

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

OPTIVER US LLC and OPTIVER TRADING US LLC,
Petitioners

v.

HFT SOLUTIONS, LLC,
Patent Owner.

Case No. IPR2026-00383
U.S. Patent No. 12,107,587

Corrected Declaration of Dr. B. Carrion Schaefer

Table of Contents

TABLE OF EXHIBITS.....	9
TABLE OF ABBREVIATIONS	10
I.—BACKGROUND AND QUALIFICATIONS.....	12
II.—SCOPE OF OPINIONS	19
III.—MATERIALS REVIEWED	19
IV.—Person of Ordinary Skill in the Art.....	20
V.—Statement of Legal Principles.....	20
A.—Legal Standard for Prior Art.....	20
B.—Anticipation and Prior Art	21
C.—Legal Standard for Obviousness	22
D.—Legal Standard for Claim Construction	28
VI.—Summary of the '587 Patent.....	31
A.—Challenged Claims.....	32
B.—File History of the '587 Patent	32
VII.—Application of the Prior Art to the CHALLENGED Claims	33
A.—Priority Date	33
B.—Claim Construction.....	34
C.—Overview of Prior Art.....	34
1.—R. Giordano and A. Aloisio, <i>Fixed-Latency, Multi-Gigabit Serial Links with Xilinx FPGAs</i> , 58-1 IEEE Transactions on Nuclear Science 194 (2011) (“Giordano”) (Ex. 1005).....	34
a)—Ex. 1009, Virtex-UG (incorporated by Giordano).....	36
2.—Virtex-5 FPGA RocketIO-GTP Transceiver User Guide, UG196 (v2.1), December 3, 2009 (“Virtex-UG”) (Ex. 1009)	37
3.—U.S. Pat. No. 7,499,513 (“Tetzlaff”) (Ex. 1006).....	38
4.—U.S. Pat. No. 10,447,463 (“Raymond”) (Ex. 1007).....	39
5.—U.S. Pat. No. 8,788,553 (“Lee”) (Ex. 1008)	41
VIII.—State of the Art.....	42

A. ~~Serialization and Deserialization (SERDES) in FPGAs~~	43
B. ~~Phase Locked Loops (PLLs) and Delay Locked Loops (DLLs) in FPGAs~~	45
C. ~~Clock Synchronization and Phase Alignment Between Receiver and Transmitter~~	47
D. ~~Application to Financial Trading Systems~~	50
E. ~~Summary of State of the Art~~	51
IX. ~~Detailed Analysis of Invalidity Grounds~~	52
A. ~~Ground 1: Claims 1 and 4–12 Are Rendered Obvious By Giordano in view of Tetzlaff~~	52
1. ~~Prior Art Overview~~	52
a) ~~Motivation to Combine Giordano and Tetzlaff~~	55
b) ~~Features of Virtex-UG are Inherent to Giordano, and Giordano Incorporates Virtex-UG by Reference~~	60
2. ~~Ground 1—Analysis~~	62
a) ~~Independent Claim 1~~	62
[1.pre]: ~~A method for processing a first serial data stream, using a field programmable gate array system, to generate a second serial data stream, wherein the method comprises the steps of:~~	62
[1.a]: ~~(a) receiving, by a deserializer in a field programmable array, a clock signal;~~	62
[1.b]: ~~(b) receiving, by the deserializer, the first serial data stream;~~	63
[1.c]: ~~(c) generating, by the deserializer, a receiver side clock signal;~~	64
[1.d]: ~~(d) converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams;~~	64
[1.e]: ~~(e) transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first plurality of parallel data streams;~~	65
[1.f]: ~~(f) transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;~~	66
[1.g]: ~~(g) generating, using the phase lock loop, a transmitter side clock signal;~~	68

[1.h]: (h) performing, by the computational circuitry, a set of operations on at least a portion of the first plurality of parallel data streams to generate a second plurality of parallel data streams;.....	68
[1.i]: (i) converting the second plurality of parallel data streams, by a serializer, into a second serial data stream; and	70
[1.j]: (j) transmitting, from the field programmable gate array system, the second serial data stream,.....	70
[1.k]: wherein said method does not use clock domain crossing operations that delay processing of the first set of parallel data streams.....	71
b) — Claim 4.....	73
[4]: The method of claim 1, wherein the 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.....	73
e) — Claims 5-7.....	74
[5]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed prior to step (g).....	74
[6]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed after step (g).....	74
[7]: The method of claim 1, wherein all of the set of operations performed in step (h) are performed after step (g).....	74
d) — Claim 8.....	75
[8.pre]: 8. The method of claim 1, wherein step (g) further comprises:	75
[8.a]: (k) generating, by the phase lock loop, a feedback clock signal associated with the transmitter side clock signal by performing the following steps until a first output of a phase detector of the field programmable gate array system is below a first threshold level:.....	75
[8.b]: (1) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal;.....	79
[8.c]: (2) generating, by the phase detector, the first output based on a comparison of the receiver side clock signal and the feedback clock signal obtained from the second clock signal;.....	80
[8.d]: (3) transmitting, from the phase detector to a phase controller of the field programmable gate array system, the first output;	80

[8.e]: (4) determining, by the phase controller, interim adjustment information based on the first output; and	82
[8.f]: (5) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;	83
[8.g]: wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information, ...	83
[8.h]: wherein steps (1) through (5) are repeated until the first output of the phase detector is below the first threshold level, and	84
[8.i]: wherein the transmitter side clock signal is associated with the second clock signal.	85
e) — Claim 9	86
[9]: The method of claim 1, wherein the first plurality of parallel data streams has the same number of data streams as the second plurality of parallel data streams.	86
f) — Claim 10	87
[10]: The method of claim 1, wherein the first plurality of data streams and the second plurality of data streams comprise 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; (viii) eighty (80) data streams; (ix) one hundred twenty-eight (128) data streams; and (x) one hundred sixty (160) data streams.	87
g) — Claim 11	88
[11]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and phase.	88
h) — Claim 12	89
[12]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and different phases.	89
B. — Ground 2: Claims 1–12 Are Rendered Obvious By Raymond in view of Tetzlaff	90
1. — Prior Art Overview	90
a) — Motivation to Combine Raymond and Tetzlaff	92

2. Ground 2—Analysis	96
a) Independent Claim 1	96
[1.pre]: A method for processing a first serial data stream, using a field programmable gate array system, to generate a second serial data stream, wherein the method comprises the steps of:	96
[1.a]: (a) receiving, by a deserializer in a field programmable array, a clock signal;	96
[1.b]: (b) receiving, by the deserializer, the first serial data stream;	96
[1.c]: generating, by the deserializer, a receiver side clock signal;	97
[1.d]: converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams;	97
[1.e]: transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first plurality of parallel data streams;	98
[1.f]: transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;	98
[1.g]: generating, using the phase lock loop, a transmitter side clock signal;	99
[1.h]: performing, by the computational circuitry, a set of operations on at least a portion of the first plurality of parallel data streams to generate a second plurality of parallel data streams;	100
[1.i]: converting the second plurality of parallel data streams, by a serializer, into a second serial data stream; and	100
[1.j]: transmitting, from the field programmable gate array system, the second serial data stream;	101
[1.k]: wherein said method does not use clock domain crossing operations that delay processing of the first set of parallel data streams.	101
b) Claim 2	103
[2]: The method of claim 1, wherein the first serial data stream comprises market data, the second serial data stream comprises order entry data, and the set of operations are associated with a trading algorithm.	103
c) Claim 3	103

[3]: The method of claim 1, wherein the first serial data stream includes market data and the second serial data stream includes trading data.....	103
d) — Claim 4.....	104
[4]: The method of claim 1, wherein the 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.....	104
e) — Claims 5–7.....	104
[5]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed prior to step (g).....	104
[6]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed after step (g).	104
[7]: The method of claim 1, wherein all of the set of operations performed in step (h) are performed after step (g).....	104
f) — Claim 8.....	105
g) — Claim 9.....	106
[9]: The method of claim 1, wherein the first plurality of parallel data streams has the same number of data streams as the second plurality of parallel data streams.	106
h) — Claim 10.....	107
[10]: The method of claim 1, wherein the first plurality of data streams and the second plurality of data streams comprise 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; (viii) eighty (80) data streams; (ix) one hundred twenty-eight (128) data streams; and (x) one hundred sixty (160) data streams.	107
i) — Claim 11.....	107
[11]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and phase.....	107
j) — Claim 12.....	108
[12]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and different phases.	108

C. ~~Ground 3: Claims 1-12 Are Rendered Obvious by Giordano in View of Tetzlaff further in View of Raymond~~	109
1. ~~Prior Art Overview~~	110
a) ~~Motivation to Combine Giordano, Tetzlaff, and Raymond~~	110
2. ~~Ground 3—Analysis~~	111
a) ~~Claims 1, 4-12~~	111
b) ~~Claims 2-3~~	111
D. ~~Ground 4: Claims 1 and 4-12 Are Rendered Obvious By Giordano in view of Lee~~	111
1. ~~Prior Art Overview~~	112
a) ~~Motivation to Combine Giordano and Lee~~	112
2. ~~Ground 4—Analysis~~	114
a) ~~Independent Claim 1~~	114
~~[1.pre] through [1.e]:~~	114
~~[1.f]: transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;~~	115
~~[1.g]: generating, using the phase lock loop, a transmitter side clock signal;~~	116
~~[1.h] through [1.k]:~~	117
b) ~~Claims 4-7~~	117
c) ~~Claim 8~~	117
~~[8.pre]: 8. The method of claim 1, wherein step (g) further comprises: ...~~	117
~~[8.a]: (k) generating, by the phase lock loop, a feedback clock signal associated with the transmitter side clock signal by performing the following steps until a first output of a phase detector of the field programmable gate array system is below a first threshold level:~~	117
~~[8.b]: (1) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal;~~	118
~~[8.c]: (2) generating, by the phase detector, the first output based on a comparison of the receiver side clock signal and the feedback clock signal obtained from the second clock signal;~~	119

[8.d]: (3) transmitting, from the phase detector to a phase controller of the field programmable gate array system, the first output;	119
[8.e]: (4) determining, by the phase controller, interim adjustment information based on the first output; and	120
[8.f]: (5) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;	120
[8.g]: wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information;	121
[8.h]: wherein steps (1) through (5) are repeated until the first output of the phase detector is below the first threshold level, and	121
[8.i]: wherein the transmitter side clock signal is associated with the second clock signal.	122
d) — Claims 9–12	122
E. — Ground 5: Claims 1–12 Are Rendered Obvious by Giordano in View of Lee further in View of Raymond	123
1. — Prior Art Overview	123
a) — Motivation to Combine Giordano, Lee, and Raymond	123
2. — Ground 5 — Analysis	123
a) — Claim 1, 4–12	123
b) — Claims 2–3	124
F. — Ground 6: Claims 1, 4–12 Are Rendered Obvious by Giordano in View of Tetzlaff further in View of Virtex-UG	124
1. — Prior Art Overview	124
a) — Motivation to Combine Giordano, Tetzlaff, and Virtex-UG	124
2. — Ground 6 — Analysis	125
a) — Claims 1, 4–12	125
G. — Ground 7: Claims 1, 4–12 Are Rendered Obvious by Giordano in View of Lee further in View of Virtex-UG	126
1. — Prior Art Overview	126
a) — Motivation to Combine Giordano, Lee, and Virtex-UG	126
2. — Ground 7 — Analysis	126

a) — Claims 1, 4–12	126
H. — Ground 8: Claims 1–12 Are Rendered Obvious by Giordano in View of Tetzlaff, Virtex-UG, and Raymond.....	127
1. — Prior Art Overview	127
a) — Motivation to Combine Giordano, Tetzlaff, Virtex-UG, and Raymond —127	
2. — Ground 8 — Analysis	127
a) — Claims 1, 4–12	127
b) — Claims 2–3	128
I. — Ground 9: Claims 1–12 Are Rendered Obvious by Giordano in View of Lee, Virtex-UG, and Raymond.....	128
1. — Prior Art Overview	128
a) — Motivation to Combine Giordano, Lee, Virtex-UG, and Raymond	128
2. — Ground 9 — Analysis	129
b) — Claims 1, 4–12	129
c) — Claims 2–3	129
X. — Secondary Considerations of Non-Obviousness	129
XI. — Conclusion	130
TABLE OF EXHIBITS	20
TABLE OF ABBREVIATIONS	21
I. — BACKGROUND AND QUALIFICATIONS.....	23
II. — SCOPE OF OPINIONS	30
III. — MATERIALS REVIEWED	31
IV. — Person of Ordinary Skill in the Art.....	31
V. — Statement of Legal Principles	31
A. — Legal Standard for Prior Art	32
B. — Anticipation and Prior Art	32
C. — Legal Standard for Obviousness	34
D. — Legal Standard for Claim Construction	39

VI. Summary of the '587 Patent	43
A. Challenged Claims.....	43
B. File History of the '587 Patent	43
VII. Application of the Prior Art to the CHALLENGED Claims	45
A. Priority Date	45
B. Claim Construction.....	45
C. Overview of Prior Art.....	46
1. R. Giordano and A. Aloisio, <i>Fixed-Latency, Multi-Gigabit Serial Links with Xilinx FPGAs</i> , 58-1 IEEE Transactions on Nuclear Science 194 (2011) (“Giordano”) (Ex. 1005).....	46
a) Ex. 1009, Virtex-UG (incorporated by Giordano).....	48
2. Virtex-5 FPGA RocketIO GTP Transceiver User Guide, UG196 (v2.1), December 3, 2009 (“Virtex-UG”) (Ex. 1009)	49
3. U.S. Pat. No. 7,499,513 (“Tetzlaff”) (Ex. 1006)	50
4. U.S. Pat. No. 10,447,463 (“Raymond”) (Ex. 1007).....	51
5. U.S. Pat. No. 8,788,553 (“Lee”) (Ex. 1008)	53
VIII. State of the Art.....	54
A. Serialization and Deserialization (SERDES) in FPGAs	55
B. Phase-Locked Loops (PLLs) and Delay-Locked Loops (DLLs) in FPGAs ..	57
C. Clock Synchronization and Phase Alignment Between Receiver and Transmitter	59
D. Application to Financial Trading Systems	62
E. Summary of State of the Art.....	63
IX. Detailed Analysis of Invalidity Grounds	64
A. Ground 1: Claims 1 and 4–12 Are Rendered Obvious By Giordano in view of Tetzlaff	65
1. Prior Art Overview	65
a) Motivation to Combine Giordano and Tetzlaff.....	68
b) Features of Virtex-UG are Inherent to Giordano, and Giordano Incorporates Virtex-UG by Reference.....	72

2. Ground 1 – Analysis	74
a) Independent Claim 1	74
[1.pre]: A method for processing a first serial data stream, using a field programmable gate array system, to generate a second serial data stream, wherein the method comprises the steps of:	74
[1.a]: (a) receiving, by a deserializer in a field programmable array, a clock signal;	74
[1.b]: (b) receiving, by the deserializer, the first serial data stream;.....	76
[1.c]: (c) generating, by the deserializer, a receiver side clock signal;	76
[1.d]: (d) converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams;.....	77
[1.e]: (e) transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first plurality of parallel data streams;.....	77
[1.f]: (f) transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;.....	78
[1.g]: (g) generating, using the phase lock loop, a transmitter side clock signal;	80
[1.h]: (h) performing, by the computational circuitry, a set of operations on at least a portion of the first plurality of parallel data streams to generate a second plurality of parallel data streams;.....	81
[1.i]: (i) converting the second plurality of parallel data streams, by a serializer, into a second serial data stream; and	82
[1.j]: (j) transmitting, from the field programmable gate array system, the second serial data stream,.....	83
[1.k]: wherein said method does not use clock domain crossing operations that delay processing of the first set of parallel data streams.....	83
b) Claim 4.....	85
[4]: The method of claim 1, wherein the 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.....	85
c) Claims 5-7	86

<u>[5]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed prior to step (g).....</u>	<u>86</u>
<u>[6]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed after step (g).</u>	<u>86</u>
<u>[7]: The method of claim 1, wherein all of the set of operations performed in step (h) are performed after step (g).....</u>	<u>86</u>
<u>d) Claim 8.....</u>	<u>88</u>
<u>[8.pre]: 8. The method of claim 1, wherein step (g) further comprises:</u>	<u>88</u>
<u>[8.a]: (k) generating, by the phase lock loop, a feedback clock signal associated with the transmitter side clock signal by performing the following steps until a first output of a phase detector of the field programmable gate array system is below a first threshold level:.....</u>	<u>88</u>
<u>[8.b]: (1) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal;.....</u>	<u>91</u>
<u>[8.c]: (2) generating, by the phase detector, the first output based on a comparison of the receiver side clock signal and the feedback clock signal obtained from the second clock signal;</u>	<u>92</u>
<u>[8.d]: (3) transmitting, from the phase detector to a phase controller of the field programmable gate array system, the first output;</u>	<u>93</u>
<u>[8.e]: (4) determining, by the phase controller, interim adjustment information based on the first output; and</u>	<u>94</u>
<u>[8.f]: (5) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;</u>	<u>95</u>
<u>[8.g]: wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information,.....</u>	<u>96</u>
<u>[8.h]: wherein steps (1) through (5) are repeated until the first output of the phase detector is below the first threshold level, and.....</u>	<u>97</u>
<u>[8.i]: wherein the transmitter side clock signal is associated with the second clock signal.....</u>	<u>97</u>
<u>e) Claim 9.....</u>	<u>99</u>
<u>[9]: The method of claim 1, wherein the first plurality of parallel data streams has the same number of data streams as the second plurality of parallel data streams.</u>	<u>99</u>

f) Claim 10.....	100
<u>[10]: The method of claim 1, wherein the first plurality of data streams and the second plurality of data streams comprise 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; (viii) eighty (80) data streams; (ix) one hundred twenty-eight (128) data streams; and (x) one hundred sixty (160) data streams.</u>	
g) Claim 11	100
<u>[11]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and phase.....</u>	
h) Claim 12.....	102
<u>[12]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and different phases.</u>	
B. Ground 2: Claims 1–12 Are Rendered Obvious By Raymond in view of Tetzlaff	103
1. Prior Art Overview	103
a) Motivation to Combine Raymond and Tetzlaff	105
2. Ground 2 – Analysis	109
a) Independent Claim 1	109
<u>[1.pre]: A method for processing a first serial data stream, using a field programmable gate array system, to generate a second serial data stream, wherein the method comprises the steps of:</u>	
<u>[1.a]: (a) receiving, by a deserializer in a field programmable array, a clock signal;</u>	
<u>[1.b]: (b) receiving, by the deserializer, the first serial data stream;.....</u>	
<u>[1.c]: generating, by the deserializer, a receiver side clock signal;.....</u>	
<u>[1.d]: converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams;</u>	

<u>[1.e]: transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first plurality of parallel data streams;</u>	<u>111</u>
<u>[1.f]: transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;.....</u>	<u>111</u>
<u>[1.g]: generating, using the phase lock loop, a transmitter side clock signal;</u>	<u>112</u>
<u>[1.h]: performing, by the computational circuitry, a set of operations on at least a portion of the first plurality of parallel data streams to generate a second plurality of parallel data streams;.....</u>	<u>113</u>
<u>[1.i]: converting the second plurality of parallel data streams, by a serializer, into a second serial data stream; and</u>	<u>113</u>
<u>[1.j]: transmitting, from the field programmable gate array system, the second serial data stream,.....</u>	<u>114</u>
<u>[1.k]: wherein said method does not use clock domain crossing operations that delay processing of the first set of parallel data streams.....</u>	<u>114</u>
<u>b) Claim 2.....</u>	<u>116</u>
<u>[2]: The method of claim 1, wherein the first serial data stream comprises market data, the second serial data stream comprises order entry data, and the set of operations are associated with a trading algorithm.</u>	<u>116</u>
<u>c) Claim 3.....</u>	<u>116</u>
<u>[3]: The method of claim 1, wherein the first serial data stream includes market data and the second serial data stream includes trading data.....</u>	<u>116</u>
<u>d) Claim 4.....</u>	<u>117</u>
<u>[4]: The method of claim 1, wherein the 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.....</u>	<u>117</u>
<u>e) Claims 5-7.....</u>	<u>117</u>
<u>[5]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed prior to step (g).....</u>	<u>117</u>
<u>[6]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed after step (g).</u>	<u>117</u>

[7]: The method of claim 1, wherein all of the set of operations performed in step (h) are performed after step (g).....	117
f) Claim 8.....	118
g) Claim 9.....	119
[9]: The method of claim 1, wherein the first plurality of parallel data streams has the same number of data streams as the second plurality of parallel data streams.	119
h) Claim 10.....	120
[10]: The method of claim 1, wherein the first plurality of data streams and the second plurality of data streams comprise 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; (viii) eighty (80) data streams; (ix) one hundred twenty-eight (128) data streams; and (x) one hundred sixty (160) data streams.	120
i) Claim 11	120
[11]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and phase.....	120
j) Claim 12.....	121
[12]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and different phases.	121
C. Ground 3: Claims 1–12 Are Rendered Obvious by Giordano in View of Tetzlaff further in View of Raymond.....	122
1. Prior Art Overview	123
a) Motivation to Combine Giordano, Tetzlaff, and Raymond	123
2. Ground 3 – Analysis	124
a) Claims 1, 4–12	124
b) Claims 2–3	124
D. Ground 4: Claims 1 and 4-12 Are Rendered Obvious By Giordano in view of Lee	124
1. Prior Art Overview	125

a) Motivation to Combine Giordano and Lee.....	125
2. Ground 4 – Analysis	127
a) Independent Claim 1	127
[1.pre] through [1.e]:	127
[1.f]: transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;.....	128
[1.g]: generating, using the phase lock loop, a transmitter side clock signal;	129
[1.h] through [1.k]:	130
b) Claims 4-7.....	130
c) Claim 8.....	130
[8.pre]: 8. The method of claim 1, wherein step (g) further comprises: ...	130
[8.a]: (k) generating, by the phase lock loop, a feedback clock signal associated with the transmitter side clock signal by performing the following steps until a first output of a phase detector of the field programmable gate array system is below a first threshold level:	130
[8.b]: (1) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal;.....	132
[8.c]: (2) generating, by the phase detector, the first output based on a comparison of the receiver side clock signal and the feedback clock signal obtained from the second clock signal;	132
[8.d]: (3) transmitting, from the phase detector to a phase controller of the field programmable gate array system, the first output;	133
[8.e]: (4) determining, by the phase controller, interim adjustment information based on the first output; and	133
[8.f]: (5) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;	134
[8.g]: wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information,.....	134
[8.h]: wherein steps (1) through (5) are repeated until the first output of the phase detector is below the first threshold level, and.....	134

<u>[8.i]: wherein the transmitter side clock signal is associated with the second clock signal.</u>	135
d) Claims 9–12	136
E. <u>Ground 5: Claims 1–12 Are Rendered Obvious by Giordano in View of Lee further in View of Raymond</u>	136
1. Prior Art Overview	136
a) Motivation to Combine Giordano, Lee, and Raymond	136
2. Ground 5 – Analysis	137
a) Claim 1, 4–12	137
b) Claims 2–3	137
F. <u>Ground 6: Claims 1, 4–12 Are Rendered Obvious by Giordano in View of Tetzlaff further in View of Virtex-UG</u>	137
1. Prior Art Overview	137
a) Motivation to Combine Giordano, Tetzlaff, and Virtex-UG	137
2. Ground 6 – Analysis	138
a) Claims 1, 4–12	138
G. <u>Ground 7: Claims 1, 4–12 Are Rendered Obvious by Giordano in View of Lee further in View of Virtex-UG</u>	139
1. Prior Art Overview	139
a) Motivation to Combine Giordano, Lee, and Virtex-UG	139
2. Ground 7 – Analysis	139
a) Claims 1, 4–12	139
H. <u>Ground 8: Claims 1–12 Are Rendered Obvious by Giordano in View of Tetzlaff, Virtex-UG, and Raymond</u>	140
1. Prior Art Overview	140
a) Motivation to Combine Giordano, Tetzlaff, Virtex-UG, and Raymond	140
2. Ground 8 – Analysis	140
a) Claims 1, 4–12	140
b) Claims 2–3	141

<u>I. Ground 9: Claims 1–12 Are Rendered Obvious by Giordano in View of Lee, Virtex-UG, and Raymond</u>	<u>141</u>
<u>1. Prior Art Overview</u>	<u>141</u>
<u>a) Motivation to Combine Giordano, Lee, Virtex-UG, and Raymond</u>	<u>141</u>
<u>2. Ground 9 – Analysis</u>	<u>142</u>
<u>b) Claims 1, 4–12</u>	<u>142</u>
<u>c) Claims 2–3</u>	<u>142</u>
<u>X. Secondary Considerations of Non-Obviousness</u>	<u>142</u>
<u>XI. Conclusion</u>	<u>143</u>

TABLE OF EXHIBITS

Exhibit No.	Document
1001	U.S. Patent No. 12,107,587 (“the ’587 patent”)
1002	File History for U.S. Patent No. 12,107,587
1003	Declaration of Benjamin Carrion Schaefer, Ph.D.
1004	Declaration of Carrie Gardner
1005	R. Giordano and A. Aloisio, <i>Fixed-Latency, Multi-Gigabit Serial Links with Xilinx FPGAs</i> , 58-1 IEEE Transactions on Nuclear Science 194 (2011) [Giordano]
1006	U.S. Pat. No. 7,499,513 [Tetzlaff]
1007	U.S. Pat. No. 10,447,463 [Raymond]
1008	U.S. Pat. No. 8,788,553 [Lee]
1009	Virtex-5 FPGA RocketIO GTP Transceiver User Guide, UG196 (v2.1), December 3, 2009 [Virtex-UG]
1010	<i>(intentionally omitted)</i>
1011	Declaration of Gordon MacPherson
<u>1012</u>	<u>Curriculum Vitae of Benjamin Carrion Schaefer, Ph.D.</u>
<u>1013</u>	<u>Redline of Corrected Petition for Inter Partes Review of U.S. Patent No. 12, 107,587</u>
<u>1014</u>	<u>Redline of Corrected Declaration of Benjamin Carrion Schaefer, Ph.D.</u>

TABLE OF ABBREVIATIONS

Abbreviation	Term
'587 patent	U.S. Patent No. 12,107,587
AIA	America Invents Act
CDC	Clock Domain Crossing
CDR	Clock and Data Recovery
Challenged Claims	Claims 1–12 of U.S. Patent No. 12,107,587
DLL	Delay Locked Loop
Ex.	Exhibit
Fig.	Figure
FPGA	Field Programmable Gate Array
Giordano	R. Giordano and A. Aloisio, <i>Fixed-Latency, Multi-Gigabit Serial Links with Xilinx FPGAs</i> , 58-1 IEEE Transactions on Nuclear Science 194 (2011)
IPR	<i>inter partes</i> review
Lee	U.S. Pat. No. 8,788,553
MAC	Media Access Controller
Patent Owner	HFT Solutions, LLC
PCS	Physical Coding Sublayer
Petitioners	Optiver US LLC and Optiver Trading US LLC
PLL	Phase Locked Loop
PMA	Physical Medium Attachment
Raymond	U.S. Pat. No. 10,447,463
Tetzlaff	U.S. Pat. No. 7,499,513
VCO	Voltage Controlled Oscillator
Virtex-UG	Virtex-5 FPGA RocketIO GTP Transceiver User Guide, UG196 (v2.1), December 3, 2009
VLSI	Very Large Scale Integration

1. I, Dr. Benjamin Carrion Schaefer, declare as follows:
2. I have been retained by counsel for Petitioners Optiver US LLC and Optiver Trading US LLC to submit a Declaration in support of the Petition for *Inter Partes* Review of U.S. Patent No. 12,107,587 (“the Petition”).
3. I have been asked to provide my independent analysis and opinion on whether the ’587 patent is invalid over the prior art. My opinions are based on my experience, knowledge, and skill in computer architecture and VLSI design, as well as my experience with field programmable gate arrays and clocking architectures. In preparation for this declaration, I have reviewed the ’587 patent, its file history, and all exhibits cited in this Declaration and in the Petition.
4. I may rely upon these materials, my experience and knowledge, and/or additional materials to rebut arguments raised by the Patent Owner. Further, I may also consider additional documents and information in forming any necessary opinions, including documents that may not yet have been provided to me.
5. My analysis of the materials produced in this investigation is ongoing and I will continue to review any new material as it is provided. This Declaration represents only those opinions I have formed to date. I reserve the right to revise, supplement, and/or amend my opinions stated herein based on new information and on my continuing analysis of the materials already provided.

