deserializer to computational circuitry in the field programmable gate array.

At a step S916, the first plurality of data streams is transmitted form the deserializer to the computational circuitry. In embodiments, the first plurality of data streams includes a first plurality of data items. In embodiments, step S916 may be performed before step S914. In embodiments step S916 may be performed contemporaneously with step S914.

At a step S918, the first receiver side clock signal is transmitted from the deserializer via a first clock output pin on a second interface of the field programmable gate array to a phase detector of the field programmable gate array system which is not on the field programmable gate array. In embodiments, the transmission of the first receiver side clock signal is also via a first zero delay buffer phase lock loop of the field programmable gate array before being transmitted via the first clock output pin of the second interface.

At a step S920, a first transmitter side clock is generated by the field programmable gate array system. In embodiments, the first transmitter side clock has a third frequency and a third phase. In embodiments, the third frequency corresponds to the second frequency. In embodiments the third phase may be aligned with the second phase. In embodiments, the third phase and the second phase may not be aligned. In embodiments, the third phase and the second phase may have a difference in phases that is less than a second threshold level. In embodiments, the first transmitter side clock signal is generated by performing the following steps until a first output of the phase detector is below a first threshold.

Referring to FIG. 10A, generating the first transmitter side clock signal may begin at step S1002. At step S1002, a second clock signal is generated by an adjustable oscillator in the field programmable gate array system. In embodiments, the second clock signal may have a fourth frequency and a fourth phase.

The process of FIGS. 10A-B may continue at a step S1004. At step S1004, a first wire rate clock signal having a fifth phase and a fifth frequency is generated. In embodiments, the generation of the wire rate clock signal may be based in part on the second clock signal. In embodiments the fifth frequency is a rational multiple of the fourth frequency. In embodiments the fifth frequency is equal to x times the fifth frequency where x is the second amount of data streams. In embodiments the fifth frequency is equal to x times the frequency of the of the transmitter side clock signal where x is half of the second amount of data streams.

In embodiments, the generation of the wire rate clock signal may be performed with the following steps. Referring to FIG. 11, at a step S1102, the second clock signal is received from the adjustable oscillator at a transceiver phase lock loop via a second reference clock pin in the first interface. At a step S1104, the second clock signal is processed by the transceiver phase lock loop. As a step S1106, the wire rate clock signal is provided by the transceiver phase lock look. The providing of the wire rate clock signal, in embodiments, may be based at least in part on the processing of the received clock signal. Once the wire rate clock signal is provided, at a step S1108. the wire rate clock signal is transmitted by the transceiver phase lock loop to a serializer on the field programmable gate array. In embodiments, the first wire rate clock is generated by a phase lock loop. In embodiments, the phase lock loop that generates the wire clock is on field programmable gate array. In embodiments, the phase lock loop that generates the wire clock is part of the field programmable gate array system and not part of the field programmable gate array.

In embodiments, the generation of the wire rate clock may include processing the second clock signal. Once the second clock signal is processed, the wire rate clock may be provided. The providing of the wire rate clock, in embodiments, is based at least in part on the processing of the second clock signal. After providing the wire rate clock, in embodiments, the wire rate clock may be transmitted to the serializer.

Referring back to FIG. 10A, at a step 1006, an interim transmitter side clock signal having a sixth frequency and a sixth phase is generated by the serializer in the field pro-

grammable gate array. In embodiments, the sixth frequency corresponds to the third frequency. In embodiments the sixth phase may be aligned with the third phase. In embodiments, the sixth frequency may correspond to the third frequency and the sixth phase may not be aligned with the third phase. In embodiments, the sixth phase and the third phase may not be aligned. In embodiments, the third phase and the sixth phase may have a difference in phases that is less than a second threshold level.

At a step **S1008**, the interim transmitter side clock signal is transmitted from the serializer via a second clock output pin in the second interface to the phase detector. In embodiments, the transmission of the interim transmitter side clock is also via a second zero delay buffer phase lock loop of the field programmable gate array before being transmitted via the second clock output in of the second interface.

The process of FIGS. **10**A-**10**B may continue with at a step **S1010**. Referring to FIG. **10**B, at a step **S1010**, a first output is generated by the phase detector. The first output, in embodiments, is based at least on a comparison of the first receiver side clock signal and the interim side clock signal. The first output, in embodiments, may include information indicating the phase differences (if there is any) between the first receiver side clock and the interim side clock signal. Additionally, in embodiments, the first output may include information indicating a difference (if there is any) of frequencies between the first receiver side clock signal and the interim side clock signal.