6. I am being paid at my normal hourly rate for my time spent working on this matter. I have no personal or financial stake or interest in the outcome of the Petition.

I. BACKGROUND AND QUALIFICATIONS

7. My qualifications are stated more fully in my curriculum vitae, which is attached as **Appendix A**. Below is a summary of my education, work experience, and other qualifications.

8. I am an Associate Professor in the Department of Electrical and Computer Engineering at The University of Texas at Dallas (“UTD”), where I also serve as Assistant Dean for Graduate Student Success in the School of Engineering.

9. I hold a Ph.D. in Reconfigurable Computing from The University of Birmingham, United Kingdom (2003); a Master of Science degree in Microelectronics from Birmingham City University, United Kingdom (1999), awarded with distinction; a Bachelor of Engineering degree in Electronic Engineering from Universidad Politecnica, Madrid, Spain (1997); and a Master of Business Administration from McGill University, Canada (2010). My doctoral research focused specifically on reconfigurable computing architectures, which include Field-Programmable Gate Arrays (“FPGAs”).

10. I have over twenty years of professional experience in the fields of reconfigurable computing, FPGAs, Electronic Design Automation, High-Level

Synthesis, hardware security, low-power design, and approximate computing. Prior to my current position at UTD, I served as an Assistant Professor in the Department of Electronic and Information Engineering at The Hong Kong Polytechnic University (2012–2016).

11. Before entering academia, I was a Researcher at NEC Corporation's Central Research and Development Laboratories in Japan (2007–2012), where I conducted research on high-level synthesis and FPGA-related design automation. I also held postdoctoral research positions at Seoul National University, Korea (2005–2007) and the University of California, Los Angeles (2003–2004).

12. At UTD, I have held the position of Assistant Professor (2016–2022) and Associate Professor (2022–present). My primary research interests include Reconfigurable Computing, FPGAs, Electronic Design Automation, High-Level Synthesis, Hardware Security, Low-power design, and Approximate Computing. I regularly teach graduate and undergraduate courses directly relevant to the subject matter of the '587 patent, including “Reconfigurable Systems” (a graduate-level course focused on FPGA architectures, design, and implementation), “Computer Architecture,” and “Embedded Systems.” I also teach an FPGA design course, and one of the projects for that course involves implementing a financial trading algorithm on an FPGA.

13. I have authored or co-authored over 154 publications, including three books, three book chapters, 41 journal articles, and over 106 conference papers. My publications have been cited over 2,400 times, and my h-index is 26 (as of January 2025). My three books focus on High-Level Synthesis, including “High-Level Synthesis Made Easy” (2023, with a Second Edition in 2026) and “High-Level Synthesis: Applications” (2025). My work has been published in leading journals including IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, ACM Transactions on Design Automation of Electronic Systems, and ACM Computing Surveys.

14. A substantial portion of my research career has been devoted specifically to FPGA design, implementation, and optimization. My Ph.D. dissertation at the University of Birmingham focused on reconfigurable computing using FPGAs. Beginning with my earliest publications in 2001 and 2002, I have published extensively on FPGA implementations, including “Scalable Implementation of the Discrete Element Method on a Reconfigurable Computing Platform” (Field-Programmable Logic and Applications, Springer, 2002), “Analysis and implementation of the discrete element method using a dedicated highly parallel architecture in reconfigurable computing” (IEEE Symposium on Field-Programmable Custom Computing Machines, 2002), “Description of a Dedicated

Hardware Architecture for the Discrete Element Method (DEM) implemented on a Field Programmable Gate Array (FPGA)” (ACME, 2002), and “Acceleration of the Discrete Element Method (DEM) on a reconfigurable co-processor” (Computers and Structures, 2004). These early works demonstrate my deep familiarity with FPGA architectures, hardware description, and implementation.

15. Throughout my career, I have continued to publish prolifically on FPGA-related topics. Representative publications include “Enabling High-Level Synthesis Resource Sharing Design Space Exploration in FPGAs Through Automatic Internal Bitwidth Adjustments” (IEEE Transactions on Computer-Aided Design, 2017); “Learning from the Past: Efficient High-Level Synthesis Design Space Exploration for FPGAs” (ACM TODAES, 2022); “Hardware-assisted Simulation of Voltage-behind-reactance Models of Electric Machines on FPGA” (IEEE Transactions on Energy Conversion, 2020); “Accelerating FPGA Prototyping through Predictive Model-Based HLS Design Space Exploration” (DAC, 2019); “Efficient and reliable High-Level Synthesis Design Space Explorer for FPGAs” (FPL, 2016); “Low Power Design of Runtime Reconfigurable FPGAs through Contexts Approximations” (ICCD, 2019); and “FREEDOM: FPGA-based Hardware Redaction Emulator” (FCCM, 2025). I have also served on the Technical Program Committees of the premier FPGA conferences, including the IEEE Conference on Field Programmable Gate Arrays (FPL), IEEE International Symposium on Field-

Programmable Custom Computing Machines (FCCM), and the IEEE Conference on Field Programmable Technology (FPT).

16. My research also encompasses clocking architectures and frequency control mechanisms used in connection with FPGAs, which are directly relevant to the subject matter of the '587 patent. I have published research addressing clock domain management and dynamic frequency scaling in reconfigurable architectures, including: “Tunable Multiprocess Mapping on Coarse-Grain Reconfigurable Architectures With Dynamic Frequency Control” (IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016), which addresses dynamic frequency control techniques in reconfigurable FPGA-based architectures; “Micro-architecture Tuning for Dynamic Frequency Scaling in Coarse-Grain Runtime Reconfigurable Arrays with Adaptive Clock Domain Support” (IEEE ISVLSI, 2021), which specifically addresses adaptive clock domain support in runtime reconfigurable arrays; and “Low Power Design through Frequency-Optimized Runtime Micro-architectural Adaptation” (ICCD, 2019), which addresses frequency optimization in reconfigurable computing systems. These publications demonstrate my expertise in the design and implementation of clocking mechanisms, clock domain crossings, and frequency management in FPGA and reconfigurable computing systems.

17. My research further encompasses high-speed data communication and interconnect technologies relevant to serialization and deserialization (“SerDes”) in

FPGA systems. I co-authored “Recent trends and considerations for high speed data in chips and system interconnects” (SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), 2015), which addresses high-speed data communication considerations in chip and system interconnects — a topic encompassing serialization and deserialization techniques used to transfer data at high speeds between components. Additionally, my extensive work on High-Level Synthesis for FPGAs inherently involves the design of data communication interfaces, including serialized data paths, between hardware modules and between on-chip and off-chip components. My work on pin multiplexing — “Investigation and Optimization of Pin Multiplexing in High-Level Synthesis” (GLSVLSI, 2018) — directly relates to serialization techniques where multiple data signals are multiplexed over shared physical connections, a fundamental concept underlying serializer/deserializer architectures.

18. I am a Senior Member of IEEE (since 2010). I currently serve as an Associate Editor for IEEE Transactions on Computer-Aided Design (TCAD) and IEEE Transactions on Sustainable Computing. I have served as General Chair of the IEEE International Conference on Computer Design (ICCD) in 2025, TPC Chair of ICCD in 2024, and Track Chair for the ACM/IEEE Design Automation Conference (DAC), IEEE Design Automation and Test Europe (DATE), and ACM/IEEE Asia

and South Pacific Design Automation Conference (ASP-DAC). I have also served as an NSF reviewer on multiple occasions.

19. I hold two patents: U.S. Patent 8,977,996 B2, “Method and Apparatus for Incremental Design Space Exploration,” and a corresponding Japanese Patent (JP 2012-522281).

20. My research has been supported by over \$4 million in external funding from sources including the National Science Foundation (NSF), the Office of Naval Research (ONR), the Air Force Research Laboratory (AFRL/AFWERX), NATO, Boeing, Huawei, and Texas Instruments.

21. I have supervised 15 Ph.D. students and 20 Master’s students to completion, many of whom have gone on to positions at leading semiconductor and technology companies including NVIDIA, Qualcomm, Apple, Cadence, Intel, AMD, and MathWorks.

22. Based on my education, research, publications, teaching, and professional experience spanning over two decades in the fields of reconfigurable computing, FPGA design and implementation, electronic design automation, clock architectures, frequency control, and high-speed data interconnects, I am qualified to offer expert opinions regarding the subject matter of the ’587 patent, including its claims relating to serialization and deserialization, Field-Programmable Gate Arrays,

and clocks used in connection with FPGAs. My opinions in this declaration are based on my knowledge, training, education, and experience in these fields.

II. SCOPE OF OPINIONS

23. I was asked to provide my opinions regarding whether the Challenged Claims of the '587 patent would have been obvious to one of ordinary skill in the art at the time of the claimed invention for claims 1 and 4–12 by Giordano in view of Tetzlaff (Ground 1), claims 1–12 by Raymond in view of Tetzlaff (Ground 2), claims ~~2–3~~1–12 by Giordano in view of Tetzlaff further in view of Raymond (Ground 3), claims 1 and 4–12 by Giordano in view of Lee (Ground 4), ~~and~~ claims ~~2–3~~1–12 by Giordano in view of Lee further in view of Raymond (Ground 5), claims 1 and 4–12 by Giordano in view of Tetzlaff further in view of Virtex-UG (Ground 6), claims 1 and 4–12 by Giordano in view of Lee further in view of Virtex-UG (Ground 7), claims 1–12 by Giordano in view of Tetzlaff, Virtex-UG, and Raymond (Ground 8), and claims 1–12 by Giordano in view of Lee, Virtex-UG, and Raymond (Ground 9).

24. This Declaration, including the exhibits cited herein, sets forth my opinions on these topics. It is my opinion that the Challenged Claims of the '587 patent would have been obvious based on the Grounds listed above. I do not take a position on whether any of the claims are invalid under other aspects of patent law, such as indefiniteness or written description, and my analysis of the claims and

opinions on validity are not intended to convey an opinion that the claims are not otherwise invalid.

III. MATERIALS REVIEWED

25. The materials I considered and relied upon in preparing my Declaration and forming my opinions include all exhibits cited in this Declaration and the Petition, including the '587 patent, the file history of the '587 patent, and the cited prior art.

26. I also relied on my professional experience, knowledge, and skill in computer architecture and VLSI design, as well as my experience with FPGAs and clocking techniques in reaching the opinions expressed in this Declaration.

IV. PERSON OF ORDINARY SKILL IN THE ART

27. A person of ordinary skill in the art ("POSITA") on February 21, 2019, would have a Bachelor's degree in Computer Engineering, Electrical Engineering, or an equivalent discipline, and at least three years of academic or industry experience relating to clocking techniques for FPGAs with high-speed transceivers. Educational requirements may be substituted by greater relevant work experience, and vice versa.

V. STATEMENT OF LEGAL PRINCIPLES

28. I am not an attorney. I am informed by attorneys of the relevant legal principles and have applied them to arrive at the opinions set forth in this declaration.

A. Legal Standard for Prior Art

29. I understand that the petitioner for *inter partes* review may request the cancellation of one or more claims of a patent based on grounds available under AIA 35 U.S.C. § 102 and 35 U.S.C. § 103 using prior art that consists of patents and printed publications.

B. Anticipation and Prior Art

30. I understand that § 102 specifies when a challenged claim is invalid for lacking novelty over the prior art, and that this concept is also known as “anticipation.” I understand that a prior art reference anticipates a challenged claim, and thus renders it invalid by anticipation, if all elements of the challenged claim are disclosed in the prior art reference. I understand the disclosure in the prior art reference can be either explicit or inherent, meaning it is necessarily present or implied. I understand that the prior art reference does not have to use the same words as the challenged claim, but all of the requirements of the claim must be disclosed so that a person of ordinary skill in the art could make and use the claimed subject matter.

31. I also understand that § 102 defines what is available for use as a prior art reference to a challenged claim.

32. Under § 102(a)(1), a challenged claim is anticipated if it was patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention.

33. Under § 102(a)(2), a challenged claim is anticipated if it was described in an issued patent or published patent application, in which the patent or application names another inventor and was effectively filed before the effective filing date of the claimed invention.

34. I understand that the effective filing date of a patent or patent application is the earlier of the actual filing date of the patent or the application for patent or, if the patent or application for patent is entitled to claim earlier priority or the benefit of an earlier filing, the filing date of the earliest application that describes the claimed subject matter.

35. I understand that when a challenged claim covers several structures, either generically or as alternatives, the claim is deemed anticipated if any of the structures within the scope of the claim is found in the prior art reference.

36. I understand that when a challenged claim requires selection of an element from a list of alternatives, the prior art teaches the element if one of the alternatives is taught by the prior art.

C. Legal Standard for Obviousness

37. I understand that even if a challenged claim is not anticipated, it is still invalid if the differences between the claimed subject matter and the prior art are such that the claimed subject matter would have been obvious to a person of ordinary skill in the pertinent art at the time of the claimed invention.

38. I understand that for a prior art reference to be proper for use in an obviousness analysis, the reference must be analogous art. I understand that a prior art reference is analogous art if: (1) the reference is from the same field of endeavor as the challenged patent (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor of the challenged patent (even if it is not in the same field of endeavor). I understand that the field of endeavor is determined by reference to explanations of the challenged claims' subject matter in the challenged patent, including the embodiments, functions, and structures of the challenged claims. A prior art reference is "reasonably pertinent" to a problem if it would have logically commended itself to an inventor's attention in considering the problem.

39. I understand that obviousness must be determined with respect to the challenged claim as a whole.

40. I understand that one cannot rely on hindsight in deciding whether a claim is obvious.

41. I also understand an obviousness analysis includes the consideration of factors such as (1) the scope and content of the prior art, (2) the differences between the prior art and the challenged claim, (3) the level of ordinary skill in the pertinent art, and (4) “secondary” or “objective” evidence of non-obviousness.¹

42. I understand a challenged claim can be invalid for obviousness over a combination of prior art references if a reason existed (at the time of the claimed invention) that would have prompted a person of ordinary skill in the art to combine elements of the prior art in the manner required by the challenged claim. I understand that this requirement is also referred to as a “motivation to combine,” “suggestion to combine,” or “reason to combine,” and that there are several rationales that meet this requirement.

43. I understand the prior art references themselves may provide a motivation to combine, but other times simple common sense can link two or more prior art references. I further understand that obviousness analysis recognizes that

¹ Secondary or objective evidence of non-obviousness includes evidence of: (1) a long felt but unmet need in the prior art that was satisfied by the claimed invention; (2) commercial success or the lack of commercial success of the claimed invention; (3) unexpected results achieved by the claimed invention; (4) praise of the claimed invention by others skilled in the art; (5) taking of licenses under the patent by others; (6) deliberate copying of the claimed invention; and (7) contemporaneous and independent invention by others. However, I understand that there must be a relationship between any secondary evidence of non-obviousness and the claimed invention.

market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may come from market forces.

44. I understand obviousness to include, for instance, scenarios where known techniques are simply applied to other devices, systems, or processes to improve them in an expected or known way. I also understand that practical and common-sense considerations should be applied in a proper obviousness analysis. For instance, familiar items may have obvious uses beyond their primary purposes.

45. I understand combining familiar elements according to known methods is obvious when it yields predictable results. For instance, obviousness bars patentability of a predictable variation of a technique even if the technique originated in another field of endeavor. This is because design incentives and other market forces can prompt variations of it, and predictable variations are not the product of innovation, but rather ordinary skill and common sense.

46. I understand a particular combination may be obvious if it was obvious to try the combination. For example, when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. This would result in something obvious because the result is the product not of innovation but of ordinary skill and common sense. However, I

understand that it may not be obvious to try a combination when it involves unpredictable technologies.

47. I also understand a proper obviousness analysis focuses on what was known or obvious to a person of ordinary skill in the art, not just the patentee. Accordingly, I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

48. It is my understanding that the Manual of Patent Examining Procedure §2143 sets forth the following as exemplary rationales that support a conclusion of obviousness:

(A) Combining prior art elements according to known methods to yield predictable results;

(B) Simple substitution of one known element for another to obtain predictable results;

(C) Use of known technique to improve similar devices (methods, or products) in the same way;

(D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;

(E) “Obvious to try” – choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;

(F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;

(G) Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

49. A person of ordinary skill in the art looking to overcome a problem will often use the teachings of multiple publications together. I understand that obviousness analysis considers inferences, creative steps, common sense, and practical logic and applications that a person of ordinary skill in the art would employ under the circumstances.

50. I understand a claim can be obvious in light of a single reference, if the elements of the challenged claim that are not explicitly or inherently disclosed in the reference can be supplied by the common sense of one of skill in the art.

51. I understand obviousness also bars the patentability of applying known or obvious design choices to the prior art. One cannot patent merely substituting one prior art element for another if the substitution can be made with predictable results. Likewise, combining prior art techniques that are interoperable with respect to one another is generally obvious and not patentable.

52. For a claim to be found invalid based upon a modification or combination of the prior art, there must be reasonable expectation that a person of ordinary skill would have successfully modified or combined the prior art to arrive at the claimed arrangement. This does not mean that it must be certain that a person of ordinary skill would have been successful—the law only requires that the person of ordinary skill in the art would have perceived a reasonable expectation of success in modifying or combining the prior art to arrive at the claimed invention.

53. In sum, my understanding is obviousness invalidates claims merely reciting combinations or obvious variations of prior art teachings using understanding and knowledge of one of skill in the art at the time and motivated by the general problem facing the inventor at the time. Under this analysis, the prior art references themselves, or any need or problem known in the field of endeavor at the time of the invention, can provide a reason for combining the elements of or attempting obvious variations on prior art references in the claimed manner.

D. Legal Standard for Claim Construction

54. I understand that before any invalidity analysis can be properly performed, the scope and meaning of the challenged claims must be determined by claim construction.

55. I understand that a patent may include two types of claims, independent claims and dependent claims. I understand that an independent claim stands alone

and includes only the limitations it recites. I understand that a dependent claim depends from an independent claim or another dependent claim. I understand that a dependent claim includes all the limitations that it recites in addition to the limitations recited in the claim (or claims) from which it depends.

56. In comparing the challenged claims to the prior art, I carefully considered the patent and its file history in light of the understanding of a person of skill at the time of the earliest filing of the applicant's claimed invention.

57. I understand that to determine how a person of ordinary skill would have understood a claim term, one should look to sources available at the time of the claimed invention that show what a person of skill in the art would have understood disputed claim language to mean. It is my understanding that this may include what is called "intrinsic" evidence as well as "extrinsic" evidence.

58. I understand that, in construing a claim term, one should primarily rely on intrinsic patent evidence, which includes the words of the claims themselves, the remainder of the patent specification, and the file history. I understand that extrinsic evidence, which is evidence external to the patent and the file history, may also be useful in interpreting patent claims when the intrinsic evidence itself is insufficient. I understand that extrinsic evidence may include principles, concepts, terms, and other resources available to those of skill in the art at the time of the invention.

59. I understand words or terms should be given their ordinary and accepted meaning unless it appears that the inventor ~~wase~~was using them to mean something else or something more specific.

60. I understand that to determine whether a term has special meaning, the claims, the patent specification, and the file history are particularly important, and may show that the inventor gave a term a particular definition or intentionally disclaimed, disavowed, or surrendered claim scope.

61. I understand the claims of a patent define the scope of the rights conferred by the patent. I understand that because the claims point out and distinctly claim the subject matter which the inventor regards as the invention, claim construction analysis must begin with and is focused on the claim language itself. I understand that the context of the term within the claim as well as other claims of the patent can inform the meaning of a claim term. For example, because claim terms are normally used consistently throughout the patent, how a term is used in one claim can often inform the meaning of the same term in other claims. Differences among claims or claim terms can also be a useful guide in understanding the meaning of particular claim terms. I understand a claim term should be construed not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the entire specification. I understand

that because the specification is a primary basis for construing the claims, a correct construction should align with the specification.

62. I understand the file history of the patent as well as art incorporated by reference or otherwise cited during the file history is also highly relevant in construing claim terms. For instance, art cited or incorporated by reference may indicate how the inventor and others of skill in the art at the time of the invention understood certain terms and concepts. Additionally, the file history may show that the inventor disclaimed or disavowed claim scope, or further explained the meaning of a claim term.

63. With regard to extrinsic evidence, I understand that all evidence external to the patent and file history, including expert and inventor testimony, dictionaries, and learned treatises, can also be considered. For example, technical dictionaries may indicate how one of skill in the art used or understood the claim terms. However, I understand that extrinsic evidence is considered to be less reliable than intrinsic evidence, and for that reason is generally given less weight than intrinsic evidence.

64. I understand that in general, a term or phrase found in the introductory words or preamble of the claim, should be construed as a limitation if it recites essential structure or steps, or is necessary to give meaning to the claim.

VI. SUMMARY OF THE '587 PATENT

65. The '587 patent purports to address “phase matching receiver side and transmitter side clocks of a FPGA without introducing unnecessary delay in processing.” Ex. 1001, 2:23–26. The patent states that FPGAs were already used in applications requiring fast processing, including high-frequency trading, because FPGAs allow computations “to occur on a single chip that has massive fine-grained parallelism.” *Id.*, 1:38–44. The patent identifies the asserted problem as “a need to synchronize receiving side and transmitting side clock signals within the FPGA,” and acknowledges that “[t]he prior art sought to address this problem by including a clock domain crossing circuit in the FPGA,” which the patent says “inherently add[s] a delay to the processing that takes place in the FPGA.” *Id.*, 1:43–53. To address that asserted latency issue, the patent describes “a phase control circuit, provided at least partially on the field programmable gate array.” *Id.*, 3:27–31.

A. Challenged Claims

66. All of the claims of the '587 patent are Challenged Claims (claims 1–12).

B. File History of the '587 Patent

67. The application of the '587 patent was filed on April 18, 2022. During prosecution, the Examiner rejected original claims 1–12 as obvious over Zhang (U.S. Patent No. 8,873,606) in view of Cho (U.S. Patent Application Publication No.

2018/0027431). Ex. 1002, pdf pp. 154–162 (10/26/2023 Non-Final Office Action, pp. 2–8).

68. In response, Applicant amended claim 1 to include the limitation “converting the second plurality of parallel data streams, by a serializer, into a second serial data stream,” and argued that Zhang did not disclose that limitation as Zhang modified the second parallel data stream into a third parallel data stream before converting to the second serial data stream. Ex. 1002, pdf pp. 142–43 (1/23/2024 Remarks, pp. 5–6). Applicant also argued Zhang’s FIFO block is “a clock-domain-crossing circuit that delayed processing of parallel signals,” and Zhang’s use of that block fails to disclose the limitation in claim [1] that excludes using such circuits. *Id.*

69. After the rejection was made final, Applicant further amended claims 1 and 9 to their current form, submitted an expert declaration from Dr. James M. Knox, filed a terminal disclaimer, and further pursued its argument that Zhang does not disclose the “second plurality of parallel data streams” and “clock domain crossing operations” limitations. *Id.*, pdf pp. 76–84 (5/14/2024 Remarks, pp. 9–17).

70. The Examiner then allowed claims 1–12 as amended. The Notice of Allowance stated that the claimed “second plurality of parallel data streams” should not be altered before conversion into the “second serial data stream,” and that

Zhang's FIFO buffer performed clock-domain-crossing operations excluded by the claims. *Id.*, pdf pp. 14–20 (5/30/2024 Notice of Allowance (Allowability), pp. 1–7).

71. The Examiner did not apply Giordano, Tetzlaff, Raymond, or Lee as the basis for any rejection, nor does the '587 patent cite to these references. Ex. 1001, 1–2.

VII. APPLICATION OF THE PRIOR ART TO THE CHALLENGED CLAIMS

A. Priority Date

72. I have been instructed to assume the '587 patent and the Challenged Claims have a priority date of February 21, 2019, through numerous continuation applications back to U.S. Provisional Application No. 62/808,404, which was filed on February 21, 2019. Ex. 1001, p. 1 at “Related U.S. Application Data.” If the actual priority date should be different than February 21, 2019, I reserve the right to supplement my opinions based on that change in priority date, if necessary.

B. Claim Construction

73. As I explained above, unless I specify otherwise, I applied the ordinary and customary meaning of the terms of the Challenged Claims as understood by a POSITA throughout my analysis in this Declaration.

C. Overview of Prior Art

74. The below challenges are made in view of the following prior art references, none of which were cited or considered during prosecution of the '587 patent.

1. **R. Giordano and A. Aloisio, *Fixed-Latency, Multi-Gigabit Serial Links with Xilinx FPGAs*, 58-1 IEEE Transactions on Nuclear Science 194 (2011) (“Giordano”) (Ex. 1005)**

75. Giordano was published in IEEE Transactions on Nuclear Science on January 28, 2011. I reviewed the declarations of Carrie Gardner and Gordon MacPherson concerning the public availability of Giordano and understand that Giordano was publicly available before February 21, 2019. In my opinion, a POSITA working in FPGA-based high-speed serial communications, clock synchronization, and SerDes systems before February 21, 2019, would have looked to publications, including those of IEEE, as a standard source of technical information. Giordano therefore qualified as prior art under AIA 35 U.S.C. §102(a)(1). Giordano was neither cited nor considered during prosecution of the '587 patent.

76. Giordano is a 2011 IEEE paper that explains timing synchronization and clock-distribution applications “would benefit from high-speed fixed-latency SerDeses.” Ex. 1005, 194. Giordano presents “a link architecture based on high-speed transceivers embedded in latest generation [FPGAs]” based on “the GTP

transceiver” of the “Xilinx Virtex 5 FPGA family.” Ex. 1005, 194. Giordano further explains that GTPs are available inside the FPGA as “configurable hard-macros (or ‘tiles’),” with each tile including a pair of transceivers (i.e., a receiver and a transmitter) that share components including a phase-locked loop. *Id.* Within each transceiver is the Physical Medium Attachment (PMA) sublayer for serializing (in transmitter PMA) and deserializing (in receiver PMA) data, and the Physical Coding Sublayer (PCS) for additional data processing before serialization (in transmitter PCS) and after deserialization (in receiver PCS). *Id.*

77. Giordano also describes an FPGA Fabric for programming circuit blocks as needed. In Giordano, circuit blocks such as “Comma Detect and Align” and DLLs are programmed in the FPGA Fabric. *Id.*, 197–199. A block diagram of the GTP transceiver is shown in Figure 1 of Giordano:

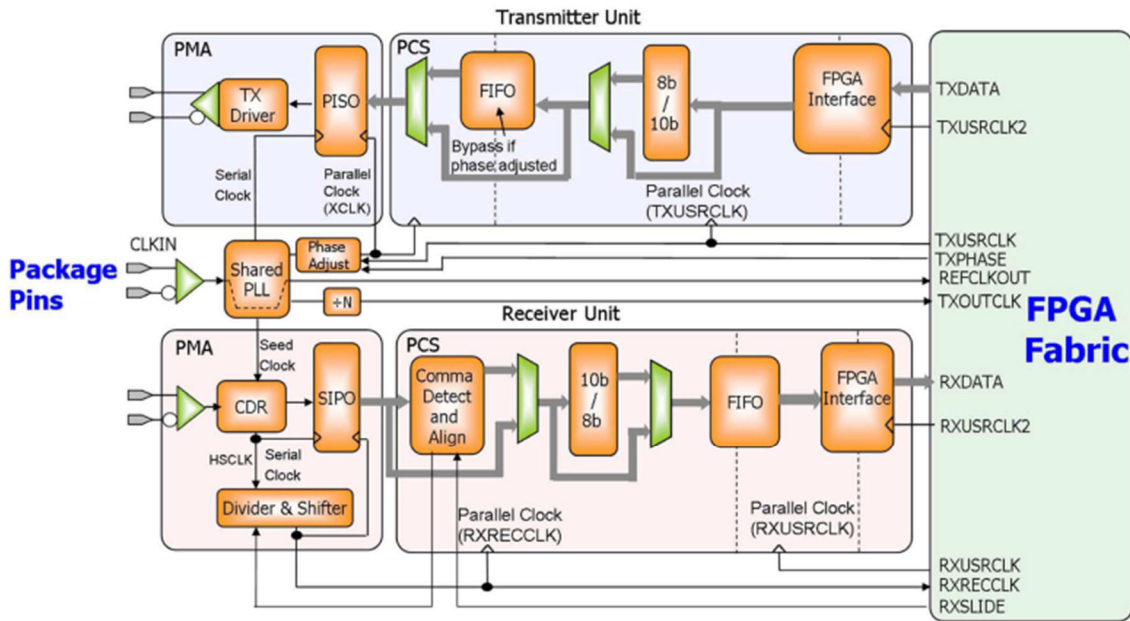


Fig. 1. Simplified block diagram of the GTP transceiver. Half a tile is shown.

a) Ex. 1009, Virtex-UG (incorporated by Giordano)

78. Giordano presents a system with the “RocketIO” GTP transceiver of the Xilinx Virtex 5 FPGA family. Ex. 1005, 194, 199. Giordano repeatedly references the RocketIO GTP Transceiver, for example referencing the overall transceiver architecture (*id.*, 195) and taking estimated latency values from the RocketIO GTP Transceiver User Guide (*id.*, 198). Giordano further states that “the procedure we follow for activating the alignment circuit is the one suggested in the GTP user’s guide.” *Id.*, 197. As such, Giordano inherently discloses the features of Virtex-UG (Ex. 1009), and further incorporates Virtex-UG by reference. Virtex-UG is Xilinx’s Virtex-5 FPGA RocketIO GTP Transceiver User Guide, UG196 (v2.1),

dated December 3, 2009, and hence publicly available before Giordano. Ex. 1009, 1.

2. **Virtex-5 FPGA RocketIO GTP Transceiver User Guide, UG196 (v2.1), December 3, 2009 (“Virtex-UG”) (Ex. 1009)**

79. Virtex-UG was published on December 3, 2009. Virtex-UG therefore qualified as prior art under AIA 35 U.S.C. §102(a)(1). Virtex-UG was neither cited nor considered during prosecution of the ~~’865~~’587 patent. Virtex-UG is Xilinx’s Virtex-5 FPGA RocketIO GTP Transceiver User Guide, UG196 (v2.1), dated December 3, 2009, hence publicly available before Giordano. Ex. 1009, 1; Ex. 1004, ¶¶41–47. I reviewed the declaration of Carrie Gardner concerning the public availability of Virtex-UG and understand that Virtex-UG was publicly available before February 21, 2019. In my opinion, a POSITA working in FPGA-based high-speed serial communications, clock synchronization, and SerDes systems before February 21, 2019, would have looked to publications, including those of IEEE and Virtex-UG, as a standard source of technical information. Virtex-UG states that it “shows how to use the RocketIO™ GTP transceivers in Virtex®-5 FPGAs,” and describes the GTP transceiver as a power-efficient, highly configurable transceiver “tightly integrated with the programmable logic resources of the FPGA.” Ex. 1009, 17, 23. The GTP transceiver supports line rates from 100 Mb/s to 3.75 Gb/s, optional PCS features such as 8B/10B encoding, comma alignment, channel bonding, and clock correction, and “fixed latency modes for minimized, deterministic datapath

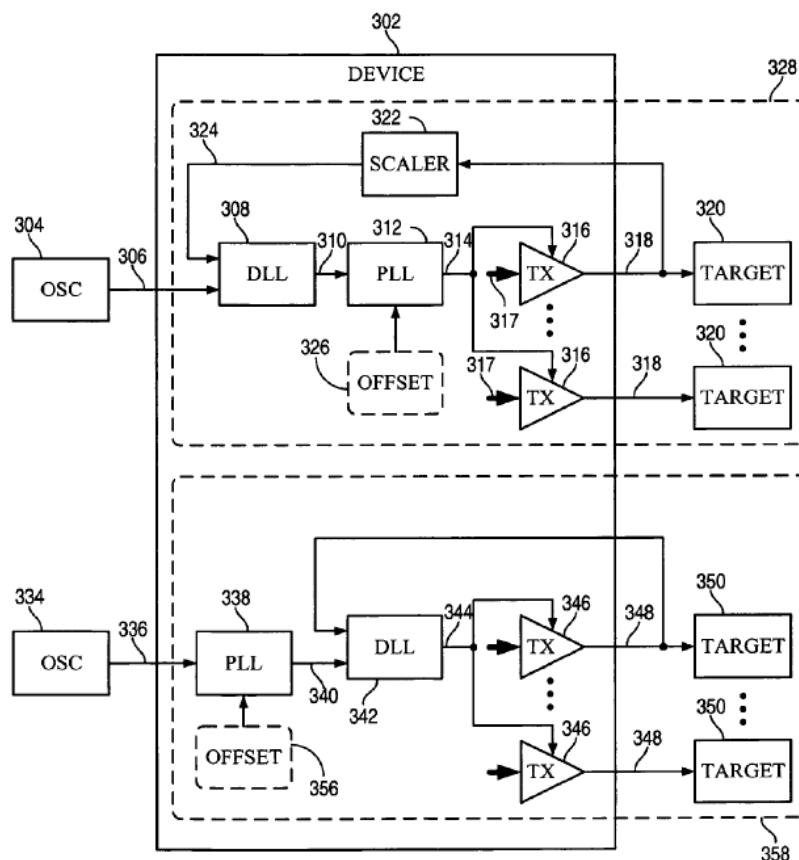
latency.” *Id.*, 23. The guide also explains that GTP transceivers are placed as dual-transceiver GTP_DUAL tiles in Virtex-5 LXT and SXT devices, allowing two transceivers to share a single PLL with the TX and RX functions of both, just how Giordano describes its use of the Virtex-5 FPGA. *Id.*, 24; Ex. 1005, 194.

3. U.S. Pat. No. 7,499,513 (“Tetzlaff”) (Ex. 1006)

80. Tetzlaff is titled “Method and Apparatus for Providing Frequency Synthesis and Phase Alignment in an Integrated Circuit,” was filed on February 2, 2005, and granted on March 3, 2009. Tetzlaff therefore qualifies as prior art under AIA 35 U.S.C. §102(a)(1) and §102(a)(2). Tetzlaff was neither cited nor considered during prosecution of the ’587 patent.

81. Tetzlaff (U.S. Pat. No. 7,499,513) explains that “Programmable logic devices (PLDs) are a well-known type of integrated circuit that can be programmed to perform specified logic functions” and that “[o]ne type of PLD, the field programmable gate array (FPGA), typically includes an array of programmable tiles.” Ex. 1006, 1:22–26. It further states that “[a]s data transfer speeds have increased, high-speed differential serial lines have replaced large parallel buses in many designs.” *Id.*, 1:60–62. Tetzlaff identifies clock skew as a problem in synchronous systems, explaining that clock edges in different parts of a system may not be synchronized due to “clock buffer delays,” “high capacitance of heavily loaded clock lines,” and “propagation delays.” *Id.*, 2:9–16. Tetzlaff addresses those issues

through “methods and apparatuses for clock distribution, phase alignment, and frequency synchronization via high speed serial I/O transceivers,” using PLLs, DLLs, and combinations of such. *Id.*, 4:21–26. For example, Figure 3 shows Tetzlaff’s clock distribution solution using DLLs and PLLs as components:



4. U.S. Pat. No. 10,447,463 (“Raymond”) (Ex. 1007)

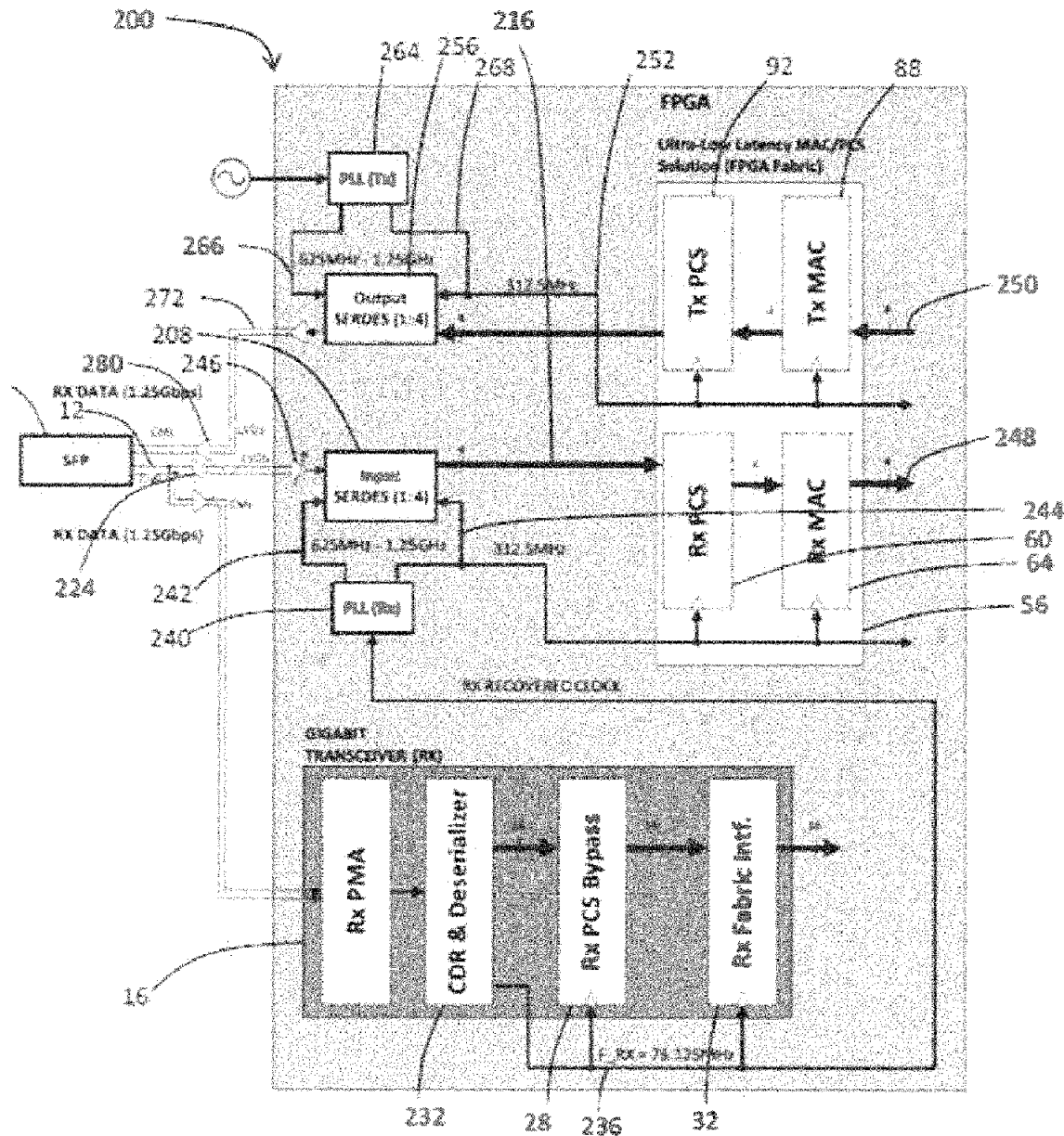
82. Raymond is titled “Device and Method for Ultra-low Latency Communication,” claims priority to provisional application No. 62/531,965 filed on July 13, 2017. Raymond was filed on July 13, 2018, the application published on January 17, 2019, and granted on October 15, 2019. Raymond therefore qualifies as

prior art under AIA 35 U.S.C. §102(a)(2). Raymond was neither cited nor considered during prosecution of the '587 patent.

83. Raymond (U.S. Patent No. 10,447,463) explains that “[u]ltra-low latency communication is critical” for applications depending on round-trip time, including “electronic trading,” and that “some trading firms are now using FPGAs to improve throughput and latency performances of their trading equipment.” Ex. 1007, 1:18–24. Raymond discusses “incoming signals” in “context of financial trading data,” and exchanging such data with “financial exchange.” *Id.*, 4:52–59.

84. Raymond discloses an “ultra-low latency communication device” that includes “a clock recovery module, a de-serializer module, an FPGA fabric and a serializer module.” *Id.*, 1:42–44.

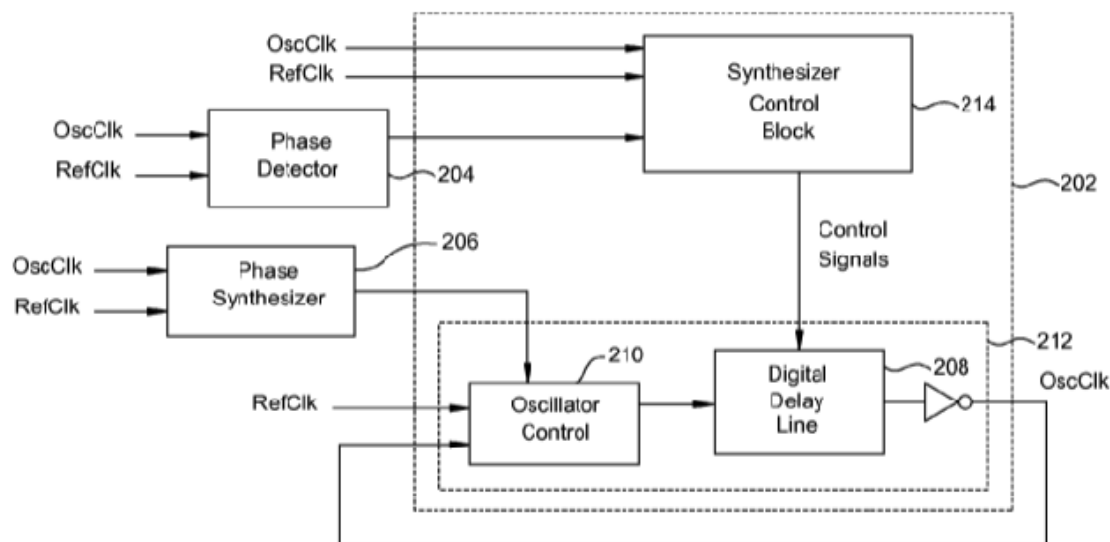
85. Raymond states that “prefabricated transceivers used to process incoming and outgoing physical layer signals” may be “replaced by low-latency serializer/de-serializer modules” that are programmed by the FPGA. *Id.*, 4:43–47; *see also* 9:3–8 (“SerDes module” is “implemented using programmable logic”), 11:65–67. Raymond’s proposed solution is depicted in Figure 2:



5. U.S. Pat. No. 8,788,553 (“Lee”) (Ex. 1008)

86. Lee is titled “Method and Circuit for Providing Digital Frequency Synthesis,” was filed January 28, 2009, and granted July 22, 2014. Lee therefore qualifies as prior art under AIA 35 U.S.C. §102(a)(1) and §102(a)(2). Lee was neither cited nor considered during prosecution of the ’587 patent.

87. Lee (U.S. Pat. No. 8,788,553) discloses adjustable PLLs to accommodate different clocks for receivers and transmitters, with capabilities for frequency synthesis and phase shifting. Ex. 1008, 1:12–23. Lee addresses taking advantage of frequency aligned mode and phase aligned mode, separately and together for finer clock management. *Id.*, 2:35–55. Such solution in Lee is depicted in Figure 2:



VIII. STATE OF THE ART

88. As of the priority date of February 21, 2019, the state of the art in FPGA-based serial communication systems was highly developed. FPGAs had been commercially available for decades and were widely deployed in applications requiring high-speed data processing, including financial trading systems. Ex. 1001, 1:41-43 (FPGAs are used in the financial industry in high frequency trading where the rapid processing of the FPGA is desired). The fundamental building blocks at

issue in this proceeding—serializers/deserializers (SERDES), phase-locked loops (PLLs), delay-locked loops (DLLs), clock and data recovery (CDR) circuits, and phase alignment mechanisms—were all well-understood, standardized components that had been available in commercial FPGA devices for many years prior to the priority date. Commercial user guides, such as Virtex-5 RocketIO GTP Transceiver User Guide (“Virtex-UG”), ~~documents~~document many of these features.

89. I am familiar with the body of technical literature and patent disclosures concerning FPGA clock synchronization, serialization/deserialization, and phase-locked loop architectures as they existed before the priority date. In my opinion, the techniques claimed in the ’587 patent represent straightforward applications of well-known principles to standard FPGA hardware resources. Below, I describe the state of the art in each relevant technical area as of February 21, 2019.

A. Serialization and Deserialization (SERDES) in FPGAs

90. By February 2019, high-speed serial transceivers with integrated SERDES functionality were a standard feature of commercial FPGAs. These transceivers converted serial data streams into parallel data for processing in the FPGA fabric, and parallel data from the FPGA fabric back into serial data streams for transmission. Xilinx’s Virtex-5 FPGA, for example, included RocketIO GTP transceivers that implemented deserialization through a serial-in/parallel-out (SIPO) block, and serialization through a parallel-in/serial-out (PISO) block and supporting

data widths of 8, 10, 16, or 20 bits. Ex. 1009, 104. These transceivers operated at multi-gigabit rates, with Giordano demonstrating operation at 2.5 Gbps. Ex. 1005, 197.

91. The architecture of these transceivers was organized into well-defined sublayers. As Giordano explains, the Physical Medium Attachment (PMA) sublayer contained the Serial In to Parallel Output (SIPO) and Parallel input to Serial Output (PISO) blocks responsible for serialization and deserialization, while the Physical Coding Sublayer (PCS) handled data processing functions such as 8b/10b encoding and decoding. Ex. 1005, 194-95. The transceiver connected to the FPGA fabric through user-accessible clock and data interfaces, allowing the FPGA designer to process deserialized parallel data using custom logic. *Id.* Raymond similarly describes an ultra-low latency communication device architecture comprising “a clock recovery module, a de-serializer module, an FPGA fabric, and a serializer module.” Ex. 1007, 1:42–44.

92. It was further known that SERDES modules could be implemented either as hardened (pre-fabricated) transceiver blocks within the FPGA or as soft logic within the programmable fabric itself. Raymond expressly teaches replacing “prefabricated transceivers with low-latency serializer/de-serializer modules,” implemented “using programmable logic.” Ex. 1007, 4:43–47, 9:3–8. Raymond discloses that Xilinx UltraScale GTY transceivers support “internal data buses of 16,

20, 32, 40 or 64 bits” for interfacing between the serial link and the parallel processing domain. Ex. 1007, 6:5–8. These various port widths were standard options that a person of ordinary skill in the art would have routinely selected based on the desired line rate and processing requirements.

B. Phase-Locked Loops (PLLs) and Delay-Locked Loops (DLLs) in FPGAs

93. Phase-locked loops and delay-locked loops were fundamental clock management resources available in FPGAs well before the priority date. PLLs were used for frequency synthesis—generating output clocks at integer or fractional multiples of an input reference clock—and for phase alignment between clock signals. Lee discloses an all-digital frequency synthesis circuit for integrated circuits, including programmable logic devices (PLDs) such as FPGAs. Ex. 1008, 7:18–35. Lee’s circuit comprises “a phase detector for receiving a reference clock signal and an oscillator clock signal, wherein said phase detector outputs an error signal,” and “a synthesizer control block” that generates “a delay select signal” controlling a digitally controlled oscillator. Ex. 1008, cl. 1, 2:57–3:27. Lee’s synthesizer control block includes both “an integral adjustment (Iadj) filter” “and a proportional adjustment (Padj) filter,” which together provide loop stability such that “a hard phase alignment is no longer needed to maintain the phase lock.” Ex. 1008, 4:27–43. Lee’s implementation achieves “PLL-like operation while using components

that are all digital.” Ex. 1008, 4:42–43. Lee’s circuit was specifically designed for implementation in types of ~~FGPAs~~FPGAs like Xilinx FPGAs. Ex. 1008, 7:18–35.

94. Tetzlaff similarly addresses the use of DLLs and PLLs for clock distribution and phase alignment in FPGA-based serial communication. Tetzlaff discloses “methods and apparatuses for clock distribution, phase alignment, and frequency synchronization via high speed serial I/O transceivers.” Ex. 1006, 4:21–26. Tetzlaff explains that a DLL uses a “delay line and phase detector” to synchronize a reference clock with a feedback clock, and that a PLL can synthesize a clock with a frequency equal to a reference clock multiplied by a ratio of selectable integers. Ex. 1006, 2:26–55, 6:15–21, claim 8. Tetzlaff further teaches that a PLL can maintain “a selectable phase offset between the synthesized clock and the reference clock,” and that a DLL can provide “a selectable quantized delay.” Ex. 1006, 10:24–33, 10:59–64.

95. These clock management resources were integral components of the FPGA silicon. Tetzlaff notes that FPGAs include “programmable tiles” comprising configurable logic blocks (CLBs), block RAM, multipliers, DSPs, processors, and “clock managers” including DLLs. Ex. 1006, 1:28–33. Lee’s digital frequency synthesis circuit employs a fractional counter (e.g., a Modulus_64 counter) that provides “finer granularity of frequency control, thereby avoiding an overshoot condition”—reducing period jitter to sub-picosecond levels. Ex. 1008, 3:47–4:7.