At a step **S1012**, the first output may be transmitted from the phase detector to a phase controller of the field programmable gate array system which is not in the field programmable gate array.

At a step **S1014**, interim adjustment information is determined by the phase controller. In embodiments, the interim adjustment information may be based at least in part on the first output. For example, the interim adjustment information may indicate necessary adjustments to align the phases of the first receiver side clock signal and the interim side clock signal. As another example, the interim adjustment information may indicate necessary adjustments to align the frequencies of the first receiver side clock signal and the interim side clock signal.

At a step **S1016**, the interim adjustment information is transmitted from the phase controller to the adjustable oscillator. In embodiments, once the interim adjustment information is received, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information. The adjustment of the second clock signal, in embodiments, may change the phase of the second clock signal. In embodiments, the interim adjustment information indicates a desired phase. In embodiments, the interim adjustment information indicates a desired frequency. In embodiments, the interim adjustment information indicates a change in phase. In embodiments, the interim adjustment information indicates a change in frequency. In embodiments, the adjustment information includes a voltage. In embodiments, the adjustment information includes a digital transmission. For example, the adjustment information may be a command over a serial bus (e.g. I^2C or SPI) that adjusts a clock divider or phase rotator. As another example, the adjustment information may trigger a set of parallel digital wires for "frequency/phase up," "frequency/phase down," and/or "frequency/phase step." In embodiments, the adjustment of the second clock signal may change the frequency of the second clock signal.

In embodiments, steps **S1002** through steps **S1016** are repeated until the first output of the phase detector is below the first threshold level. This threshold level may indicate that the first receiver side clock signal and the interim transmitter side clock signal have phases that are aligned. This threshold level may indicate that the first receiver side clock signal and the interim transmitter side clock signal have frequencies that correspond to one another.

Referring back to FIG. **9**B, once the first output of the phase detector is below a first threshold level, at a step **S922**, the serializer may transmit the first transmitter side clock signal to the computational circuitry. At a step **S924**, a first set of operations ins performed by the computational circuitry on at least a portion of the first plurality of data items to generate a second plurality of data streams. In embodiments, the first operation includes a trading algorithm. In embodiments the trading algorithm may include the parsing the market data. Once the market data is parsed, mathematical operations are performed at a portion of the market data. After the mathematical operations are performed, order packets are generated using at least an output of the mathematical operations. In embodiments, the first set of operations includes at least one of the following: (i) an arithmetic operation; (ii) a logical operation; (iii) a pipeline operation; and (iv) a memory access operation.

In embodiments, at least a portion of the first set of operations is performed prior to step **S922**. In embodiments, at least a portion of the first set of operations is performed after step **S922**. In embodiments all of the first set of operations is performed after step **S922**.

In embodiments, the second plurality of data streams may include a second amount of data streams. In embodiments, the second amount of data streams may be the same as the first amount of data streams. In embodiments, the second amount of data streams is equal to eight (8) data streams. In embodiments, the second amount of data streams is equal to ten (10) data streams. In embodiments, the second amount of data streams is equal to sixteen (16) data streams. In embodiments, the second amount of data streams is equal to twenty (20) data streams. In embodiments, the second amount of data streams is equal to thirty-two (32) data streams. In embodiments, the second amount of data streams is equal to forty (40) data streams. In embodiments, the second amount of data streams is equal to sixty-four (64) data streams. In embodiments, the second amount of data streams is equal to eighty (80) data streams. In embodiments, the second amount of data streams is equal to one hundred twenty-eight (128) data streams. In embodiments, the second amount of data streams is equal to one hundred sixty (160) data streams. In embodiments, the second amount of data streams is equal to two (2) to the power of N, where N is an integer. In embodiments, the second amount of data streams is equal to ten (10) times two (2) to the power of N, where N is an integer. In embodiments the second amount of data streams is equal to thirty-three (33) times two (2) to the power of N, where N is an integer.

At a step **S926**, the second plurality of data streams is transmitted from the computational circuitry to the serializer. Once received by the serializer, at a step **S928**, the serializer converts the second plurality of parallel data streams into the second serial data stream. The second serial data stream, in embodiments, includes trading data. Trading data, for example, may be any data related to purchasing or selling of stocks, commodities, goods, and/or services.

At a step **S930**, the second serial data stream may be transmitted from the serializer off the field gate programmable array system via a second plurality of data pins in the first interface. In embodiments, the second serial data stream is transmitted from the serializer on the field programmable

gate array to an input/output module of the field gate array system but not on the field gate array. The second serial data stream may then be transmitted from the input/output module off the field gate array system.