Lee further discloses a “phase deviation filter” that “counts concurrences when the output of the phase detector 510 is the same” as the reference clock to provide a measure of phase deviation, dynamically adjusting the proportional and integral gain values to achieve low jitter and rapid settling. Ex. 1008, 5:1–64. These were not novel mechanisms—they were standard, commercially available building blocks that a person of ordinary skill in the art would routinely employ.

C. Clock Synchronization and Phase Alignment Between Receiver and Transmitter

96. A central challenge in high-speed FPGA transceiver design—one that was well-recognized and addressed in the prior art long before 2019—is synchronizing the clock domains of the receiver and transmitter paths to minimize latency. When data arrives at an FPGA’s receiver, a clock and data recovery (CDR) circuit extracts a recovered clock (often called RXRECCLK) from the incoming data stream. This recovered clock is phase-aligned to the incoming data but is independent of the transmitter’s clock. If the receiver and transmitter clocks are not synchronized, a clock domain crossing (CDC) mechanism—typically implemented as an elastic buffer or FIFO—is required to safely transfer data between domains. Such CDC mechanisms introduce additional latency.

97. The prior art recognized this latency problem and provided solutions to eliminate or reduce clock domain crossing delays. Giordano describes the GTP transceiver architecture in which a PLL is shared between both the receiver and

transmitter within the same tile. Ex. 1005, 194-95. Giordano implements phase alignment to bypass the TX FIFO buffer, eliminating clock domain crossing. Ex. 1005, 195, 197. By aligning the phase of the transmitter user clock (TXUSRCLK) with the internal parallel clock, Giordano eliminates the need for a TX elastic buffer that would otherwise introduce variable latency due to clock domain crossing. *Id.*

98. The Virtex-5 RocketIO GTP Transceiver User Guide documents these phase alignment procedures in detail. For the receive path, the RX phase alignment allows bypassing the RX elastic buffer, resulting in “SIPO parallel clock phase matching RXUSRCLK phase” and “[n]o phase difference between XCLK and RXUSRCLK.” Ex. 1009, 180. Such receiver phase alignment and bypass of clock domain crossing circuitry is depicted in Figure 7-22 of the GTP Transceiver User Guide as seen below:

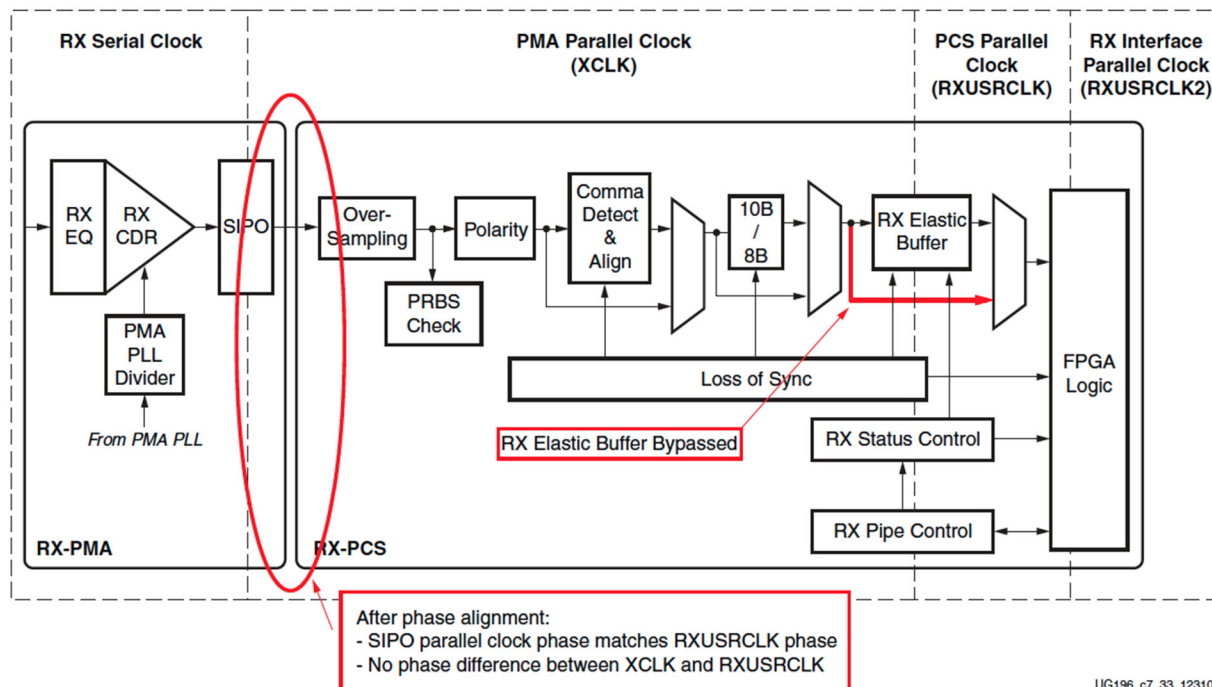


Figure 7-22: Using Phase Alignment

99. For the transmit path, TX phase alignment similarly allows bypassing the TX buffer. Ex. 1009, 121. These mechanisms were documented standard features of the hardware, available to any designer using these FPGAs.

100. Giordano further explains the relationship between the recovered receiver clock and the transmitter reference clock: “The clock recovered by the receiver cannot be directly used as a reference clock for the re-transmission. In fact, the internal PLL is shared among the transmitters and receivers in the same tile and at the receiver tile it is already locked to the seed clock. Moreover, the recovered clock might also require to be filtered in order to match the jitter specifications for the GTP reference clock.” Ex. 1005, 198. To address this, Giordano uses a DLL “external to the GTP” to multiply the reference clock frequency by 4, “thus

providing TXUSRCLK.” Ex. 1005, 197–198. This is precisely the type of clock conditioning that Tetzlaff’s DLL and PLL circuits are designed to perform.

101. Raymond addresses the same clock synchronization challenge in the context of ultra-low latency financial trading applications. Raymond discloses a receiving-side PLL that “up-converts the recovered clock signal” to provide transmitting driving clock signals. Ex. 1007, 9:52–66. Critically, Raymond teaches that the “receiving-side driving signals and transmitting-side driving signals have the same clock frequencies.” Ex. 1007, 2:29–31. This frequency relationship between receiver and transmitter clocks is what enables data to pass between the receiver and transmitter paths without requiring an asynchronous FIFO or elastic buffer for clock domain crossing.

D. Application to Financial Trading Systems

102. The application of FPGA-based low-latency serial communication to financial trading was well-established before the priority date. Raymond expressly teaches an “ultra-low latency communication” device for “electronic trading” applications. Ex. 1007, 1:18–24. Raymond describes processing “financial trading data” from the market exchange, and uses for “market data feed handler” and “market data feed distribution systems.” Ex. 1007, 4:52–59, 13:61–64. Raymond further describes that outgoing data is generated after “logical processing (e.g. applying financial trading algorithms).” Ex. 1007, 11:57–60.

103. As such, the motivation to minimize latency in financial trading systems was universally understood in the art. Accordingly, a person of ordinary skill in the art working on financial trading FPGA systems would have been strongly motivated to eliminate unnecessary sources of latency, including clock domain crossing delays introduced by elastic buffers and FIFOs. The prior art demonstrates that the technical means to accomplish this—phase alignment, PLL-based clock synthesis, and FIFO bypass—were all well-known and documented.

E. Summary of State of the Art

104. In sum, the '587 patent disclosed and claims components and methods well known in the art at the time of its claimed invention. The '587 patent itself admits that FPGAs and PLLs were commercially available before the patent, and that FPGAs were already used in financial trading applications. The technique of phase-matching FPGA receiver and transmitter clocks to avoid clock domain crossing was well-known and documented in the prior art. Ex. 1005, 195, 197; Ex. 1009, 121, 180. These were standard operating procedures for engineers working with these FPGAs. All of the components at issue—DLLs, PLLs, SERDES blocks, phase detectors, CDR circuits, and elastic buffers—were standard building blocks available in commercial FPGA devices. Their implementation in FPGA fabric requires only programming existing silicon resources according to well-documented procedures, such as applying PLL-based clock synthesis, phase alignment, FIFO

bypass to known hardware like Xilinx FPGAs with integrated transceivers to achieve low-latency data processing without clock domain crossing delays. A person of ordinary skill in the art could have combined these concepts using ordinary creativity with a reasonable expectation of success because the art was well aware of the same technical problems and used the same hardware platform contemplated by the '587 patent.

IX. DETAILED ANALYSIS OF INVALIDITY GROUNDS

105. I have analyzed all Challenged Claims of the '587 patent, and it is my opinion that claims 1 and 4–12 are rendered obvious by Giordano in view of Tetzlaff (Ground 1), claims 1–12 are rendered obvious by Raymond in view of Tetzlaff (Ground 2), claims ~~2–3~~1–12 are rendered obvious by Giordano in view of Tetzlaff further in view of Raymond (Ground 3), claims 1 and 4–12 are rendered obvious by Giordano in view of Lee (Ground 4), ~~and~~ claims ~~2–3~~1–12 are rendered obvious by Giordano in view of Lee further in view of Raymond (Ground 5), claims 1 and 4–12 are rendered obvious by Giordano in view of Tetzlaff further in view of Virtex-UG (Ground 6), claims 1 and 4–12 are rendered obvious by Giordano in view of Lee further in view of Virtex-UG (Ground 7), claims 1–12 are rendered obvious by Giordano in view of Tetzlaff, Virtex-UG, and Raymond (Ground 8), and claims 1–12 are rendered obvious by Giordano in view of Lee, Virtex-UG, and Raymond (Ground 9).

A. Ground 1: Claims 1 and 4–12 Are Rendered Obvious By Giordano in view of Tetzlaff

1. Prior Art Overview

106. As discussed above in Sections VII.C and VIII, the prior art references Giordano and Tetzlaff and the state of the art at the time of the claimed invention help form my detailed analysis of Ground 1. I also provide the following descriptions below that further form the basis of my analysis.

107. Giordano’s FPGA system is comprised of two portions—Xilinx Virtex 5 FPGA “RocketIO” GTP transceiver (which has a receiver side and transmitter side) and the FPGA Fabric. Ex. 1005, 195. The GTP transceiver includes preprogrammed, fixed hardware components (such as dedicated pins and built-in analog circuits). *Id.*, 4; Ex. 1009, 50 (discussing the “analog nature of some blocks” within the GTP transceiver). These fixed components, commonly referred to as hard-macro tiles, cannot be modified or changed like FPGA programmable logic, but the user may still configure operating parameters and settings. *Id.*, 101 (noting that the FPGA logic may provide configuration data to the GTP transceiver). In contrast, the FPGA Fabric comprises programmable hardware that can be modified by the user. For example, Giordano discloses user-programmed hardware blocks such as “Comma Detector and Aligner” off the GTP transceiver and on the FPGA Fabric. Ex. 1005, 198.

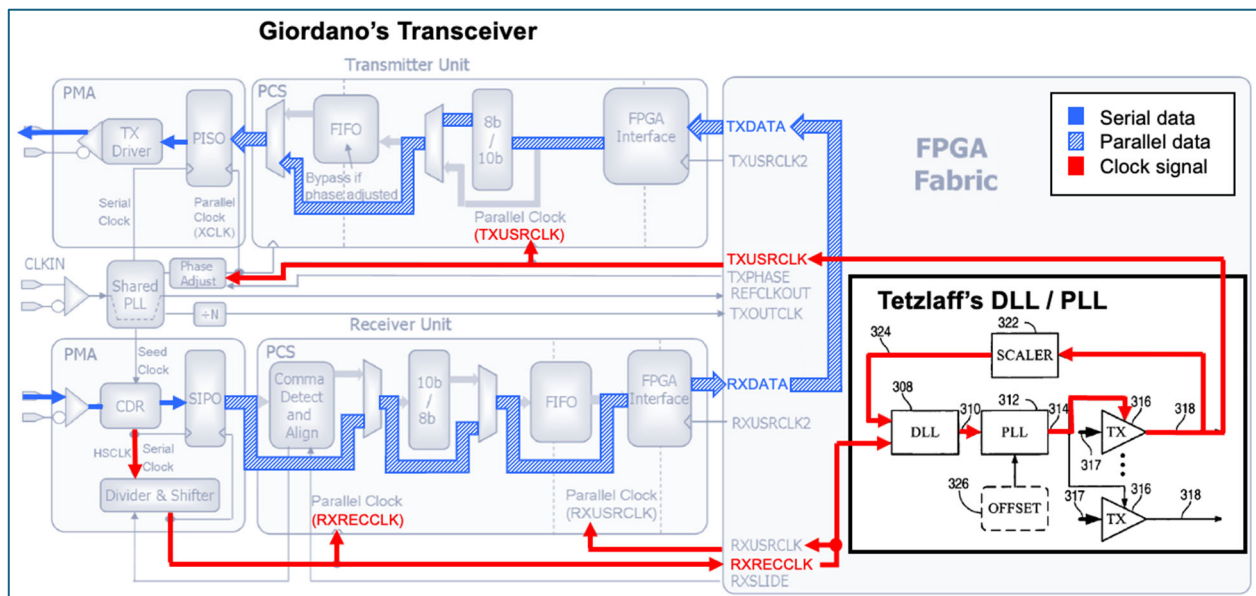
108. The mix of fixed and programmable hardware is common and well-understood in FPGAs. For example, Giordano designed and implemented a custom Comma Detector and Aligner in the programmable fabric because the GTP's built-in comma detector (which is a fixed block inside the GTP transceiver) only operates in certain modes, and not the mode in which Giordano was interested. Ex. 1005, 198. In addition, Giordano discloses implementing DLL blocks "external to the GTP" transceiver, *i.e.*, on the FPGA Fabric. Ex. 1005, Fig. 5. The flexibility of the FPGA Fabric allows for customizing hardware as needed.

109. Indeed, such FPGA hardware programming is precisely how the '587 patent suggests implementing the claimed invention. The '587 patent discusses how the phase detector, phase controller, delay locked loop (DLL) components, and phase locked loop (PLL) are implemented in a "FPGA Core." Ex. 1001, Figs. 3A–F, 4A–C, 5A–F. The '587 patent describes the FPGA Core as including logic elements, DSP blocks, clock distribution components, and memories, which corresponds to the FPGA Fabric of Giordano. Ex. 1001, 26:44–50. Thus, the programming of Tetzlaff's DLL/PLL in Giordano's FPGA Fabric mirrors the '587 patent's programming of its FPGA Core.

110. The proposed prior art combination takes the Xilinx "RocketIO" GTP transceiver and FPGA Fabric disclosed in Giordano, and programs parts of said FPGA Fabric to implement the PLL and DLL from Tetzlaff. Tetzlaff is a Xilinx

patent, which expressly incorporates by reference “RocketIO™ Transceiver User Guide UG024 Dec. 9, 2004, of Xilinx Inc.,” and other works by Xilinx. Ex. 1006, 7:25–36. Given the technical similarity, a POSITA would have understood how to program specific custom hardware (such as that of Tetzlaff) on the Xilinx FPGA Fabric disclosed in Giordano.

111. The annotated image below, based on Figure 1 of Giordano and Figure 3 of Tetzlaff with modifications, depicts this proposed combination:



112. Giordano’s GTP transceiver generates a receiver side clock signal, RXRECCLK, from clock extracted from incoming serial data, and provides RXRECCLK to Tetzlaff’s DLL/PLL implemented in the FPGA Fabric. The DLL/PLL block creates a transmitter side clock signal TXUSRCLK that is phase-matched to RXRECCLK, and sent to the transmitter of the GTP transceiver.

a) Motivation to Combine Giordano and Tetzlaff

113. The '587 patent sought to address the synchronization of receiving side and transmitting side clock signals within the FPGA. Ex. 1001, 1:43–46. Giordano and Tetzlaff demonstrate how a POSITA would have been motivated to combine the two references to address that problem.

114. A POSITA would have been motivated to look to Tetzlaff to combine with Giordano's system as both are based on Xilinx FPGAs. Giordano's architecture is built on the "RocketIO" GTP transceiver of the Xilinx Virtex 5 FPGA family, and references the RocketIO GTP Transceiver User Guide. Ex. 1005, 194, 197, 199. Tetzlaff, which is a patent assigned to Xilinx, Inc., also expressly incorporates by reference "RocketIO™ Transceiver User Guide UG024 Dec. 9, 2004, of Xilinx Inc.," and other work by Xilinx. Ex. 1006, 7:25–36. A POSITA working with Xilinx GTP transceivers, as Giordano does, would naturally consult Xilinx's own materials, such as Tetzlaff and RocketIO Transceiver User Guide, for guidance on clock distribution and phase alignment using those same transceiver blocks. In addition, a POSITA would have understood that incorporating Tetzlaff's PLL feedback and phase alignment teachings into Giordano requires no novel hardware adaptation—only configuring existing FPGA programmable logic according to Tetzlaff's teachings.

115. Giordano identifies the need that Tetzlaff's PLL and phase alignment teachings fulfill. While Giordano's fixed latency architecture achieves deterministic

(i.e., fixed) phase relationships so that the receiver clock always has “the same phase with respect to the transmit clock,” Giordano identifies gaps for directly using this receiver clock for the transmitter. Ex. 1005, 198. Giordano states:

The clock recovered by the receiver cannot be directly used as a reference clock for the re-transmission. In fact, the internal PLL is shared among the transmitters and receivers in the same tile and at the receiver tile it is already locked to the seed clock. Moreover, the recovered clock might also require to be filtered in order to match the jitter specifications for the GTP reference clock

Ex. 1005, 198.

116. Giordano thus recognizes that a jitter-filtering PLL is beneficial to condition the recovered receiver clock to serve as a transmitter reference clock. A POSITA reading Giordano would therefore be motivated to look to other references, and particularly to Xilinx’s own materials (such as Tetzlaff, which is a Xilinx patent), for a PLL-based phase alignment solution.

117. Tetzlaff addresses clock distribution and phase alignment issues in FPGA systems—a solution that a POSITA would have recognized as directly applicable to Giordano’s Xilinx-FPGA-based system. Tetzlaff teaches a two-stage architecture in which a delay lock loop (“DLL”) produces a phase-adjusted reference clock that feeds a PLL, and the PLL generates a transmit clock for a serializing data transmitter. Ex. 1006, 9:29–51. Tetzlaff further teaches that the PLL filters jitter in the reference clock, and produces a transmit clock for which may be aligned “to the

phase of the source clock.” *Id.*, cls. 1, 5; 6:27–47. A POSITA would have recognized that Tetzlaff’s DLL-to-PLL-to-transmitter chain directly addresses Giordano’s identified need: the recovered clock (RXRECCLK) in Giordano can serve as the “source clock” input to Tetzlaff’s DLL/PLL subsystem, which filters its jitter and synthesizes a clean transmit clock to drive the transmitter. Indeed, Giordano already discloses implementing a similar two-stage DLL-PLL structure for other clock signals. A DLL on the FPGA Fabric (and off the GTP transceiver) receives the REFCLKOUT from the GTP transceiver and generates TXUSRCLK based on REFCLKOUT. Ex. 1005, 197. This ensures REFCLKOUT and TXUSRCLK are phase matched. *Id.* TXUSRCLK is then sent back to the GTP transceiver and provided to the PLL on the GTP for further processing. Ex. 1005, 195, 197. Thus, implementing Tetzlaff’s DLL/PLL in the FPGA Fabric represents a straightforward extension of techniques already used in Giordano.

118. A POSITA would have naturally looked to Tetzlaff’s clock conditioning circuits to address the jitter-filtering need that Giordano identifies, because both references concern the same hardware platform and Tetzlaff’s solution is specifically designed for the purpose Giordano requires. As such, a POSITA would have been motivated to combine Giordano and Tetzlaff as they share the same technical objective to achieve controlled clock relationships, with deterministic fixed phase differences, throughout a transceiver in FPGAs. This shared goal supports that a

POSITA would have been motivated to combine the references. Tetzlaff identifies that clock skew in synchronous digital systems—when “rising edges of a clock signal in different parts of the system may not be synchronized”—can cause systems to malfunction. Ex. 1006, 2:9–16. Tetzlaff addresses this by “clock distribution, phase alignment, and frequency synchronization via high speed serial I/O transceivers.” *Id.*, 4:24–26. Giordano addresses the same problem, that is, “timing synchronization and clock distribution applications,” and proposes to address this by generating a fixed-latency serial link in FPGAs. Ex. 1005, 194. In other words, the proposed solution is to align the phase between two signals to a fixed value, *i.e.*, the same phase offset. *Id.*, 194, 197.

119. A POSITA would have had a reasonable expectation of success in implementing Tetzlaff’s teaching to Giordano’s system. Giordano’s architecture was physically programmed on the FPGA and tested on off-the-shelf Xilinx evaluation boards, demonstrating a measured link latency through experimentation. Ex. 1005, 197–200. Tetzlaff’s clock distribution architecture uses similar circuit blocks—PLLs, DLLs, PISO serializers, and pattern generators—that Giordano already successfully deploys. Giordano acknowledges the elements used for phase matching (such as DLL) are implemented on the FPGA Fabric, which is circuitry that can be programmed for whatever function the user desires. Ex. 1005, 195, 197; *see also* Ex. 1006, 1:23–48 (discussing FPGA as including programmable tiles). A POSITA

may simply reprogram the FPGA Fabric of Giordano to include the clock path modifications based on the teachings of Tetzlaff. Because both references operate on the same Xilinx transceiver infrastructure, a POSITA would have had high expectations that the combination would function as intended.

**b) Features of Virtex-UG are Inherent to
Giordano, and Giordano Incorporates Virtex-
UG by Reference**

120. Giordano presents a system with the “RocketIO” GTP transceiver of the Xilinx Virtex 5 FPGA family. Ex. 1005, 194, 199. Giordano repeatedly references the RocketIO GTP Transceiver, for example referencing the overall transceiver architecture (*id.*, 195) and taking estimated latency values from the RocketIO GTP Transceiver User Guide (*id.*, 198). Giordano further describes phase alignment of clock signals, and that “the procedure we follow for activating the alignment circuit is the one suggested in the GTP user’s guide.” *Id.*, 197. A POSITA would have known to reference Xilinx documents on the GTP transceiver to understand the relevant inherent features. As such, Giordano inherently discloses the features of Virtex-UG. *Id.* In such a case, the features of the Virtex-UG are necessarily present in Giordano, and I have been informed by counsel that it is proper to use extrinsic evidence (such as the Virtex-UG) to understand said features. *See Telemac Cellular Corp. v. Topp Telecom, Inc.*, 247 F.3d 1316, 1328 (Fed. Cir. 2001) (“[R]ecourse to extrinsic evidence is proper to determine whether a feature, while

not explicitly discussed, is necessarily present in a reference. The evidence must make clear that the missing feature is necessarily present, and that it would be so recognized by persons of ordinary skill in the relevant art.”); *Sage Prods., LLC v. Stewart*, 133 F.4th 1376, 1385 (Fed. Cir. 2025) (“There was likewise nothing improper in the Board relying on evidence outside of the PAR to make findings as to what the skilled artisan would understand the PAR to be disclosing.”); *Monsanto Tech. LLC v. E.I. DuPont de Nemours & Co.*, 878 F.3d 1336, 1345 (Fed. Cir. 2018) (“It is well established that such reliance on extrinsic evidence is proper in an inherency analysis.”); *see also In re Haines*, No. 2024-2338, 2025 WL 1403578, at *2 (Fed. Cir. May 15, 2025) (examiner properly used another publication “as evidence to explain why [the prior art’s] disclosure of a mass spectrometer implicitly discloses ‘obtaining mass spectral channel signals (m/z)’” (cleaned up)).

121. For the same reasons as above, Giordano incorporates Virtex-UG (Ex. 1009) by reference. Ex. 1005, 197 (“the procedure we follow for activating the alignment circuit is the one suggested in the GTP user’s guide.”). Virtex-UG, by incorporation, ~~become~~becomes part of the disclosures of Giordano. *Verizon Connect Inc. v. Omega Patents, LLC*, IPR2023-01162, 2024 WL 714951, at *3, 14-15 (P.T.A.B. Feb. 21, 2024) (the incorporated secondary reference’s disclosures become part of the primary reference); *Paice LLC v. Ford Motor Co.*, 881 F.3d 894, 906-07 (Fed. Cir. 2018) (“[T]he standard of one reasonably skilled in the art should be used

to determine whether the host document describes the material to be incorporated by reference with sufficient particularity.”) (citing *Advanced Display Systems, Inc. v. Kent State University*, 212 F.3d 1272 (2000)).

2. Ground 1 – Analysis

a) Independent Claim 1

[1.pre]: A method for processing a first serial data stream, using a field programmable gate array system, to generate a second serial data stream, wherein the method comprises the steps of:

122. To the extent the preamble of claim [1] is limiting, Giordano discloses [1.pre].

123. Giordano discloses an FPGA system, such as Xilinx Virtex 5, that includes a “serializer-deserializer (SerDes).” Ex. 1005, 194. And as such, Giordano discloses a receiver that takes in serial data, and a transmitter that sends out post-processed serial data.

124. I provide below a discussion of limitations [1.b] and [1.i] further explaining Giordano’s disclosure of first and second serial data streams.

[1.a]: (a) receiving, by a deserializer in a field programmable array, a clock signal;

125. Giordano discloses this limitation. Giordano’s Receiver Unit PMA discloses the claimed “deserializer,” and HSCLK clock discloses the “clock signal.” And the Giordano Receiver Unit PMA “receives” signal comprising HSCLK.

126. Giordano discloses the Gigabit Transceiver with Low Power (“GTP”) transceiver of the Xilinx Virtex 5 FPGA family, where the GTPs are available as fixed, hard-macro tiles (yet still configurable) inside the FPGA chip. Ex. 1005, 194. The GTP transceiver includes a “Receiver Unit,” which includes a Physical Medium Attachment (“PMA”) and Physical Coding Sublayer (“PCS”). *Id.*, 194–195. Giordano points to the PMA sublayer as the component “actually serializing and deserializing the data,” and the PCS layer as “processing” data. *Id.*, 194. The Receiver Unit PMA further includes a “Serial In to Parallel Output (SIPO)” block, which deserializes incoming serial data. *Id.*, 195–196.

127. Giordano’s Receiver Unit receives a clock signal from the incoming serial data stream. The Receiver Unit PMA has a Clock and Data Recovery (“CDR”) block, “which extracts a clock (HSCLK) and uses it to sample” the incoming serial data. Ex. 1005, 195.

128. The foregoing configuration mirrors the ’587 patent, which describes a deserializer with a receiver PMA. Ex. 1001, 53:31–33, Fig. 6E. The ’587 patent describes that this receiver PMA further comprises a CDR block. *Id.* Like Giordano, this CDR block receives serial data and provides clock signals and serial data to the Serial to Parallel block, which converts the incoming serial data to parallel data. Ex. 1001, 53:34–48.

[1.b]: (b) receiving, by the deserializer, the first serial data stream;

129. Giordano discloses this limitation. As I discussed in element [1.a] above, the Receiver Unit PMA in Giordano discloses the claimed “deserializer.” The Receiver Unit PMA in Giordano has a CDR that receives incoming serial data, which discloses the “first serial data stream.” Ex. 1005, 195. The foregoing configuration mirrors the ’587 patent, which describes a receiver PMA with a CDR that receives serial data. Ex. 1001, 53:34–48, Fig. 6E.

[1.c]: (c) generating, by the deserializer, a receiver side clock signal;

130. Giordano discloses this limitation. As I discussed in element [1.a] above, the Receiver Unit PMA in Giordano discloses the claimed “deserializer.”

131. Giordano’s parallel recovered clock (RXRECCLK) discloses the “receiver side clock signal.” The Receiver Unit PMA in Giordano has a CDR that extracts a clock (HSCLK) signal from the incoming serial data. Ex. 1005, 195. The extracted HSCLK is used to “generate a parallel recovered clock (RXRECCLK)” for the SIPO block and Receiver Unit PCS. *Id.* This RXRECCLK signal discloses the claimed “receiver side clock signal.”

132. The foregoing configuration mirrors the ’587 patent, which describes a CDR that receives serial data, and generates clock signals for other blocks such as the Serial to Parallel block and the receiver PCS. Ex. 1001, 53:34–48, Fig. 6E.

[1.d]: (d) converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams;

133. Giordano discloses this limitation. As I discussed in elements [1.a] and [1.b] above, the Receiver Unit PMA in Giordano disclose the claimed “deserializer,” and the serial data into the PMA discloses the “first serial data stream.”

134. Giordano’s SIPO block output, which becomes RXDATA, discloses the “first plurality of parallel data streams.” The Receiver Unit PMA in Giordano has a “Serial In to Parallel Output (SIPO)” block that converts incoming serial data to parallel data, which discloses this limitation. Ex. 1005, 195.

135. The foregoing configuration mirrors the ’587 patent, which describes a Serial to Parallel block within the Receiver PMA that “converts the serial data into parallel data, transmitting the parallel data to the receiver PCS.” Ex. 1001, 53:44–48, Fig. 6E.

[1.e]: (e) transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first plurality of parallel data streams;

136. Giordano discloses this limitation. As I discussed in elements [1.a] and [1.d] above, the Receiver Unit PMA in Giordano disclose the claimed “deserializer,” and that RXDATA discloses the “first plurality of parallel data streams.”

137. Giordano’s Receiver Unit PCS, FPGA Fabric, and Transmitter Unit PCS together disclose the “computational circuitry in the [FPGA].” Giordano points to the PCS layer as “processing” data, rendering obvious computational circuitry.

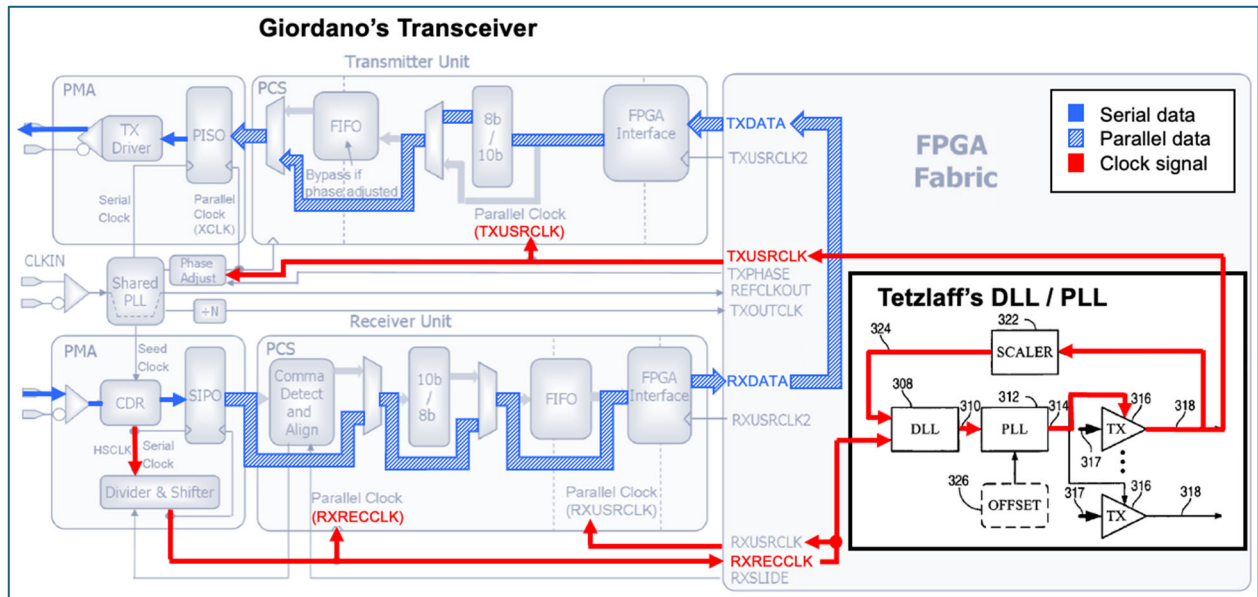
Ex. 1005, 194. Giordano further describes an FPGA Fabric that includes “logic,” which discloses the claimed “computational circuitry in the [FPGA].” Ex. 1005, 195–196. A POSITA would have understood Giordano’s FPGA Fabric includes computational circuitry. Giordano discloses that the output of the SIPO block is transmitted to the Receiver Unit PCS. Ex. 1005, Fig. 1.

138. The foregoing mirrors the ’587 patent, with figures that depict the computational circuitry as “Computation Logic,” and capable of performing logical operation. *E.g.*, Ex. 1001, Figs. 4B–C, Figs. 5A–F, cl. 4.

[1.f]: (f) transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;

139. Giordano in view of Tetzlaff renders obvious this limitation. As I discussed in elements [1.a] and [1.c] above, the Receiver Unit PMA in Giordano discloses the claimed “deserializer,” and the recovered clock signal RXRECCLK from the PMA block discloses “receiver side clock signal.”

140. Tetzlaff’s DLL/PLL programmed in Giordano’s FPGA Fabric renders obvious the claimed “phase lock loop” of the ’587 patent. The reasonably straightforward combination of Giordano and Tetzlaff is depicted in the annotated image below:



141. Giordano discloses that the recovered RXRECCLK (*i.e.*, “receiver side clock signal”) is provided to the FPGA Fabric. Ex. 1005, 195; Ex. 1006, Fig. 3. A POSITA would have understood this RXRECCLK is transmitted to blocks implemented within the FPGA Fabric, including Tetzlaff’s DLL/PLL.

142. As I explained above in Section IX.A.1.a., a POSITA would have known how to implement Tetzlaff’s DLL/PLL in Giordano’s FPGA Fabric. Indeed, Giordano discloses implementing a Comma Detector and Aligner on the FPGA Fabric (*i.e.*, not on the GTP transceiver itself but still on the FPGA) that allows for the recovered RXRECCLK clock to always have “the same phase with respect to the transmit clock.” Ex. 1005, 198. This phase matching is done through modifications to how the RXRECCLK is generated. Ex. 1005, 197–198. It would have been obvious for a POSITA to implement Tetzlaff’s teachings in Giordano’s

FPGA Fabric, which is akin to how Giordano already does so with the Comma Detector and Aligner.

143. While Giordano already phase-aligns the receiver side clock RXRECCLK and transmitter clock TXUSRCLK (as a result of modifying the receiver side clock), Giordano acknowledges that further jitter cleaning would be necessary to use this receiver side clock RXRECCLK for the transmitter. Ex. 1005, 198. Accordingly, Tetzlaff's teachings provide the solution for this further jitter cleaning.

144. In the combination of Giordano and Tetzlaff, Giordano's FPGA Fabric is programmed to create a clock signal path from the receiver clock RXRECCLK to the transmitter clock TXUSRCLK, where Tetzlaff's PLL-based jitter-filtering and phase-alignment feedback loop is inserted between the two clocks. Giordano's RXRECCLK (*i.e.*, "receiver side clock signal") is transmitted to Tetzlaff's DLL/PLL. This DLL/PLL outputs TXUSRCLK (*i.e.*, "transmitter side clock signal"), which is looped back as input to the DLL for phase matching. Thus, resulting in synthesizing a clean transmitter side clock phase matched with the receiver side clock. Ex. 1005, 198.

[1.g]: (g) generating, using the phase lock loop, a transmitter side clock signal;

145. Giordano in view of Tetzlaff renders obvious this limitation. As I discussed in element [1.f] above, Tetzlaff's DLL/PLL implemented in Giordano's

FPGA Fabric renders obvious the claimed “phase lock loop.” That is because the output of the combined DLL/PLL results in a “transmitter side clock signal” TXUSRCLK.

[1.h]: (h) performing, by the computational circuitry, a set of operations on at least a portion of the first plurality of parallel data streams to generate a second plurality of parallel data streams;

146. Giordano discloses this limitation. As I discussed in elements [1.d] and [1.e] above, Giordano’s RXDATA discloses the “first plurality of parallel data streams” and FPGA Fabric discloses the “computational circuitry.” Giordano’s processed parallel data exiting the Transmitter Unit PCS into the PMA discloses the “second plurality of parallel data streams.” Ex. 1005, Fig. 1.

147. For the “set of operations,” Giordano describes an FPGA Fabric that includes “logic,” which discloses this limitation. Ex. 1005, 195–196, 198. Giordano describes programming a Comma Detector and Aligner in the FPGA Fabric, which performs “comma detection.” Ex. 1005, 198. Giordano’s “comma detector looks for commas (e.g., the one included in the K28.5 symbol) and when it finds one it determines how many bit shifts” are “required in order to align the parallel data.” Ex. 1005, 198. A POSITA would have understood this pattern matching operation (to locate comma characters) and computing number of bit shifts for alignment to disclose the claimed “set of operations.”

148. Giordano also points to the PCS layer (both receiver and transmitter sides) as “processing” data, rendering obvious computational circuitry. Ex. 1005, 194. The PCS layer includes 8b/10b encoding / decoding functions, which requires arithmetic calculations hence rendering obvious “set of operations.” Ex. 1005, 197. Tetzlaff further renders obvious “set of operations,” through disclosures that FPGA’s programmable elements include “arithmetic logic.” Ex. 1006, 1:37–40.

149. The foregoing configurations mirrors the ’587 patent, with figures that depict the computational circuitry as “Computation Logic,” and capable of performing logical operation. *E.g.*, Ex. 1001, Figs. 4B–C, Figs. 5A–F, cl. 4.

[1.i]: (i) converting the second plurality of parallel data streams, by a serializer, into a second serial data stream; and

150. Giordano discloses this limitation. As I discussed in element [1.h] above, Giordano’s TXDATA discloses the “second plurality of parallel data streams.”

151. The Transmitter Unit PMA of Giordano, mirroring the receiver side, discloses the claimed “serializer.” Ex. 1005, 195–196. The Transmitter PMA includes a Parallel Input to Serial Output (“PISO”) block that converts parallel data to serial data. Ex. 1005, 195. Giordano discloses that this TXDATA is transmitted to the PISO block through other blocks within the transmitter. Ex. 1005, 195. This serial data output of PISO discloses the “second serial data stream.”

152. The foregoing configuration mirrors the ’587 patent, which describe a serializer with a transmitter PMA with a Parallel to Serial block. Ex. 1001, 54:9–29,

Fig. 7B. Like Giordano, this Parallel to Serial block converts the incoming parallel data to serial data. *Id.*

[1.j]: (j) transmitting, from the field programmable gate array system, the second serial data stream,

153. Giordano discloses this limitation. As I discussed in element [1.i] above, the Transmitter Unit PMA in Giordano discloses the claimed “serializer,” where the output of the PISO block discloses the “second serial data stream.” The Transmitter Unit PMA in Giordano has a TX Driver that sends this serial data out of the FPGA system, which discloses this limitation. Ex. 1005, 195.

154. The foregoing configuration mirrors the ’587 patent, which describes a Transmitter PMA with a Transmitter Buffer that sends out serial data. Ex. 1001, 54:9–29, Fig. 7B.

[1.k]: wherein said method does not use clock domain crossing operations that delay processing of the first set of parallel data streams.

155. Giordano renders obvious this limitation. As I discussed in element [1.g] above, Giordano and Tetzlaff render obvious phase matching the receiver side clock with the transmitter side clock.

156. Relatedly, Giordano discloses bypassing First In First Out (FIFO) buffer, a circuit block allowing safe data transfer between different clock domains (e.g., “data transfers between TXUSRCLK domain and the XCLK domain”). Ex. 1005, 195. A POSITA would have understood FIFO as a clock domain crossing

circuit, and that Giordano's bypassing of the FIFO block renders obvious this limitation.

157. Giordano discloses bypassing FIFO buffer by having the two clock domains have clocks with "the same phase" and "therefore also the same latency for the data." Ex. 1005, 195. Giordano provides at least two exemplary disclosures for such FIFO bypassing, one in the transmitter and one in the receiver, each discussed below.

158. In the transmitter, Giordano discusses two clocks (XCLK and TXUSRCLK) that are adjusted to have the same frequency and phase. Ex. 1005, 195. The transmitter has a FIFO block for crossing between XCLK and TXUSRCLK clock domains. *Id.* Giordano discloses bypassing this transmitter FIFO block by using a phase alignment option provided in Xilinx Virtex-5 FPGA systems. Ex. 1005, 195. Through this built-in phase alignment, XCLK and TXUSRCLK are phase aligned, in which "the transmit FIFO is unnecessary and therefore we bypassed it." Ex. 1005, 197. Giordano further describes that "the procedure we follow for activating the alignment circuit is the one suggested in the GTP user's guide." Ex. 1005, 197. Virtex-UG further discloses this limitation and that the TX phase-alignment circuit is used to align the PMA clock (*i.e.*, XCLK) and the PCS clock (*i.e.*, TXUSRCLK). Ex. 1009, 121. This allows bypassing the TX

Buffer, which is the clock domain crossing circuit. *Id.* The Virtex-UG's disclosures are express and also inherent.

159. The bypass of all clock domain crossing circuits after phase alignment of clock signals, as disclosed in Giordano, renders obvious this limitation, because, at the very least, a POSITA could easily implement this configuration using known technologies.

b) Claim 4

[4]: The method of claim 1, wherein the 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.

160. Giordano in view of Tetzlaff renders obvious this claim. As I discussed in element [1.h] above, Giordano's disclosures on PCS layer processing and FPGA Fabric "logic," including operations such as "comma detection" by a Comma Detector and Aligner, discloses the claimed "set of operations."

161. Giordano's Comma Detector and Aligner implemented in the FPGA Fabric performs pattern matching (detecting comma characters in the raw 10-bit data), comparison (determining the number of bit shifts required), and conditional branching (the even/odd decision logic that either applies RXSLIDE assertions or resets the CDR). Ex. 1005, 198. These are examples of logical operations, which disclose this limitation. In addition, Giordano points to the PCS layer (both receiver and transmitter sides) as "processing" data, rendering obvious computational

circuitry. Ex. 1005, 194. The PCS layer includes 8b/10b encoding / decoding functions, which requires arithmetic calculations hence rendering obvious this claim. Ex. 1005, 197.

162. Tetzlaff discloses that FPGAs may be programmed to include “logic of a user design” including “arithmetic logic.” Ex. 1006, 1:28–33. Tetzlaff further discloses “digital signal processing blocks (DSPs)” as a type of programmable tile in the FPGA which are purpose-built hardware for arithmetic operations. Ex. 1006, 1:28–33. Tetzlaff also teaches that the FPGA includes “dedicated random access memory blocks (BRAM)” as one of the programmable tile types, which discloses the claimed “memory access operation.” Ex. 1006, 1:28–33.

c) Claims 5-7

[5]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed prior to step (g).

[6]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed after step (g).

[7]: The method of claim 1, wherein all of the set of operations performed in step (h) are performed after step (g).

163. Claims [5] through [7] are directed to the order between step (g) (*i.e.*, generating transmitter side clock) and step (h) (*i.e.*, performing computational operations). The combination of Giordano and Tetzlaff renders obvious treating these steps separately, where neither depends on the other’s completion to begin or continue executing. Both steps (g) and (h) require the receiver side clock, but may

be performed independent of each other. As such, Giordano in view of Tetzlaff renders obvious these claims.

164. For step (h), the combination of Giordano and Tetzlaff only requires that the receiver side clock be prepared before the start of computational operations; it is agnostic to whether the transmitter side clock is prepared. The computational circuitry in the FPGA fabric receives deserialized parallel data from the GTP receiver unit and operates on it using the receiver-side clock. Giordano states that “[t]he recovered clock (RXRECCLK) drives the receive clocks (RXUSRCLK and RXUSRCLK2).” Ex. 1005, 198. In Giordano’s architecture, the parallel data exiting the deserializer into the FPGA fabric is clocked by RXUSRCLK, which is sourced from RXRECCLK. *Id.* None of this requires the transmitter side clock to be prepared.

165. Step (g) is a clock synthesis process where the PLL-DLL of Tetzlaff receives RXRECCLK and generates a transmitter-side clock. Hence, the combination of Giordano and Tetzlaff may perform step (g) independent of when the incoming parallel data are processed. Because the two steps are independent, a POSITA would have found it obvious to perform the computations in step (h) prior to, during, or after the transmitter-side clock generation of step (g).

d) Claim 8

[8.pre]: 8. The method of claim 1, wherein step (g) further comprises:

166. Giordano in view of Tetzlaff renders obvious this limitation, as I discussed claim [1] and claim limitation [1.g] above.

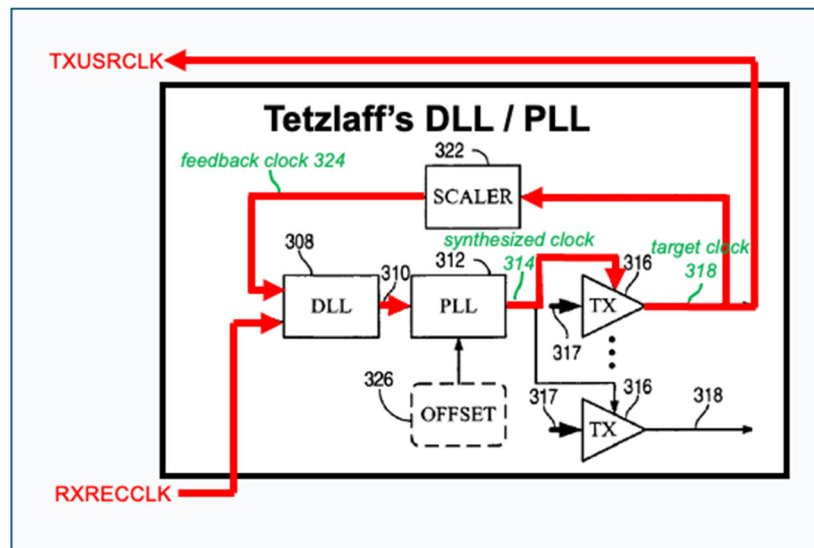
[8.a]: (k) generating, by the phase lock loop, a feedback clock signal associated with the transmitter side clock signal by performing the following steps until a first output of a phase detector of the field programmable gate array system is below a first threshold level:

167. Tetzlaff discloses this limitation, and Giordano in view of Tetzlaff renders obvious this limitation. As I discussed in elements [1.f] and [1.g] above, Tetzlaff's DLL/PLL renders obvious the claimed "phase lock loop," which generates the "transmitter side clock signal" TXUSRCLK.

168. Tetzlaff's feedback clock 324 discloses the claimed "feedback clock signal." Tetzlaff's DLL includes a phase detector that compares the relative phase of the feedback clock 324 and the reference clock, and uses that to control other components to modify the output signal phase. Ex. 1006, 2:49–55. That phase detector and its output therefore disclose and renders obvious the claimed "phase detector" and "first output."

169. Tetzlaff further renders obvious an iterative process until the "first output" "is below a first threshold level" through the clock feedback path and signal that performs multiple rounds of steps until the target clock 318 is phase matched with the incoming reference clock.

170. Annotated image of Tetzlaff's DLL/PLL is below:



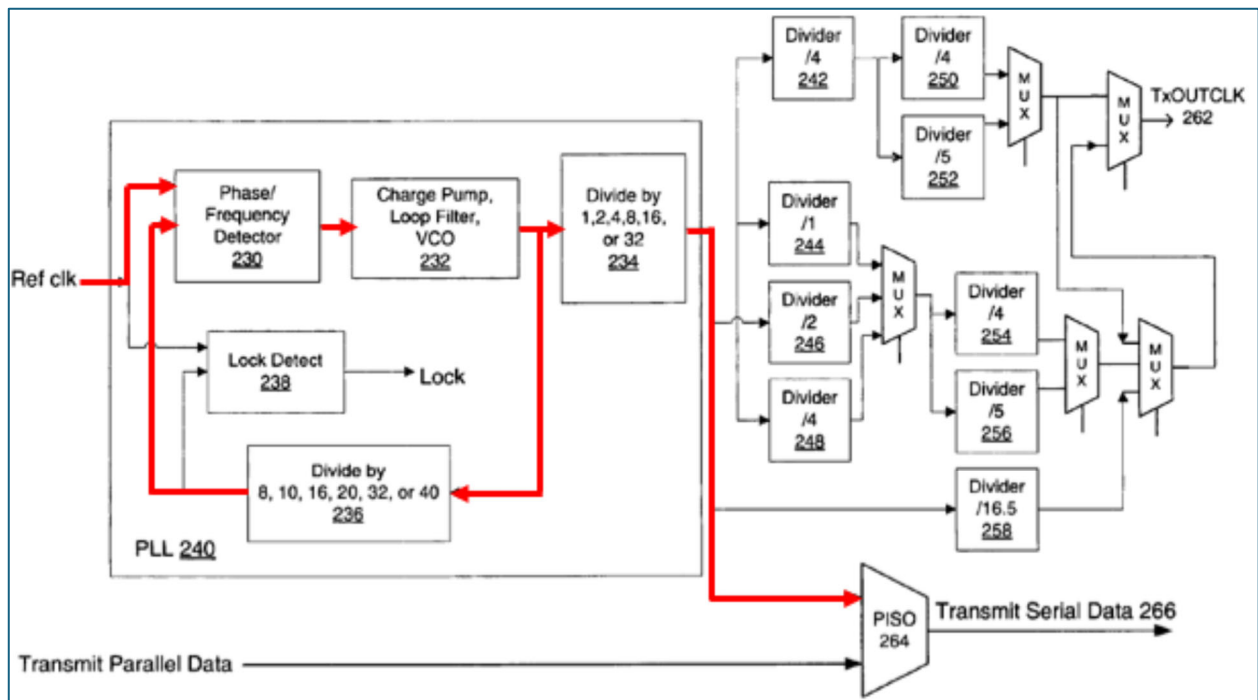
Tetzlaff discloses its “DLL’s phase detector controls a delay line,” where the phase detector can modify the delay applied on the target clock signal “until the reference clock signal is synchronized” with the target clock signal. Ex. 1006, 2:49–55. Since the DLL phase detector “controls” the DLL delay line, interim components and signals for such control, a POSITA would readily recognize Tetzlaff’s DLL discloses the claimed “first output of a phase detector.”

171. Tetzlaff describes this iterative lock acquisition process explicitly, noting that the DLL undergoes “lock acquisition”—defined as “[t]he time in which the DLL is attempting to first synchronize the reference clock signal and skewed clock signal.” *Id.*, 2:65–3:1. This DLL then “monitors the reference clock signal” and the target clock signal, and “adjusts the amount of [] delay to maintain synchronization.” *Id.*, 2:56–59. “The time in which the DLL is attempting to

maintain synchronization is referred to as lock maintenance.” Ex. 1006, 2:67–3:4. Therefore, Tetzlaff teaches that this feedback loop operates iteratively until lock is achieved, hence rendering obvious this limitation.

172. Tetzlaff also discloses that the “feedback clock on line 324 may represent the phase of the target clock on line 318.” Ex. 1006, 10:15–23. Hence Tetzlaff’s feedback clock 324 discloses the claimed “feedback clock signal.” The target clock 318 is the transmitter side clock signal (TXUSRCLK), and the feedback clock 324 is “associated with” TXUSRCLK.