EXAMPLES

The following examples may be used to illustrate embodiments of the present invention. They are meant solely for illustration and not intended to be limiting.

Example 1

In embodiments a trading algorithm may be applied in a crossing auction on an option exchange. In embodiments, a third serial data stream including target price information on a set of options contracts is provided as in input to the FPGA. In embodiments, the first serial stream includes market data includes bid, asks, trades and auction notifications. When an auction is announced, as indicated in the market data, the FPGA compares the announced auction price to its target price. In embodiments, the target price is stored in the FPGA and is provided via the third serial data stream. In embodiments, the target price may be generated using a suitable pricing model, e.g. Black Sholes. When there is some overlap, the field programmable gate array generates and sends an order at its target price which is transmitted out of the FPGA in the second serial data stream. In embodiments, the target price information on a set of options contracts is streamed into the field programmable gate array as a simple set of triggers in the third serial data stream. The target price information, may include a linearization of the result of the Black Scholes computation or some other model. The field programmable gate array may implement Black-Scholes, and instead receive market data on the underlying asset as a secondary data stream.

Example 2

Another example of a trading algorithm is "signal based". In embodiments, signals refer to a trigger that is activated when certain events happen. In embodiments, a signal may simply track a sale price which changes every time the market information indicates a sale has taken place. In embodiments, a signal may be used as a measure of book pressure to determine whether prices are likely to rise or fall. Some signals may be calculated quickly while others are calculated more slowly because they require more data over a longer period of time. In embodiments, order information may be based on a single signal or based on a plurality of signals. Signals depending on the historical behavior of a financial instrument are programmed into the field programmable gate array. A few simple and well-known examples of a signal here are moving averages of past prices or book pressure signals (looking for an imbalance in the order book). Signal-based trading involves receiving market data on the financial instrument, constructing the order book for that instrument, calculating a predicted price based on a pre-defined collection of signals, and sending an order if the signals indicate that the signal indicates that it is a favorable time to trade. A secondary data stream here could be used to enable or disable signals or to change signal parameter values. In embodiments, the secondary data stream may be omitted. An example of a signal-based trading system might be a system that updates the quotes for a market maker when a set of signals indicates that a large directional move in the price of a stock is coming. The parallel processing advantage shows up here more so than in the first example.

Now that embodiments of the present invention have been shown and described in detail, various modifications and improvements thereon can become readily apparent to those skilled in the art. Accordingly, the exemplary embodiments of the present invention, as set forth above, are intended to be illustrative, not limiting. The spirit and scope of the present invention is to be construed broadly.

What is claimed is:

1. A method for processing a first serial data stream comprising market data, using a field programmable gate array system, to generate a second serial data stream comprising order entry data, wherein the method comprises the steps of:

a) receiving, by a first plurality of data pins in a first interface of a field programmable gate array in the field programmable gate array system, the first serial data stream;

b) receiving, by a first reference clock pin in the first interface, a first clock signal having a first frequency and a first phase;

c) transmitting, from the first plurality of data pins in the first interface to a deserializer in the field programmable gate array, the first serial data stream;

d) transmitting, from the first reference clock pin in the first interface to the deserializer, the first clock signal;

e) generating, by the deserializer, a first receiver side clock signal having a second frequency and a second phase, based on the first clock signal;

f) converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams;

g) transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first receiver side clock signal;

h) transmitting, from the deserializer to the computational circuitry, the first plurality of data streams comprising a first plurality of data items and a first amount of data streams;

i) transmitting, from the deserializer via a first clock output pin on a second interface of the field programmable gate array to a phase detector that is part of the field programmable gate array system and not part of the field programmable gate array, the first receiver side clock signal;

j) generating, using the field programmable gate array system, a first transmitter side clock signal having a third frequency and a third phase, by performing the following steps until a first output of the phase detector is below a first threshold level:

i. generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal having a fourth frequency and a fourth phase;

ii. generating a first wire rate clock signal having a fifth frequency and a fifth phase based on the second clock signal;

iii. generating, by a serializer in the field programmable gate array, an interim transmitter side clock signal having a sixth frequency and a sixth phase;

iv. transmitting, from the serializer via a second clock output pin in the second interface to the phase detector, the interim transmitter side clock signal;

v. generating, by the phase detector, the first output based on a comparison of the first receiver side clock signal and the interim transmitter side clock signal;

vi. transmitting, from the phase detector to a phase controller of the field programmable gate array system which is not in the field programmable gate array, the first output;

vii. determining, by the phase controller, interim adjustment information based on the first output; and

viii. transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;

wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information and steps (i) through (viii) are repeated until the first output of the phase detector is below the first threshold level;

k) transmitting, from the serializer to the computational circuitry, first transmitter side clock signal;

l) performing, by the computational circuitry, a first set of operations on at least a portion of the first plurality of data items to generate a second plurality of parallel data streams comprising a second amount of data streams;

m) transmitting, from the computational circuitry to the serializer, the second plurality of parallel data streams;

n) converting, by the serializer, the second plurality of parallel data streams into the second serial data stream; and

o) transmitting, from the serializer off the field gate programmable array system via a second plurality of data pins in the first interface, the second serial data stream.