173. In addition, Tetzlaff’s PLL 312 also renders obvious this claim. Tetzlaff discloses “a phase detection circuitry of the PLL,” where said PLL aligns the phases of the synthesized clock with the reference clock. Ex. 1006, 10:24–33. This is shown in Annotated Figure 1B of Tetzlaff:



Here, a phase / frequency detector takes in both a reference signal and the feedback loop signal. In Tetzlaff, “the PLL acts to maintain a selectable phase offset between the synthesized clock and the reference clock.” Ex. 1006, 10:24–33. Tetzlaff discloses that the selectable offset may be added to the synthesized clock “in addition to or instead of the phase adjustment provided by the DLL,” which renders obvious this claim limitation. *Id.*

[8.b]: (1) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal;

174. Tetzlaff discloses this limitation, and Giordano in view of Tetzlaff renders obvious this limitation. Tetzlaff’s DLL delay line and PLL voltage controlled oscillator (VCO) render obvious the claimed “adjustable oscillator.” Tetzlaff’s synthesized clock 314 discloses the claimed “second clock signal.”

175. Tetzlaff discusses that conventional DLLs typically include a phase detector that compares the phase of two signals. Ex. 1006, 2:26–55. That “phase detector is then used to detect the associated phase differences and to control the delay line” to phase match the reference and output clocks. Ex. 1006, 2:38–41.

176. Tetzlaff also discloses a charge pump, loop filter, VCO within the PLL block, which discloses the claimed “adjustable oscillator.” Ex. 1006, 6:27–47; Fig. 1B. This VCO “or similar variable frequency oscillator” aligns the phases of the synthesized clock with the reference clock. *Id.*, 10:24–33.

177. For the phase matching capabilities, the aforementioned DLL delay line and PLL VCO are exemplary disclosures that render obvious the claimed “adjustable oscillator.” Referring to the Tetzlaff image above in element [8.a], after the DLL 308 and emanating from PLL 312, is a synthesized clock 314. Ex. 1006, 9:34–51. This synthesized clock 314 discloses the claimed “second clock signal.”

[8.c]: (2) generating, by the phase detector, the first output based on a comparison of the receiver side clock signal and the feedback clock signal obtained from the second clock signal;

178. Tetzlaff discloses this limitation. As I discussed in elements [8.a] and [8.b] above, Tetzlaff’s DLL/PLL discloses the “phase detector” and the “first output,” its feedback clock 324 discloses the claimed “feedback clock signal,” and its synthesized clock 314 discloses the “second clock signal.”

179. Tetzlaff's DLL/PLL discloses comparing the phase of the receiver side clock (RXRECCLK) with that of the feedback clock 324, rendering obvious this limitation. Tetzlaff discusses that conventional DLLs typically include a delay line and a phase detector, and receives a reference clock signal and drives an output clock signal. Ex. 1006, 2:26–55. “A phase detector is then used to *detect the associated phase differences* and to *control* the delay line” which allows phase matching between the reference and output clocks. Ex. 1006, 2:38–41. This necessitates generating a control signal for the delay line. The DLL/PLL's detection of the phase difference between signals, and using that difference to generate a control signal for the delay line, renders obvious “generating...the first output.”

[8.d]: (3) transmitting, from the phase detector to a phase controller of the field programmable gate array system, the first output;

180. Tetzlaff discloses this limitation. As I discussed in elements [8.a] and [8.c] above, Tetzlaff's DLL renders obvious the “phase detector” and “first output.” And Tetzlaff's phase adjustments in the DLL/PLL renders obvious this limitation [8.d].

181. Tetzlaff discloses that “[a] phase detector is ... used to detect the associated phase differences and to *control* the delay line,” which allows phase matching between the reference and output clocks. Ex. 1006, 2:38–41. Since the DLL phase detector “controls” the DLL delay line, interim components and signals

for such control, a POSITA would readily recognize Tetzlaff's DLL discloses the claimed "first output of a phase detector."

182. Indeed, Tetzlaff discloses that "the actual control mechanism for the DLL can differ" as long as "the DLL monitors" the reference and output clocks, and "adjust the amount of propagation delay to maintain synchronization." Ex. 1006, 2:49–59. Tetzlaff discloses that DLL "signal control is achieved" by its embodiments. *Id.*, 4:36–38. The foregoing configuration mirrors how the '587 patent suggests programming the FPGA for "event-based system to run the control algorithm for either the PLL or DLL paths on the device." Ex. 1001, 55:45–55. And then using the detected phase difference to control the DLL delay line renders obvious "transmitting...the first output" to the "phase controller."

[8.e]: (4) determining, by the phase controller, interim adjustment information based on the first output; and

183. Tetzlaff discloses this limitation. As I discussed in element [8.d] above, Tetzlaff's DLL renders obvious the "phase controller." Then Tetzlaff's phase adjustments in the DLL/PLL renders obvious this limitation [8.e].

184. Tetzlaff discloses that "[a] phase detector is ... used to detect the associated phase differences and to *control* the delay line" which allows phase matching between the reference and output clocks. Ex. 1006, 2:38–41. Since the DLL phase detector "controls" the DLL delay line, interim components and signals for such control, a POSITA would readily recognize Tetzlaff's DLL discloses the

claimed “first output of a phase detector.” The control of the DLL delay line to phase match clocks, using information from the phase detector, necessarily requires information for said control, which renders obvious the “interim adjustment information.”

185. Tetzlaff further describes this phase matching process as iterative, noting that “[t]he time in which the DLL is attempting to first synchronize the reference clock signal and skewed clock signal, is referred to as lock acquisition.” Ex. 1006, 2:65–3:4. The iterative process of Tetzlaff renders obvious the “interim” nature of this “adjustment information.”

[8.f]: (5) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;

186. Tetzlaff discloses claim [8.f]. As I discussed in elements [8.b], [8.c], and [8.e] above, Tetzlaff’s DLL delay line and/or PLL VCO renders obvious the “adjustable oscillator,” and its DLL renders obvious the “phase controller” and “interim adjustment information.”

187. Tetzlaff’s phase adjustments in the DLL/PLL renders obvious this limitation [8.e]. Tetzlaff discloses that the DLL’s phase detector “control[s]” the delay line to synchronize the reference clock with the feedback signal. Ex. 1006, 2:26–55. Since the DLL phase detector “controls” the DLL delay line, interim components and signals for such control, a POSITA would readily recognize Tetzlaff’s DLL discloses the claimed “first output of a phase detector.”

188. Control of DLL delay line necessitates sending information to said delay line for such control. As the DLL delay line renders obvious the “adjustable oscillator,” Tetzlaff’s control of delay line then renders obvious this limitation.

[8.g]: wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information.

189. Tetzlaff discloses this limitation. As I discussed in elements [8.b] and [8.e] above, Tetzlaff’s DLL delay line and/or PLL VCO renders obvious the “adjustable oscillator,” its synthesized clock 314 discloses the “second clock signal,” and its DLL inherently discloses the “interim adjustment information.”

190. Referring to the Tetzlaff image above in element [8.a], after the DLL 308 and PLL 312 is a synthesized clock 314. Ex. 1006, 9:34–51. As such, Tetzlaff discloses that the DLL delay line and PLL VCO adjust the phase for the signals along the feedback data path, resulting in adjusting the synthesized clock 314.

191. Tetzlaff’s DLL phase detector “control[s]” the delay line to synchronize the reference clock with the feedback signal. Ex. 1006, 2:26–55. Since the DLL phase detector “controls” the DLL delay line, interim components and signals for such control, a POSITA would readily recognize Tetzlaff’s DLL discloses the claimed “first output of a phase detector.” As such, Tetzlaff’s delay line modifying the feedback signal renders obvious this limitation.

192. Additionally, Tetzlaff’s PLL “acts to maintain a selectable phase offset between the synthesized clock and the reference clock.” Ex. 1006, 10:24–33.

Tetzlaff discloses that the selectable offset may be added to the synthesized clock “in addition to or instead of the phase adjustment provided by the DLL.” *Id.* This phase modification of the feedback signal renders obvious this limitation.

[8.h]: wherein steps (1) through (5) are repeated until the first output of the phase detector is below the first threshold level, and

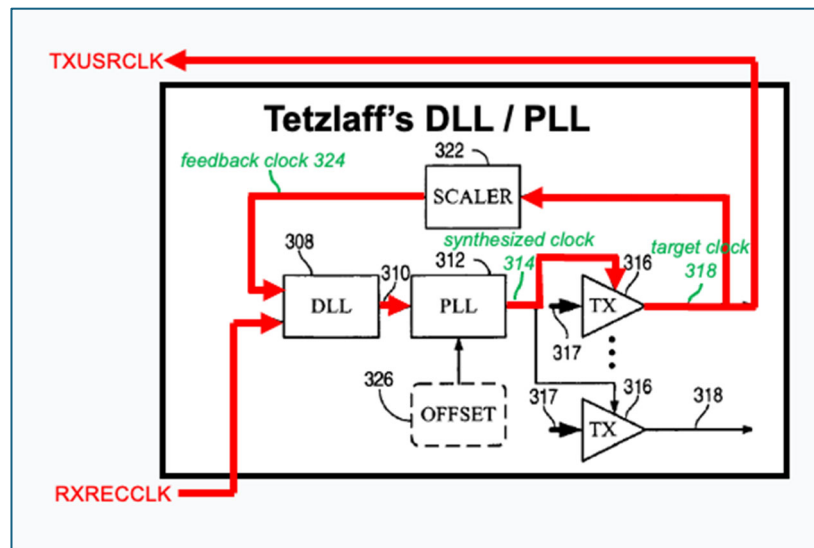
193. Tetzlaff discloses this limitation. As I discussed in element [8.a] above, Tetzlaff’s DLL (which includes a phase detector) renders obvious the “first output.” Tetzlaff’s DLL/PLL discloses this limitation [8.h].

194. Tetzlaff discloses the DLL producing the transmit clock “by phase shifting the synthesized clock [] to match the phase of the feedback.” Ex. 1006, 10:55–59. Tetzlaff describes this iterative lock acquisition process explicitly, noting that “[t]he time in which the DLL is attempting to first synchronize the reference clock signal and skewed clock signal, is referred to as lock acquisition. The time in which the DLL is attempting to maintain synchronization is referred to as lock maintenance.” Ex. 1006, 2:65–3:4. Tetzlaff teaches that this feedback loop operates iteratively until lock is achieved, hence rendering obvious this limitation.

[8.i]: wherein the transmitter side clock signal is associated with the second clock signal.

195. Tetzlaff discloses this limitation. As I discussed in element [8.b] above, the synthesized clock 314 discloses the “second clock signal.” Tetzlaff’s DLL/PLL discloses this limitation [8.i].

196. An annotated image of Tetzlaff's DLL/PLL is below:



The target clock 318 is the transmitter side clock signal (TXUSRCLK). Tetzlaff discloses that the target clock 318 is used to generate the feedback clock 324, which in turn is used to generate the synthesized clock 314. Ex. 1006, 9:61–10:14. This relationship supports that the target clock 318 is “associated with” synthesized clock 314, rendering obvious this limitation. The feedback clock 324, synthesized clock 314, and target clock 318 are generated by Tetzlaff’s DLL/PLL, mirroring how the ’587 patent’s “associated” signals—“feedback clock signal,” “second clock signal,” and “transmitter side clock signal”—are all generated by the claimed “phase lock loop.” Ex. 1001, cl. 8.

e) Claim 9

[9]: The method of claim 1, wherein the first plurality of parallel data streams has the same number of data streams as the second plurality of parallel data streams.

197. Giordano's disclosure of 8b/10b encoding / decoding renders obvious this claim. Giordano discloses using 8b/10b encoding, where the SIPO deserializes incoming stream 10-bit parallel words, then decoded into 8-bit parallel data words. Ex. 1005, 195. On the transmit side, the FPGA fabric supplies 8-bit parallel data words to the GTP, which encodes them using the internal 8b-to-10b encoder before the PISO serializes the resulting 10-bit words. Ex. 1005, 195, 197. A POSITA would have understood this to result in ten parallel data streams on both the receiver and transmitter side, with the same number of data streams.

198. Virtex-UG, which Giordano incorporates by reference, confirms the equal data streams in the receiver and transmitter. The data width is controlled by a single shared attribute, INTDATAWIDTH. Ex. 1009, 104. INTDATAWIDTH controls the width of the internal data, whether or not the 8B / 10B encoder is enabled. *Id.* "Port widths can be 8 bits, 10 bits, 16 bits, and 20 bits." *Id.* Because this is a shared attribute that controls both TX and RX, the internal parallel data width is necessarily the same on both sides—the first plurality of parallel data streams and the second plurality of parallel data streams always have the same number of data streams.

199. Virtex-UG further describes a user-facing data width (TXDATAWIDTH and RXDATAWIDTH) that can double the parallel interface to 16 or 20 bits. Ex. 1009, 106, 168. Yet RXDATAWIDTH and TXDATAWIDTH are set symmetrically, and both paths are given the same width. *Id.*, 76.

f) Claim 10

[10]: The method of claim 1, wherein the first plurality of data streams and the second plurality of data streams comprise 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; (viii) eighty (80) data streams; (ix) one hundred twenty-eight (128) data streams; and (x) one hundred sixty (160) data streams.

200. Giordano discloses this limitation. As I discussed in claim [9] above, Giordano renders obvious an internal parallel data width to be the same on both receiver and transmitter sides. Giordano also discloses the use of 8-bit parallel words, and further port widths of “8 bits, 10 bits, 16 bits, and 20 bits.” Ex. 1005, 195; Ex. 1009, 104. Giordano’s example parallel data pathwidths disclose this claim. For example, the use of 8, 10, 16, and 20 data streams.

g) Claim 11

[11]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and phase.

201. Giordano in view of Tetzlaff renders obvious this claim. As I discussed in claim [1] above, in the combination of Giordano and Tetzlaff, RXRECCLK

renders obvious the receiver side clock signal, and TXUSRCLK renders obvious the transmitter side clock signal.

202. As to frequency alignment, Tetzlaff teaches that the PLL synthesizes a clock “having a frequency corresponding to a frequency of the reference clock multiplied by a ratio of selectable integers.” Ex. 1006, cl. 8. When a ratio of 1 is selected, the PLL output (e.g., TXUSRCLK) has the same frequency as its reference clock (e.g., RXRECCLK).

203. As to phase alignment, both Giordano and Tetzlaff disclose two clocks having the same phase. Giordano uses “the phase alignment circuit, which aligns the phase of XCLK to the one of TXUSRCLK. Since TXUSRCLK is generated with a deterministic phase from the reference clock, this leads to have XCLK, TXUSRCLK and the reference clock running [*i.e.*, RXRECCLK] with a deterministic fixed phase.” Ex. 1005, 197. This “deterministic fixed phase” would be understood as a constant difference, including zero, meaning the two signals have same phase.

204. Tetzlaff discloses how DLLs may use a delay line to control the phase difference between two signals, including making the two “synchronized.” Ex. 1006, 2:26–41, 2:49–55. In addition, Tetzlaff discloses that “PLL acts to maintain a selectable phase offset between the synthesizesynthesized clock and the reference clock.” Ex. 1006, 10:24–33. This includes “phase shifting the synthesized clock”

to “match the phase of the feedback.” *Id.*, 10:55–59. The combination of Giordano and Tetzlaff would therefore be able to generate a TXUSRCLK signal that has the same frequency and phase as RXRECCLK. The combined system therefore renders obvious this limitation.

h) Claim 12

[12]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and different phases.

205. Giordano in view of Tetzlaff renders obvious this claim. As I discussed in claim [11] above, the combination of Giordano and Tetzlaff would be able to generate a TXUSRCLK signal that has the same frequency as RXRECCLK. In addition, the same combination would be able to generate a TXUSRCLK signal that has the same frequency but different, fixed phase to RXRECCLK.

206. Giordano uses “the phase alignment circuit, which aligns the phase of XCLK to the one of TXUSRCLK. Since TXUSRCLK is generated with a deterministic phase from the reference clock, this leads to have XCLK, TXUSRCLK and the reference clock running [*i.e.*, RXRECCLK] with a deterministic fixed phase.” Ex. 1005, 197. This includes a non-zero fixed phase difference.

207. Tetzlaff discloses that the phase shift in the DLL is capable of dynamically compensating for delays with “one of 256 possible phase shifts,” and may provide “a selectable quantized delay” of phase. Ex. 1006, 10:59–64. The combination of Giordano and Tetzlaff would therefore be able to generate a

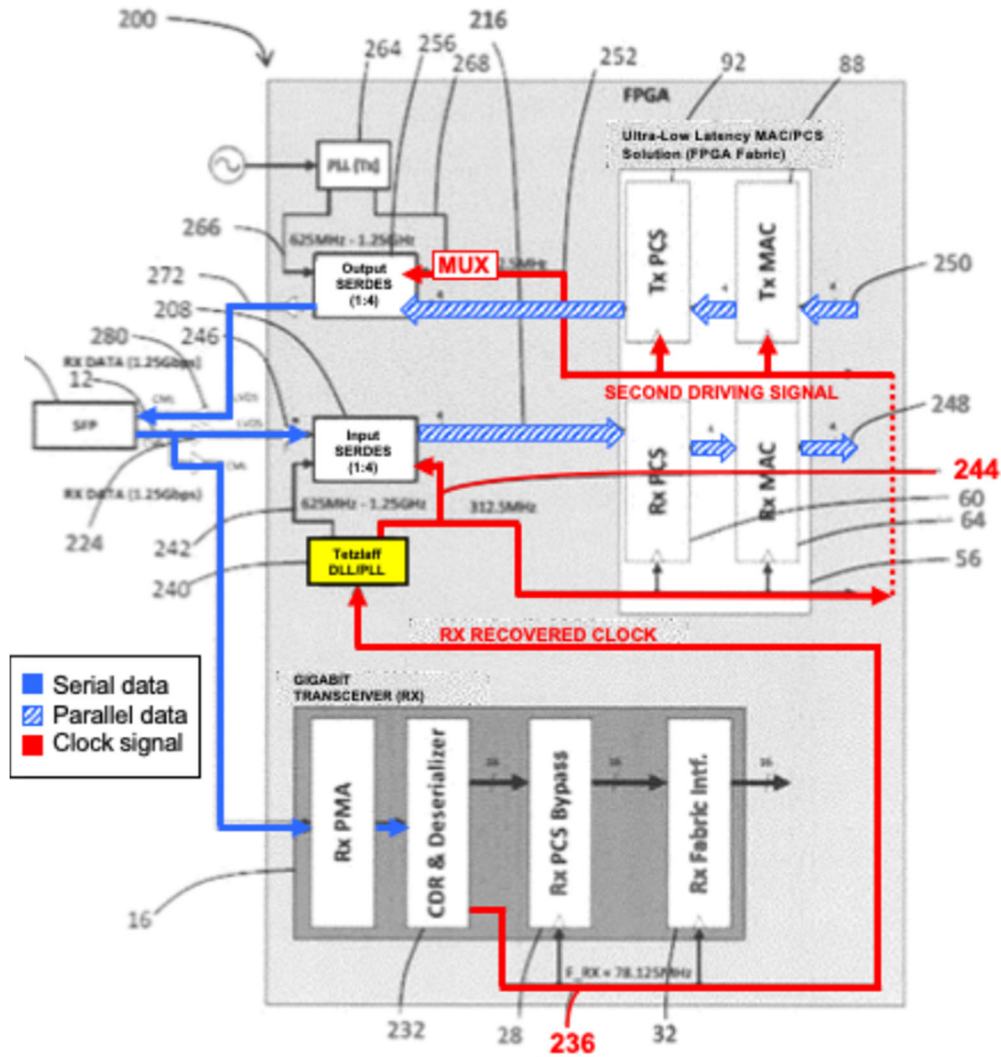
TXUSRCLK signal that has the same frequency yet different phase from RXRECCLK. The combined system therefore renders obvious this limitation.

B. Ground 2: Claims 1–12 Are Rendered Obvious By Raymond in view of Tetzlaff

208. Raymond in view of Tetzlaff renders obvious each and every element of claims 1–12.

1. Prior Art Overview

209. The annotated image below, Figure 2 of Raymond modified in light of the teachings of Tetzlaff, depicts the modifications to Raymond and insertion of ~~Tetzlaf~~Tetzlaff according to this combination:



Raymond’s FPGA includes a preprogrammed, fixed transceiver block (RX) (bottom dark box) surrounded by programmable FPGA (light dark box). Raymond discloses the components such as receiver (input) and (output) transmitter SERDES, and supporting components, are implemented on the programmable FPGA. Ex. 1007, 8:64–9:8 (describing receiving SERDES modules as “being implemented using programmable logic”), 11:57–67 (likewise for transmitting SERDES modules).

210. The proposed combination modifies Figure 2 of Raymond to include Tetzlaff's DLL/PLL (replacing Raymond's receiver side PLL 240), and for said DLL/PLL to generate the clock (second driving signal 244) for both deserializer (Input SERDES) and serializer (Output SERDES). Using programmable logic, akin to how Raymond programs the SERDES blocks and related wiring in Figure 2, it would have been obvious for a POSITA to program the wiring so that Tetzlaff's DLL/PLL, and not Raymond's PLL 264, is driving the transmitter side clock. Ex. 1007 at 8:64–9:8, 11:57–67. The Output SERDES block would be able to select a clock signal via a multiplexer (MUX), which is shown directly to the right of Output SERDES in the image above. Tetzlaff teaches using a MUX to select an output clock among multiple input clocks. Ex. 1006, Fig. 1B, 6:33–37. In the proposed combination, the MUX allows the Output SERDES block to select between Tetzlaff DLL/PLL and Raymond's transmitter side PLL 264 to provide the transmitter side clock.

a) Motivation to Combine Raymond and Tetzlaff

211. The '587 patent sought to address the synchronization of receiving side and transmitting side clock signals within the FPGA. Ex. 1001, 1:43–46. Raymond and Tetzlaff demonstrate how a POSITA would have been motivated to combine the two references to address that problem. Indeed, both references concern FPGA-based communication systems requiring clock synthesis.

212. The problems and solutions related to Tetzlaff were discussed above in Section IX.A.1.a (Motivation to Combine Giordano and Tetzlaff). In short, Tetzlaff teaches a two-stage architecture in which a delay lock loop (“DLL”) produces a phase-adjusted reference clock that feeds a PLL, and the PLL synthesizes a transmit clock for a serializing data transmitter. Ex. 1006, 9:29–51.

213. Raymond emphasized the importance of minimizing latency in communication devices, particularly for applications such as financial exchange. Ex. 1007 at 1:18–38. Raymond recognized that latency “can be reduced by driving them using a faster clock,” but “[t]his would require another clock signal, which would also require a clock rate conversion” and that “[s]uch a conversion is undesirable for low-latency applications because it introduces another source of latency.” *Id.*, 8:40–50. A POSITA reading Raymond would therefore have recognized that Tetzlaff offers a solution for synthesizing clock signals at the required frequencies while managing phase alignment and filtering jitter—precisely the clock distribution challenges Raymond identifies.

214. A POSITA would have recognized that Tetzlaff’s DLL-to-PLL-to-transmitter chain would be an attractive solution for the PLL functionality depicted in Raymond’s system. Tetzlaff relates to programmable logic devices, providing “methods and apparatuses for clock distribution, phase alignment, and frequency synchronization via high speed serial I/O transceivers.” Ex. 1006 at 4:21–27.

215. Raymond's architecture is implemented on an FPGA, confirming the applicability of Tetzlaff's teachings. A POSITA working with FPGA-based communication devices, as Raymond does, would naturally consult clock distribution architectures designed for programmable logic devices, such as Tetzlaff, for guidance on implementing the PLL-based clock synthesis and phase alignment that Raymond's system requires.

216. Indeed, a POSITA would have been motivated to look to Tetzlaff to combine with Raymond because a person of ordinary skill seeking to implement Raymond's ultra-low latency system would have looked to Tetzlaff for the specific PLL and DLL implementations needed to achieve the clock frequency matching and phase alignment that Raymond requires. Raymond repeatedly discusses implementing its invention with Xilinx UltraScale GTY transceivers. Ex. 1007 at 5:48–56, 6:3–8, 8:18–28, 8:51–55. Tetzlaff, which is a patent assigned to Xilinx, Inc., also expressly incorporates Xilinx transceivers by reference. Ex. 1006, 7:25–36.

217. A POSITA working with Xilinx GTP transceivers, as Raymond suggests, would naturally consult Xilinx's own materials, such as Tetzlaff. In addition, a POSITA would have understood that incorporating Tetzlaff's PLL feedback and phase alignment teachings into Raymond requires no novel hardware

adaptation—only configuring existing transceiver resources and programming the FPGA to implement the teachings of Tetzlaff.

218. A POSITA would have had a reasonable expectation of success in implementing Tetzlaff’s teaching to Raymond. An FPGA is programmable, meaning the computational circuitry and clock management resources can be configured for whatever function the user desires. Ex. 1007 at 8:64–9:8 (describing receiving SERDES modules as “being implemented using programmable logic”), 11:57–67 (likewise for transmitting SERDES modules); Ex. 1006 at 1:23–48 (describing that an FPGA “typically includes an array of programmable tiles” including “configurable logic blocks (CLBs), dedicated random access memory blocks (BRAM), multipliers, digital signal processing blocks (DSPs), processors, clock managers, delay lock loops (DLLs), and so forth”). A POSITA would have recognized that the FPGA’s programmable logic and on-chip clock management resources could be configured to implement Tetzlaff’s DLL-to-PLL-to-serializer clock distribution chain at necessary points in Raymond’s clock path. Because both references suggest implementations with Xilinx transceiver infrastructure, a POSITA would have had high expectations that the combination would function as intended.

2. Ground 2 – Analysis

a) Independent Claim 1

[1.pre]: A method for processing a first serial data stream, using a field programmable gate array system, to generate a second serial data stream, wherein the method comprises the steps of:

219. To the extent the preamble of claim [1] is limiting, Raymond discloses [1.pre]. Raymond discloses “low-latency serializer/de-serializer modules” which discloses this limitation. Ex. 1007, 4:43–51.

220. I provide below a discussion of limitations [1.b] and [1.i] further explaining Raymond’s disclosure of first and second serial data streams.

[1.a]: (a) receiving, by a deserializer in a field programmable array, a clock signal;

221. Raymond discloses this limitation. Raymond’s transceiver 16 and Input SERDES 208 together disclose the claimed “deserializer,” both of which receive incoming a physical layer serial signal 12. Ex. 1007, 8:64–9:8, 10:34–44. Raymond’s CDR 232 recovers a clock signal from this physical layer serial signal 12, and as such this signal comprises the claimed “clock signal.” Ex. 1007, 9:52–57, 10:34–44.

[1.b]: (b) receiving, by the deserializer, the first serial data stream;

222. Raymond discloses this limitation. As I discussed in element [1.a] above, Raymond’s transceiver 16 and Input SERDES 208 disclose the claimed “deserializer.” Raymond discloses that both transceiver 16 and Input SERDES 208

receive an incoming physical layer serial signal 12, which discloses the claimed “first serial data stream.” Ex. 1007, 8:64–9:8, 9:25–28.

[1.c]: (c) generating, by the deserializer, a receiver side clock signal;

223. Raymond discloses this limitation. As I discussed in element [1.a] above, Raymond’s transceiver 16 and Input SERDES 208 disclose the claimed “deserializer.” Raymond’s recovered clock 236 discloses the claimed “receiver side clock signal.” Raymond includes a clock recovery module 232 that recovers a synchronizing clock signal from the incoming electrical physical layer serial signal 12 and outputs a recovered clock signal 236. Ex. 1007, 9:52–57, 10:34–44.

[1.d]: (d) converting, by the deserializer, the first serial data stream into a first plurality of parallel data streams;

224. Raymond discloses this limitation. As I discussed in elements [1.a] and [1.b] above, Raymond’s transceiver 16 and Input SERDES 208 disclose the claimed “deserializer,” and incoming physical layer serial signal 12 discloses “first serial data stream.” Raymond’s parallel signal 216, which becomes the incoming data-link layer parallel signal 248, discloses the “first plurality of parallel data streams.”

225. Raymond’s Input SERDES 208 performs deserialization of the incoming electrical physical layer serial signal 12 to output an incoming electrical physical layer parallel signal 216. Ex. 1007, 8:64–9:8. The incoming electrical physical layer parallel signal 216 discloses the claimed “first plurality of parallel data streams.”

[1.e]: (e) transmitting, from the deserializer to computational circuitry in the field programmable gate array, the first plurality of parallel data streams;

226. Raymond discloses this limitation. As I discussed in elements [1.a] and [1.d] above, Raymond’s transceiver 16 and Input SERDES 208 disclose the claimed “deserializer,” and parallel signal 216 discloses the “first plurality of parallel data streams.”

227. Raymond’s FPGA fabric 56 discloses the claimed “computational circuitry.” The incoming electrical physical layer parallel signal 216 “is received at the FPGA fabric 56 and processed by the receiving-side PCS module 60 and the receiving-side MAC module 64.” Ex. 1007, 11:31–36.² The FPGA fabric 56—including the PCS module 60 and MAC module 64—comprises the claimed “computational circuitry.” The signal may be “further received and processed at a downstream computing unit implementing application-specific algorithms (e.g., electronic trading algorithms)” which “may be implemented within the FPGA fabric 56.” Ex. 1007, 7:19–27.

[1.f]: (f) transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;

228. Raymond in combination with Tetzlaff renders obvious this limitation. As I discussed in elements [1.a] and [1.c] above, Raymond’s transceiver 16 and

² A MAC module is a media access controller, that processes raw data and does error checking.

downstream computing unit disclose the claimed “deserializer,” and the recovered clock signal 236 discloses “receiver side clock signal.”

229. Tetzlaff’s DLL/PLL renders obvious the claimed “phase lock loop.” Raymond discloses that the recovered clock signal 236 (the receiver side clock signal) is fed from the clock recovery module 232 to the receiving-side PLL 240. Ex. 1007, 9:52–66. In the combination of Raymond and Tetzlaff, Tetzlaff’s DLL/PLL replaces PLL 240. A POSITA would have understood that clock signal 236 is transmitted to Tetzlaff’s DLL/PLL in the proposed combination.

[1.g]: (g) generating, using the phase lock loop, a transmitter side clock signal;

230. Raymond in combination with Tetzlaff renders obvious this limitation. As I discussed in element [1.f] above, Tetzlaff’s DLL/PLL implemented in Raymond renders obvious the claimed “phase lock loop.”

231. Tetzlaff’s DLL/PLL results in a clock output, which discloses the claimed “transmitter side clock signal.” In the combination of Raymond and Tetzlaff, Tetzlaff’s DLL/PLL generates the transmitter-side clock signal from the recovered clock signal 236, and this second driving signal 244 drives Raymond’s Output SERDES 256, Tx MAC 88, and Tx PCS 92.

[1.h]: (h) performing, by the computational circuitry, a set of operations on at least a portion of the first plurality of parallel data streams to generate a second plurality of parallel data streams;

232. Raymond discloses this limitation. As I discussed in elements [1.d] and [1.e] above, Raymond's parallel signal 216 discloses the "first plurality of parallel data streams" and FPGA fabric 56 discloses the "computational circuitry."

233. Raymond's outgoing parallel signal 252 discloses the claimed "second plurality of parallel data streams." The FPGA fabric 56 performs multiple operations on the incoming parallel data. For example, Raymond discloses that data on the transmitting side includes "an outgoing data link layer signal 250 that has been outputted after logical processing (e.g. applying financial trading algorithms)" and "[a] transmitting side PCS module 92 converts the data link layer signal to an outgoing electrical physical layer parallel output signal 252." Ex. 1007, 11:57–63. The applying of financial trading algorithms in Raymond discloses the claimed "set of operations."

[1.i]: (i) converting the second plurality of parallel data streams, by a serializer, into a second serial data stream; and

234. Raymond discloses this limitation. As I discussed in element [1.h] above, Raymond's outgoing parallel signal 252 discloses the "second plurality of parallel data streams."

235. Raymond's Output SERDES 256 discloses the claimed "serializer," and the outgoing electrical physical layer serial signal 272 discloses the "second serial

data stream.” The outgoing electrical physical layer parallel signal 252 “is serialized by a transmitting side SerDes module 256 operating in serialization mode.” Ex. 1007, 11:63–65. The transmitting-side SERDES module 256 “may output an outgoing electrical physical layer serial signal 272.” *Id.*, 11:38–39.

[1.i]: (i) transmitting, from the field programmable gate array system, the second serial data stream,

236. Raymond discloses this limitation. As I discussed in element [1.i] above, Raymond’s outgoing electrical physical layer serial signal 272 discloses the “second serial data stream.” The outgoing electrical physical layer serial signal 272 output by the Output SERDES 256 “may be converted at transmitting-side converter 280, prior to being received at the SFP 8.” Ex. 1007, 11:38–41. The transmission from the FPGA system through converter 280 to small form-factor pluggable SFP 8 discloses this limitation.

[1.k]: wherein said method does not use clock domain crossing operations that delay processing of the first set of parallel data streams.

237. Raymond in combination with Tetzlaff renders obvious this limitation. As I discussed in element [1.g] above, Raymond and Tetzlaff render obvious phase matching the receiver side clock with the transmitter side clock. In the combination of Raymond and Tetzlaff, the transmitter-side clock is synthesized by Tetzlaff’s PLL / DLL from the RX recovered clock signal 236 rather than from an independent clock source. This means the Input SERDES 208 and Output SERDES 256 operate in a

single, phase-coherent clock domain derived from the same recovered clock, eliminating the need for clock domain crossing circuitry between the receiver and transmitter paths.

238. Raymond teaches that the “receiving-side driving signals and the transmitting-side driving signals have the same clock frequencies.” Ex. 1007, 2:29–31, cl. 6; 12:17–21 (“the transmitting-side phase lock loop 264 may output driving signals 266, 268 having the same clock frequencies as the receiving-side driving signals 242, 244”).

239. Tetzlaff’s PLL / DLL feedback architecture ensures phase alignment between the source clock (the recovered clock 236) and the target clocks distributed to the transmitter, because the DLL “align[s] a phase of a feedback clock, which is based on one of the target clocks, to the phase of the source clock.” Ex. 1006, cl. 1; 10:15–23. This feedback-based phase alignment ensures that the transmitter operates in phase with the receiver, eliminating the need for elastic buffers or FIFO-based clock domain crossing that would otherwise be required.

240. Because the entire data path—from de-serialization through computational processing to serialization—is driven by clocks derived from the single recovered clock signal 236 through Tetzlaff’s DLL/PLL, the combined system renders obvious this limitation.

b) Claim 2

[2]: The method of claim 1, wherein the first serial data stream comprises market data, the second serial data stream comprises order entry data, and the set of operations are associated with a trading algorithm.

241. Raymond discloses this limitation. Raymond expressly identifies electronic trading as a primary application for its ultra-low latency communication device, and describes the incoming and outgoing data streams in terms that correspond directly to market data and order entry data. For example, Raymond discloses incoming and outgoing serial data streams in context of “financial trading data,” which are transmitted to and from “financial exchange.” Ex. 1007, 4:52–65.

242. Indeed, Raymond discloses “market data feed handler” and “market data feed distribution systems” applications, disclosing receiving market data. Ex. 1007, 13:61–64. Raymond further discloses that the outgoing data link layer signal 250 is “outputted after logical processing (e.g. applying financial trading algorithms),” hence disclosing sending out financial order related data and operations associated with trading algorithms. *Id.* at 11:57–60.

c) Claim 3

[3]: The method of claim 1, wherein the first serial data stream includes market data and the second serial data stream includes trading data.

243. Raymond discloses this limitation. As I discussed in claim [2] above, Raymond discloses receiving market data and transmitting the results of applying financial trading algorithms, hence disclosing this limitation.

d) Claim 4

[4]: The method of claim 1, wherein the 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.

244. Raymond discloses this limitation. As I discussed above in element [1.h] above, Raymond's logical processing (e.g. applying financial trading algorithms) discloses the claimed "set of operations." Raymond's application of financial trading algorithms involves arithmetic operations, hence discloses this limitation. Ex. 1007, 11:57–60. Tetzlaff further discloses this limitation, as I discussed above in Section IX.A.2.b (Ground 1, Cl. 4).

e) Claims 5-7

[5]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed prior to step (g).

[6]: The method of claim 1, wherein at least a portion of the set of operations performed in step (h) are performed after step (g).

[7]: The method of claim 1, wherein all of the set of operations performed in step (h) are performed after step (g).

245. Claims [5] through [7] are directed to the order between step (g) (*i.e.*, generating transmitter side clock) and step (h) (*i.e.*, performing computational operations). The combination of Raymond and Tetzlaff discloses claims [5] through [7]. In the combined system, certain receiving-side operations within the computational circuitry begin processing the incoming parallel data streams before the transmitting-side PLL has generated the transmitter side clock signal.

246. Raymond discloses that clock recovery and de-serialization on the receiving side operate in parallel: “The recovery of the synchronizing clock signal by the clock recovery module 232 may be carried out in parallel with the de-serialization of the incoming electrical physical layer serial signal 12 by the SerDes module 208.” Ex. 1007, 10:7–10. Raymond further discloses that receiving-side operations—PCS decoding, MAC processing, and initial algorithmic processing—are driven by the receiving-side clock and begin as soon as de-serialized data is available. Ex. 1007, 7:19–27.

247. Because the two steps may occur in “parallel,” a POSITA would have found it obvious to perform the computations in step (h) prior to, during, or after the transmitter-side clock generation of step (g).

f) Claim 8

[8.pre]: 8. The method of claim 1, wherein step (g) further comprises:

[8.a]: (k) generating, by the phase lock loop, a feedback clock signal associated with the transmitter side clock signal by performing the following steps until a first output of a phase detector of the field programmable gate array system is below a first threshold level:

[8.b]: (1) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal;

[8.c]: (2) generating, by the phase detector, the first output based on a comparison of the receiver side clock signal and the feedback clock signal obtained from the second clock signal;

[8.d]: (3) transmitting, from the phase detector to a phase controller of the field programmable gate array system, the first output;

[8.e]: (4) determining, by the phase controller, interim adjustment information based on the first output; and

[8.f]: (5) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;

[8.g]: wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information,

[8.h]: wherein steps (1) through (5) are repeated until the first output of the phase detector is below the first threshold level, and

[8.i]: wherein the transmitter side clock signal is associated with the second clock signal.

248. Raymond in view of Tetzlaff renders obvious this claim. As I discussed above for claim [1], Raymond and Tetzlaff renders obvious claim [1] (Section IX.AB.2.a). And as I discussed above in Ground 1 claim [8], Tetzlaff renders obvious claim [8] (Section IX.A.2.d).

g) Claim 9

[9]: The method of claim 1, wherein the first plurality of parallel data streams has the same number of data streams as the second plurality of parallel data streams.

249. Raymond discloses this limitation. Raymond discloses that “the transmitting-side SerDes module 256 can have the same serialization ratio as the de-serialization ratio of the receiving side SerDes module 240.” Ex. 1007, 12:23–25. The same ratio results in the same parallel data streams, hence rendering this limitation obvious.

h) Claim 10

[10]: The method of claim 1, wherein the first plurality of data streams and the second plurality of data streams comprise 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; (viii) eighty (80) data streams; (ix) one hundred twenty-eight (128) data streams; and (x) one hundred sixty (160) data streams.

250. Raymond discloses this limitation. Raymond discloses that “the UltraScale™ GTY from Xilinx™ has internal data buses of 16, 20, 32, 40 or 64 bits, with a minimum bit-width of 16 bits and a minimum de-serialization ratio of 1:16.” Ex. 1007, 6:5–8. A POSITA would have understood that the SerDes module 208 could be configured to apply any of these de-serialization ratios depending on the application requirements, and that the serialization ratio of SerDes module 256 would match.

i) Claim 11

[11]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and phase.

251. Raymond in combination with Tetzlaff renders obvious this limitation. As I discussed in claim [1] above, in the combination of Raymond and Tetzlaff, recovered clock signal 236 renders obvious the receiver side clock signal, and the clock output from Tetzlaff’s DLL/PLL renders obvious the transmitter side clock signal.

252. As to frequency alignment, Raymond discloses that “the receiving-side driving signals and the transmitting-side driving signals have the *same clock frequencies*.” Ex. 1007, 2:29–31, 12:17–25. Raymond further confirms that “transmitting-side phase lock loop 264 may output driving signals 266, 268 having the same clock frequencies as the receiving-side driving signals 242, 244 outputted from the receiving-side phase lock loop 240.” *Id.*, 12:17–22.

253. As to phase alignment, Tetzlaff discloses how DLLs may use a delay line to control the phase difference between two signals, including making the two “synchronized.” Ex. 1006, 2:26–41, 2:49–55. In addition, Tetzlaff discloses that “PLL acts to maintain a selectable phase offset between the ~~synthesizes~~synthesized clock and the reference clock.” Ex. 1006, 10:24–33. This includes “phase shifting the synthesized clock” to “match the phase of the feedback.” *Id.*, 10:55–59.

254. The combination of Raymond and Tetzlaff would therefore be able to generate a transmitter signal that has the same frequency and phase as the recovered clock signal 236, rendering obvious this limitation.

j) Claim 12

[12]: The method of claim 1, wherein the transmitter side clock signal and the receiver side clock signal have the same frequency and different phases.

255. Raymond in combination with Tetzlaff renders obvious this limitation. As discussed in claim [11], Raymond discloses that “the receiving-side driving

signals and the transmitting-side driving signals have the *same clock frequencies*.”

Ex. 1007, 2:29–31, 12:17–25.

256. Raymond expressly discloses an embodiment in which the receiving-side and transmitting-side driving signals are non-synchronized—*i.e.*, have different phases. Raymond teaches “the receiving-side driving signals are non-synchronized with the transmitting-side driving signals.” Ex. 1007, cl. 10; *see also id.*, 12:9–13 (“the transmitting-side phase lock loop 264 can be non-synchronized with the clock signals of the receiving-side phase lock loop 240”). Non-synchronized signals operating at the same frequency necessarily have different phases.

257. Tetzlaff independently teaches a mechanism for maintaining a controlled phase offset between the synthesized clock and the reference clock. Tetzlaff teaches that “the PLL acts to maintain a selectable phase offset between the synthesized clock and the reference clock.” Ex. 1006, 10:24–33. Tetzlaff further teaches that “[t]he selectable phase offset may be used in addition to or instead of the phase adjustment provided by the DLL 308.” *Id.*, 10:31–33. A POSITA could configure Tetzlaff’s PLL to synthesize a transmitter side clock signal that has the same frequency as the receiver side clock signal but a controlled, different phase.

C. Ground 3: Claims 1–12 Are Rendered Obvious by Giordano in View of Tetzlaff further in View of Raymond

258. Giordano in view of Tetzlaff further in view of Raymond renders obvious claims [1]–[12].

1. Prior Art Overview

259. The proposed combination is to use the Giordano and Tetzlaff combination of Ground 1 and further implementing components for financial trading context explicitly disclosed in Raymond.

a) Motivation to Combine Giordano, Tetzlaff, and Raymond

260. Motivations to combine Giordano and Tetzlaff were detailed in Ground 1 at Section IX.A.1.a (Ground 1 Prior Art Overview).

261. A POSITA would have been motivated to look to Raymond to combine with Giordano and Tetzlaff.

262. Raymond emphasized the importance of minimizing latency in communication devices, particularly for applications such as financial exchange. Ex. 1007 at 1:18–38; *see also* Section IX.B.1.a (Motivation to Combine Raymond and Tetzlaff). A POSITA reading Raymond would therefore have recognized that Giordano in view of Tetzlaff offers a solution for synthesizing clock signals at the required frequencies while managing phase alignment and filtering jitter—precisely the clock distribution challenges Raymond identifies.

263. A POSITA would have recognized Giordano in view of Tetzlaff as an attractive solution for the PLL functionality depicted in Raymond’s system, and motivated to look to Giordano and Tetzlaff to combine with Raymond, as I discussed above in Sections IX.A.1.a and IX.B.1.a.

264. A POSITA would have had a reasonable expectation of success in combining Giordano, Tetzlaff, and Raymond. A POSITA would have recognized that the FPGA's programmable logic and on-chip clock management resources could be configured to implement the teachings of the three references as needed. Because all references suggest implementations with Xilinx transceiver infrastructure, a POSITA would have had high expectations that the combination would function as intended, which I discussed above in Sections IX.A.1.a and IX.B.1.a.

2. Ground 3 – Analysis

a) Claims 1, 4–12

265. Giordano in view of Tetzlaff renders obvious claims [1], [4]–[12], as I discussed above in Sections IX.A.2.a–IX.A.2.h (Ground 1, cls. 1, 4–12).

b) Claims 2–3

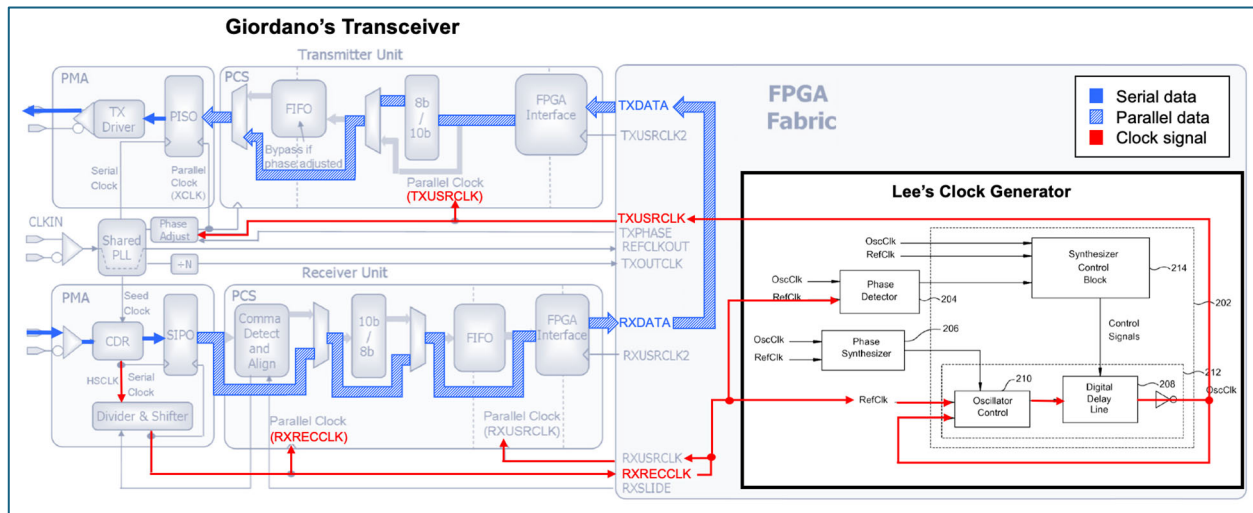
266. Raymond discloses claims [2]–[3], as I discussed above in Sections IX.B.2.b–IX.B.2.c (Ground 2, cls. 2–3).

D. Ground 4: Claims 1 and 4-12 Are Rendered Obvious By Giordano in view of Lee

267. Giordano in view of Lee renders obvious claims [1] and [4] through [12].

1. Prior Art Overview

268. The proposed combination is to program Giordano's FPGA Fabric to implement Lee's Clock Generator, as seen below (based on Giordano Figure 1 and Lee Figure 2):



Giordano's GTP transceiver generates a receiver side clock signal, RXRECCLK, from clock extracted from incoming serial data, and provides RXRECCLK to Lee's Clock Generator implemented in the FPGA Fabric. The Clock Generator creates a transmitter side clock signal TXUSRCLK that is phase-matched to RXRECCLK and sent to the transmitter of the GTP transceiver.

a) Motivation to Combine Giordano and Lee

269. The problems and solutions related to Giordano were discussed above. Section IX.A.1.a (Motivation to Combine Giordano and Tetzlaff). Giordano recognizes that a jitter-filtering PLL is beneficial to condition the recovered receiver clock to serve as a transmitter reference clock. Ex. 1005, 198.

270. Lee discusses solutions for achieving “low output jitter of the frequency aligned mode along with the low phase error of the phase aligned mode.” Ex. 1008, 2:35–55. Lee discloses a clock generator that filters jitter and phase matches the output to the input reference clock. Ex. 1008, 2:35–51, Fig. 2. A POSITA therefore would have recognized Lee’s clock generator as a solution for synthesizing clocks while maintaining phase alignment and filtering jitter. A POSITA reading Giordano would have naturally consulted Xilinx’s own materials (such as Lee, a Xilinx patent), for a phase alignment solution. Giordano repeatedly references the Xilinx RocketIO GTP Transceiver User Guide and Lee is a Xilinx patent, hence both are within the Xilinx FPGA infrastructure. Ex. 1005, 194, 197–199; Ex. 1008, Cover.

271. Incorporating Lee’s Clock Generator into Giordano would only require routine FPGA programming and configuration of existing transceiver resources. Giordano itself demonstrates programming custom hardware in the FPGA Fabric. Section IX.A.1.a (Motivation to Combine Giordano and Tetzlaff). Similarly, Lee also discusses implementing its clock generating teachings “using components that are all digital.” Ex. 1008, 2:35–37, 4:31–35. A POSITA would have understood this to support implementing Lee’s teachings on programmable logic such as on Giordano’s FPGA Fabric. Such FPGA hardware programming is how the ’587 patent suggests implementing the claimed invention. *Supra*, Section IX.A.1.a. Thus,

implementing Lee's Clock Generator circuitry in Giordano's FPGA Fabric mirrors the '587 patent's own implementation approach.

272. A POSITA would have had a reasonable expectation of success. Giordano's FPGA implementation was physically tested on Xilinx evaluation boards and used circuit blocks similar to those disclosed in Lee, including PLLs, DLLs, serializers, and pattern generators. Ex. 1005, 197–200. Giordano acknowledges the elements used for phase matching (such as DLL) are implemented on the FPGA Fabric, circuitry that can be programmed for whatever function the user desires. Ex. 1005, 195, 197. A POSITA may simply reprogram the FPGA Fabric of Giordano to include the clock path modifications based on the teachings of Lee's Clock Generator. Because both references rely on the same Xilinx FPGA infrastructure, there would have been high expectations that, through routine FPGA programming, the combination functions as intended.

2. Ground 4 – Analysis

a) Independent Claim 1

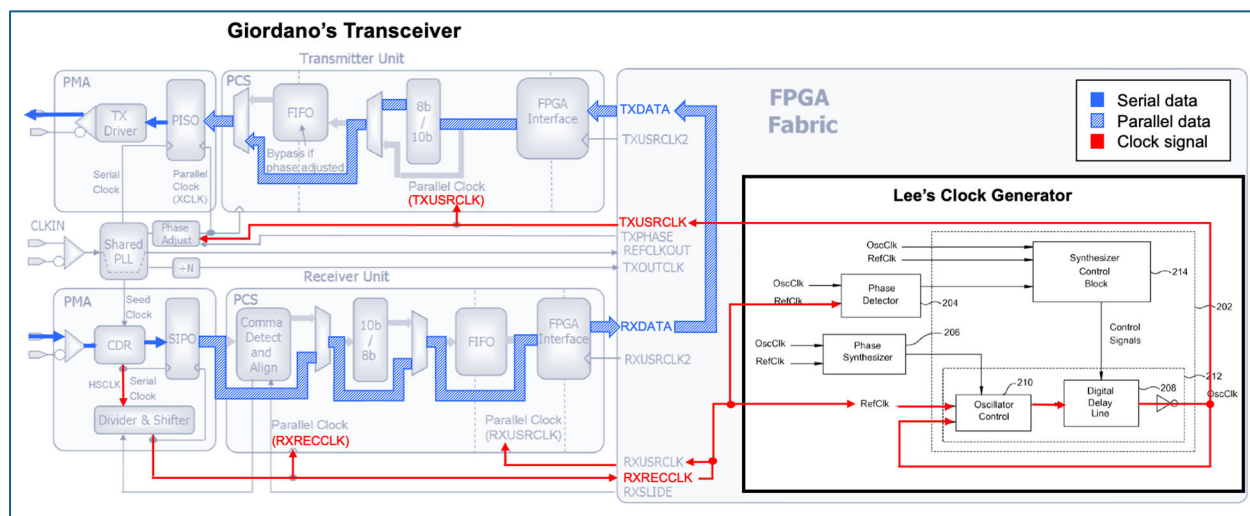
[1.pre] through [1.e]:

273. As I described above in ~~Sections~~Section IX.A.2.a., Giordano discloses limitations [1.pre] through [1.e].

[1.f]: (f) transmitting, from the deserializer to a phase lock loop of the field programmable gate array system, the receiver side clock signal;

274. Giordano in view of Lee renders obvious this limitation. As I discussed in elements [1.a] and [1.c] above, the Receiver Unit PMA in Giordano discloses the claimed “deserializer,” and the recovered clock signal RXRECCLK from the PMA block discloses “receiver side clock signal.”