2. The method of claim **1**, wherein the second frequency corresponds to the first frequency.

3. The method of claim **1**, wherein the second phase is not aligned with the first phase.

4. The method of claim **1**, wherein the second frequency is different from the first frequency.

5. The method of claim **1**, wherein the transmission of the first receiver side clock signal from the deserializer to the phase detector is also via a first zero delay buffer phase lock loop of the field programmable gate array before being transmitted via the first clock output pin of the second interface.

6. The method of claim **1**, wherein the third frequency corresponds to the second frequency.

7. The method of claim **1**, wherein the third phase is aligned with the second phase.

8. The method of claim **1**, wherein a difference between the third phase and the second phase is less than a second threshold level.

9. The method of claim **1**, wherein the sixth frequency corresponds to the third frequency.

10. The method of claim **9**, wherein the sixth phase is not aligned with the third phase prior to the first output of the phase detector being below the first threshold level.

11. The method of claim **1**, wherein the transmission of the interim transmitter side clock signal from the serializer to the phase detector is also via a second zero delay buffer phase lock loop of the field programmable gate array before being transmitted via the second clock output pin of the second interface.

12. The method of claim **1**, wherein the first set of operations does not include clock domain crossing operations that delays processing of the first set of parallel data streams.

13. The method of claim **1**, wherein the first amount of data streams is the same as the second amount of data streams.

14. The method of claim **13**, the first amount of data streams and the second amount of data streams is one of the following:

(i) eight (8) data streams;

(ii) ten (10) data streams;

(iii) sixteen (16) data streams;

(iv) twenty (20) data streams;

(v) thirty-two (32) data streams;

(vi) forty (40) data streams;

(vii) sixty-four (64) data streams; and

(viii) eighty (80) data streams;

(ix) one hundred twenty-eight (128) data streams; and

(x) one hundred sixty (160) data streams.

15. The system of claim **13**, wherein the first amount of data streams and the second amount of data streams are equal to two (2) to the power of N, where N is an integer.

16. The system of claim **13**, wherein the first amount of data streams and the second amount of data streams are equal to ten (10) times two (2) to the power of N, where N is an integer.

17. The method of claim **14**, wherein the first amount of data streams and the second amount of data streams are equal to thirty-three (33) times two (2) to the power of N, where N is an integer.

18. The method of claim **1**, wherein at least a portion of the first set of operations is performed prior to step (k).

19. The method of claim **1**, wherein at least a portion of the first set of operations is performed after step (k).

20. The method of claim **1**, wherein all of the first set of operations is performed after step (k).

21. The method of claim **1**, wherein the transmission of the second serial data stream from the serializer off the field programmable gate array is transmitted to an input/output module off the field programmable gate array.

22. The method of claim **1**, wherein the first serial data stream includes market data and the second serial data stream includes trading data.

23. The method of claim **1**, wherein the step of generating the first wire rate clock signal further includes:

(a) receiving, from the adjustable oscillator at a transceiver phase lock loop via a second reference clock pin in the first interface, the second clock signal;

(b) processing, by the transceiver phase lock loop, the received second clock signal;

(c) providing, by the transceiver phase lock loop, the wire rate clock signal based on the processing of the received clock signal; and

(d) transmitting, by the transceiver phase lock loop, the wire rate clock signal to the serializer.

24. The method of claim **1**, wherein a phase lock loop generates the first wire rate clock signal.

25. The method of claim **24**, wherein the phase lock loop is on the field programmable gate array.

26. The method of claim **24**, wherein the phase lock loop that is part of the field programmable gate array system and not part of the field programmable gate array.

27. The method of claim **1**, the step of generating the first wire rate clock signal further includes:

(a) processing the second clock signal;

(b) providing the wire rate clock signal based on the processing of the second clock signal; and

(c) transmitting, to the serializer, the wire rate clock signal.

28. The method of claim **1**, wherein the sixth frequency corresponds to the third frequency.

29. The method of claim 28, wherein the sixth phase is not aligned with the third phase.

* * * * *