275. Lee’s Clock Generator programmed in Giordano’s FPGA Fabric renders obvious the claimed “phase lock loop” of the ’587 patent. The combination of Giordano and Lee is depicted in the annotated image below:



Giordano discloses that the recovered RXRECCLK (*i.e.*, “receiver side clock signal”) is provided to the FPGA Fabric. Ex. 1005, 195; Ex. 1008, Fig. 2. A POSITA would have understood this RXRECCLK is transmitted to blocks implemented within the FPGA Fabric, including Lee’s Clock Generator.

276. A POSITA would have known how to implement Lee's Clock Generator in Giordano's FPGA Fabric. *Supra*, **Error! Reference source not found.** (Motivation to Combine Giordano and Lee). While Giordano already phase-aligns the receiver side clock RXRECCLK and transmitter clock TXUSRCLK (as a result of modifying the receiver side clock), Giordano acknowledges that further jitter cleaning would be necessary to use this receiver side clock RXRECCLK for the transmitter. Ex. 1005, 198. Lee's teachings provide the solution for this further jitter cleaning.

277. In the combination of Giordano and Lee, Giordano's FPGA Fabric is programmed to create a clock signal path from the receiver clock RXRECCLK to the transmitter clock TXUSRCLK, where Lee's jitter-filtering and phase-alignment Clock Generator is inserted between the two clocks. Giordano's RXRECCLK (*i.e.*, "receiver side clock signal") is transmitted to Lee's Clock Generator. This generator outputs TXUSRCLK (*i.e.*, "transmitter side clock signal"), which is looped back as input to the Clock Generator for phase matching. This results in synthesizing a clean transmitter side clock phase matched with the receiver side clock. Ex. 1005, 198.

[1.g]: (g) generating, using the phase lock loop, a transmitter side clock signal;

278. Giordano in view of Lee renders obvious this limitation. As I discussed in element [1.f] above, Lee's Clock Generator implemented in Giordano's FPGA

Fabric renders obvious the claimed “phase lock loop.” The output of the Clock Generator results in a “transmitter side clock signal” TXUSRCLK.

[1.h] through [1.k]:

279. Giordano discloses these limitations as I discussed above in Section IX.A.2.a (Ground 1, cl. 1) addressing elements [1.h]–[1.k].

b) Claims 4-7

280. As I discussed above in Sections IX.A.2.b and IX.A.2.c (Ground 1, cls. 4–7), Giordano discloses these claims.

c) Claim 8

[8.pre]: 8. The method of claim 1, wherein step (g) further comprises:

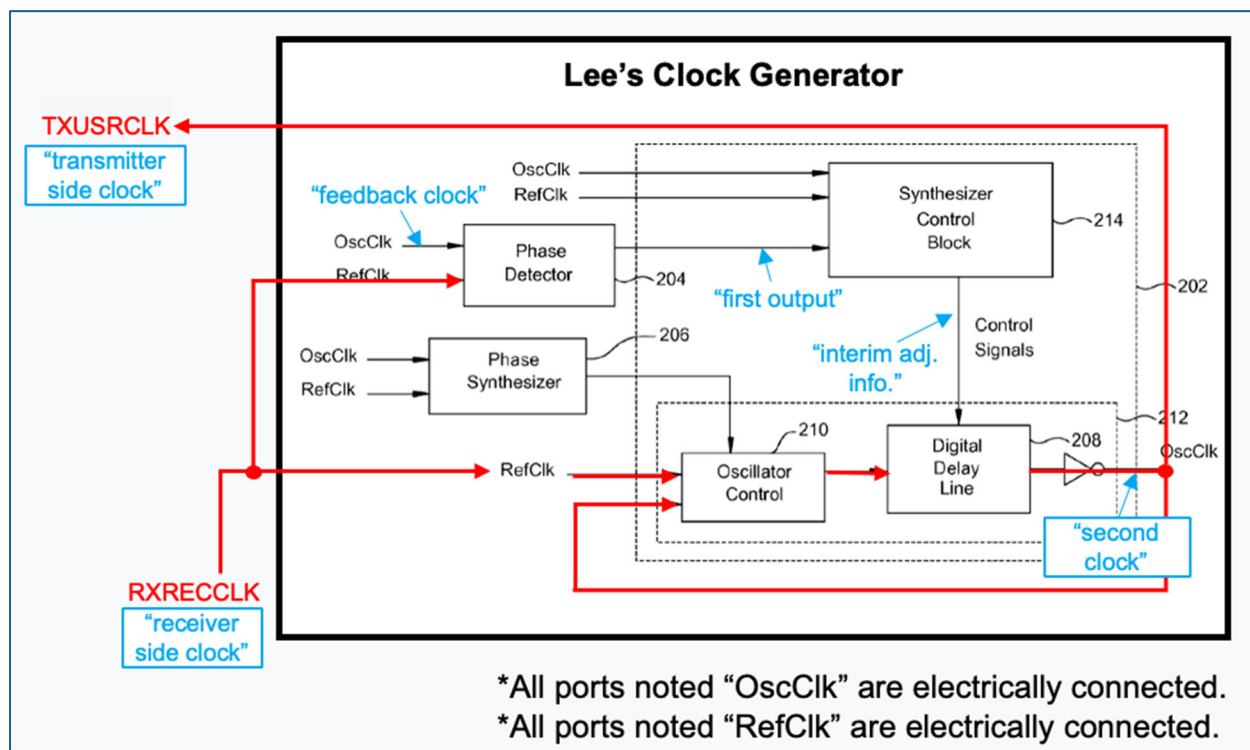
281. To the extent limiting, Giordano renders obvious this limitation, as I discussed in claim [1] and claim limitation [1.g] above.

[8.a]: (k) generating, by the phase lock loop, a feedback clock signal associated with the transmitter side clock signal by performing the following steps until a first output of a phase detector of the field programmable gate array system is below a first threshold level:

282. Lee discloses this limitation. As I discussed in claims [1.f] and [1.g] above, Lee’s Clock Generator renders obvious the claimed “phase lock loop,” which generates the “transmitter side clock signal” TXUSRCLK. Lee’s Phase Detector 204 discloses the claimed “phase detector,” and the OscClk input of Phase Detector 204 discloses the “feedback clock signal.” This feedback clock signal is connected to the “transmitter side clock signal” TXUSRCLK hence is “associated with”

TXUSRCLK (*i.e.*, “transmitter side clock signal”). Lee’s Phase Detector 204 compares its two input signals and provides a signal to the Synthesizer Control Block 214 for adjusting the generated clock. Ex. 1008, 3:16–31. This signal to the Synthesizer Control Block 214 discloses the claimed “first output.”

283. Lee further discloses an iterative process until the two input clocks are aligned (Ex. 1008, 1:30–35, 3:16–24), through the feedback loop depicted in the annotated image below (based on Lee Figure 2):



A POSITA would have understood the repeated steps to achieve this alignment to render obvious this limitation.

[8.b]: (1) generating, by an adjustable oscillator in the field programmable gate array system, a second clock signal;

284. Lee discloses this limitation. Lee's Digital Delay Line 208 renders obvious the claimed "adjustable oscillator." Lee's Digital Delay Line 208 output OscClk discloses the claimed "second clock signal." Lee discloses Digital Delay Line 208 can modify the frequency and phase of the output signal OscClk. Ex. 1008, 2:56–3:24.

[8.c]: (2) generating, by the phase detector, the first output based on a comparison of the receiver side clock signal and the feedback clock signal obtained from the second clock signal;

285. Lee discloses this limitation. As discussed in [8.a] and [8.b], Lee's Phase Detector 204 discloses the claimed "phase detector," its OscClk input discloses the "feedback clock signal," and signal into Synthesizer Control Block 214 discloses the "first output." Lee's Phase Detector 204 compares its two input signals (*i.e.*, "feedback clock signal" and "receiver side clock signal") and provides a signal to the Synthesizer Control Block 214 for adjusting the generated clock. Ex. 1008, 3:16–31. This signal to the Synthesizer Control Block 214 discloses the claimed "first output." The relationships between the aforementioned signals disclose this limitation.

[8.d]: (3) transmitting, from the phase detector to a phase controller of the field programmable gate array system, the first output;

286. Lee discloses this limitation. As discussed in [8.a] and [8.c], Lee's Phase Detector 204 discloses the claimed "phase detector," and signal from the Phase Detector 204 into Synthesizer Control Block 214 discloses the "first output." Lee's Synthesizer Control Block 214 discloses the claimed "phase controller." Lee discloses Phase Detector 204 sending a control signal based on the measured phase difference between two signals to Synthesizer Control Block 214. Ex. 1008, 3:16–31.

[8.e]: (4) determining, by the phase controller, interim adjustment information based on the first output; and

287. Lee discloses this limitation. As discussed in [8.d], Lee's Synthesizer Control Block 214 discloses the "phase controller." The output of Synthesizer Control Block 214 discloses the claimed "interim adjustment information." Lee discloses Synthesizer Control Block 214 providing control signals to Digital Delay Line 208 for modifying the output clock signal, rendering obvious this limitation. Ex. 1008, 3:16–31. Lee discloses multiple variations of Synthesizer Control Blocks, each disclosing how that block may generate said control signals. *E.g.*, Ex. 1008, Figs. 3–5.

[8.f]: (5) transmitting, from the phase controller to the adjustable oscillator, the interim adjustment information;

288. Lee discloses this limitation. As discussed in [8.b], [8.c], and [8.e], Lee's Digital Delay Line 208 renders obvious the "adjustable oscillator," Synthesizer Control Block 214 discloses the "phase controller," and its output discloses "interim adjustment information." Lee discloses Synthesizer Control Block 214 sending control signals to Digital Delay Line 208. Ex. 1008, Fig. 2.

[8.g]: wherein, the adjustable oscillator adjusts the second clock signal based on the interim adjustment information,

289. Lee discloses this limitation. As discussed in [8.b] and [8.e], Lee's Digital Delay Line 208 renders obvious the "adjustable oscillator," its output the "second clock signal," and Synthesizer Control Block 214 output discloses "interim adjustment information." Lee discloses Digital Delay Line 208 can modify the frequency and phase of the output signal OscClk based on the control signal from the Synthesizer Control Block 214. Ex. 1008, 2:56–3:24.

[8.h]: wherein steps (1) through (5) are repeated until the first output of the phase detector is below the first threshold level, and

290. Lee discloses this limitation. As discussed in [8.a] and [8.c], Lee's Phase Detector 204 discloses the claimed "phase detector," and signal into Synthesizer Control Block 214 discloses the "first output." Lee's Phase Detector 204 compares its two input signals and provides a signal to the Synthesizer Control

The diagram illustrates the internal components of Lee's Clock Generator, which is used for clock distribution in a system. It features two main clock paths: a transmitter side clock and a receiver side clock.

Transmitter Side Clock (TXUSRCLK): This path is represented by a red line entering the generator from the top left. It is labeled "transmitter side clock" in a blue box. This clock signal is connected to the RefClk input of the Phase Detector (204) and the RefClk input of the Phase Synthesizer (206). The Phase Detector (204) also receives an OscClk input from the feedback clock (202) and provides a "first output" (204) to the Synthesizer Control Block (214). The Phase Synthesizer (206) receives OscClk and RefClk inputs and provides an "interim adj. info." (206) to the Synthesizer Control Block (214). The Synthesizer Control Block (214) provides Control Signals to the Oscillator Control (210) and the Digital Delay Line (208). The Oscillator Control (210) receives the RefClk signal and provides an output to the Digital Delay Line (208). The Digital Delay Line (208) provides a "second clock" (208) to the receiver side clock path.

Receiver Side Clock (RXRECCLK): This path is represented by a red line entering the generator from the bottom left. It is labeled "receiver side clock" in a blue box. This clock signal is connected to the RefClk input of the Phase Detector (204) and the RefClk input of the Phase Synthesizer (206). The Phase Detector (204) also receives an OscClk input from the feedback clock (202) and provides a "first output" (204) to the Synthesizer Control Block (214). The Phase Synthesizer (206) receives OscClk and RefClk inputs and provides an "interim adj. info." (206) to the Synthesizer Control Block (214). The Synthesizer Control Block (214) provides Control Signals to the Oscillator Control (210) and the Digital Delay Line (208). The Oscillator Control (210) receives the RefClk signal and provides an output to the Digital Delay Line (208). The Digital Delay Line (208) provides a "second clock" (208) to the receiver side clock path.

Feedback Clock (202): This is a common clock signal that is connected to the OscClk input of the Phase Detector (204) and the Phase Synthesizer (206). It is labeled "feedback clock" in a blue box.

Other Components: The diagram includes a Synthesizer Control Block (214), an Oscillator Control (210), and a Digital Delay Line (208). The Synthesizer Control Block (214) is connected to the Phase Detector (204) and the Phase Synthesizer (206). The Oscillator Control (210) is connected to the Phase Synthesizer (206) and the Digital Delay Line (208). The Digital Delay Line (208) is connected to the Phase Synthesizer (206) and the Digital Delay Line (208).

Legend:

- *All ports noted "OscClk" are electrically connected.
- *All ports noted "RefClk" are electrically connected.

[8.i]: wherein the transmitter side clock signal is associated with the second clock signal.

Optiver Ex. 1014
Optiver US LLC v. HFT Solutions, LLC
IPR2026-00383
Page 135 of 165

clock signal” TXUSRCLK hence is “associated with” TXUSRCLK (*i.e.*, “transmitter side clock signal”).

d) Claims 9–12

292. As I discussed above in Sections IX.A.2.e through IX.A.2.h (Ground 1, cls. 9–12), Giordano discloses these claims.

E. Ground 5: Claims 1–12 Are Rendered Obvious by Giordano in View of Lee further in View of Raymond

293. Giordano in view of Lee further in view of Raymond renders obvious claims [1]–[12].

1. Prior Art Overview

294. The proposed combination is to use the Giordano and Lee combination of Ground 4 and further implementing components for financial trading context explicitly disclosed in Raymond.

a) Motivation to Combine Giordano, Lee, and Raymond

295. I discussed the motivation to combine Giordano and Lee in Section IX.D.1 (Ground 4 Prior Art Overview). A POSITA would have been motivated to look to Raymond to combine with Giordano and Lee, for the same reasons to combine Raymond with Giordano and Tetzlaff, as I discussed in above in Section IX.AC.1 (Ground 13 Prior Art Overview).

2. Ground 5 – Analysis

a) Claim 1, 4–12

296. As I discussed above in Sections IX.D.2.b–IX.D.2.d (Ground 4, cls. 1, 4–12), Giordano in view of Lee renders obvious claims [1], [4]–[12].

b) Claims 2–3

297. As I discussed above in Sections IX.B.2.b–IX.B.2.c (Ground 2, cls. 2–3), Raymond discloses claims [2]–[3].

F. Ground 6: Claims 1, 4–12 Are Rendered Obvious by Giordano in View of Tetzlaff further in View of Virtex-UG

298. Giordano in view of Tetzlaff further in view of Virtex-UG renders obvious claims [1], [4]–[12].

1. Prior Art Overview

299. The proposed combination is to use the Giordano and Tetzlaff combination of Ground 1 and further modify transceiver and FPGA configurations as necessary based on the teachings of Virtex-UG.

a) Motivation to Combine Giordano, Tetzlaff, and Virtex-UG

300. While Giordano already inherently discloses and incorporates Virtex-UG, if necessary, a POSITA would have been motivated to look to Virtex-UG to combine with Giordano and Tetzlaff. Giordano repeatedly references the Xilinx RocketIO GTP Transceiver User Guide, and Tetzlaff (a Xilinx patent) expressly

incorporates that same guide by reference. Ex. 1005, 194, 197–199; Ex. 1006, 7:25–36. Virtex-UG details how to use, configure, and modify the Virtex-5 FPGA “RocketIO” GTP transceiver.

301. A POSITA reading Virtex-UG would therefore have recognized that its teachings would be helpful in the use, configuration, and modification of the FPGA system based on Giordano in view of Tetzlaff, including for synthesizing clock signals at the required frequencies while managing phase alignment and filtering jitter. Section IX.A.1.a (Motivation to Combine Giordano and Tetzlaff).

302. A POSITA would have had a reasonable expectation of success in combining Giordano, Tetzlaff, and Virtex-UG. A POSITA would have recognized that the FPGA’s programmable logic and on-chip clock management resources could be configured to implement the teachings of the three references as needed. Because all references suggest implementations with Xilinx transceiver infrastructure, a POSITA would have had a high expectation that the combination would function as intended. Section IX.A.1.a (Motivation to Combine Giordano and Tetzlaff).

2. Ground 6 – Analysis

a) Claims 1, 4–12

303. As I discussed above in Sections IX.A.2.a–IX.A.2.h (Ground 1, cls. 1, 4–12), Giordano in view of Tetzlaff and Virtex-UG renders obvious these claims.

G. Ground 7: Claims 1, 4–12 Are Rendered Obvious by Giordano in View of Lee further in View of Virtex-UG

304. Giordano in view of Lee further in view of Virtex-UG renders obvious claims [1], [4]–[12].

1. Prior Art Overview

305. The proposed combination is to use the Giordano and Lee combination of Ground 4 and further modify transceiver and FPGA configurations as necessary based on the teachings of Virtex-UG.

a) Motivation to Combine Giordano, Lee, and Virtex-UG

306. Motivation to combine Giordano and Lee is detailed in Ground 4. Section IX.D.1 (Ground 4 Prior Art Overview).

307. A POSITA would have been motivated to look to Virtex-UG to combine with Giordano and Lee, for the same reasons to combine Virtex-UG with Giordano and Tetzlaff. Section IX.F.1 (Ground 6 Prior Art Overview).

2. Ground 7 – Analysis

a) Claims 1, 4–12

308. As I discussed above in Sections IX.D.2.a–IX.D.2.d (Ground 4, cls. 1, 4–12), Giordano in view of Lee and Virtex-UG renders obvious these claims.

H. Ground 8: Claims 1–12 Are Rendered Obvious by Giordano in View of Tetzlaff, Virtex-UG, and Raymond

309. Giordano in view of Tetzlaff, Virtex-UG, and Raymond renders obvious claims [1]–[12].

1. Prior Art Overview

310. The proposed combination is to use the Giordano, Tetzlaff, and Virtex-UG combination of Ground 6 and further modify the FPGA for financial trading applications explicitly disclosed in Raymond.

a) Motivation to Combine Giordano, Tetzlaff, Virtex-UG, and Raymond

311. Motivations to combine Giordano, Tetzlaff, and Virtex-UG were detailed in Ground 6. Section IX.F.1 (Ground 6 Prior Art Overview).

312. A POSITA would have been motivated to look to Raymond to combine with Giordano, Tetzlaff, and Virtex-UG, for the same reasons to combine Raymond with Giordano and Tetzlaff. Section IX.C.1 (Ground 3 Prior Art Overview).

2. Ground 8 – Analysis

a) Claims 1, 4–12

313. Giordano in view of Tetzlaff and Virtex-UG renders obvious claims [1], [4]–[12], as I discussed above in Section IX.F.2.a (Ground 6, cls. 1, 4–12).

b) Claims 2–3

314. Raymond discloses claims [2]–[3], as I discussed above in Sections IX.B.2.b–IX.B.2.c (Ground 2, cls. 2–3).

I. Ground 9: Claims 1–12 Are Rendered Obvious by Giordano in View of Lee, Virtex-UG, and Raymond

315. Giordano in view of Lee, Virtex-UG, and Raymond renders obvious claims [1]–[12].

1. Prior Art Overview

316. The proposed combination is to use the Giordano, Lee, and Virtex-UG combination of Ground 7 and further modify the FPGA for financial trading applications explicitly disclosed in Raymond.

a) Motivation to Combine Giordano, Lee, Virtex-UG, and Raymond

317. Motivations to combine Giordano, Lee, and Virtex-UG were detailed in Ground 7. Section IX.G.1 (Ground 7 Prior Art Overview).

318. A POSITA would have been motivated to look to Raymond to combine with Giordano, Lee, and Virtex-UG, for the same reasons to combine Raymond with Giordano and Lee. Section IX.E.1 (Ground 5 Prior Art Overview).

2. Ground 9 – Analysis

b) Claims 1, 4–12

319. Giordano in view of Lee and Virtex-UG renders obvious claims [1], [4]–[12], as I discussed above in Section IX.G.2.a (Ground 7, cls. 1, 4–12).

c) Claims 2–3

320. Raymond discloses claims [2]–[3], as I discussed above in Sections IX.B.2.b–IX.B.2.c (Ground 2, cls. 2–3).

X. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

321. In my opinion, the prior art references that I rely on for the grounds of invalidity are enabled.

322. As I explained above in Section V, I understand an obviousness analysis includes consideration of “secondary” or “objective” evidence of non-obviousness, including things such as: (1) a long-felt but unmet need in the prior art that was satisfied by the claimed invention; (2) commercial success or the lack of commercial success of the claimed invention; (3) unexpected results achieved by the claimed invention; (4) praise of the claimed invention by others skilled in the art; (5) taking of licenses under the patent by others; (6) deliberate copying of the claimed invention; and (7) contemporaneous and independent invention by others.

323. I am not aware of any secondary considerations of non-obviousness raised by Patent Owner with respect to the Challenged Claims of the ’587 patent.

324. As an example, I am not aware of any showing of a long-felt but unmet need satisfied by the '587 patent. As I explained in my state of the art and obviousness analyses above, any alleged “need” that would have been satisfied by the '587 patent had already been satisfied in light of the prior art cited herein. As another example, I am not aware of any commercial success of any data processing system that is due to its practicing of the Challenged Claims of the '587 patent. I likewise am not aware of any evidence relating to the other “secondary” or “objective” considerations listed above.

325. Accordingly, the consideration of “secondary” or “objective” evidence of nonobviousness does not change my opinion that claims [1] through [12] of the '587 patent would have been obvious to one of skill in the art.

XI. CONCLUSION

326. Based on my review and above analysis of the prior art, it is my opinion that claims [1] through [12] of the '587 patent would have been obvious to one of skill in the art.

I, Benjamin Carrion Schaefer, hereby declare that all statements made herein of my own knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with knowledge that willful false statements and the like are so made punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the Petition for *inter partes* review.

Executed on June 29, 2026

By



Benjamin Carrion Schaefer

Appendix A

Hack at DAC competition, third place, 2017
Brain Korean 21 (BK21) scholarship, Seoul, Korea, 2005
Interdisciplinary Research Fund scholarship, University of Birmingham, UK, 1999
M.Sc. Birmingham City University, UK, awarded with distinction, 1999
GFTN Gesellschaft zur Förderung technischen Nachwuchses, Darmstadt, Germany, 1998

SCHOLARLY AND PROFESSIONAL SERVICE ACTIVITIES

Service within the University

<u>Member, CE Graduate Committee</u>	<u>2025 - 2026</u>
<u>Member, Search Committee</u>	<u>2024 - 2025</u>
<u>Member, Search Committee</u>	<u>2023 - 2024</u>
<u>Member, University Committee on Student Technology Requirements</u>	<u>2022 - 2023</u>
<u>Member, Search Committee</u>	<u>2022 - 2023</u>
<u>Member, CE Graduate Committee</u>	<u>2020 - 2021</u>
<u>Member, University Library Committee</u>	<u>2020 - 2021</u>
<u>Member, CE Graduate Committee</u>	<u>2019 - 2020</u>
<u>Member, University Library Committee</u>	<u>2019 - 2020</u>
<u>Member, Undergraduate Teaching Committee</u>	<u>2018 - 2019</u>
<u>Member, Search Committee</u>	<u>2018 - 2019</u>
<u>Member, Search Committee</u>	<u>2017 - 2018</u>
<u>Member, Undergraduate Teaching Committee</u>	<u>2016 - 2017</u>

Journal Editorial Activities

<u>Associate Editor, IEEE Transaction on Computer-Aided Design (TCAD)</u>	<u>2025-present</u>
<u>Associate Editor, IEEE Transaction on Sustainable Computing</u>	<u>2022-present</u>
<u>Associate Editor, Integration, The VLSI Journal (Elsevier)</u>	<u>2014 - 2022</u>

Conference Organization, Executive Committee, Track chair

<u>Track Chair, IEEE Design Automation and Test Europe (DATE)</u>	<u>2027</u>
<u>Track Chair, IEEE Design Automation and Test Europe (DATE)</u>	<u>2026</u>
<u>Track Chair, Innovative Applications of Artificial Intelligence (IAAI)</u>	<u>2026</u>
<u>General Chair, IEEE International Conference on Computer Design (ICCD)</u>	<u>2025</u>
<u>TPC Chair, IEEE International Conference on Computer Design (ICCD)</u>	<u>2024</u>
<u>Track Chair, IEEE Design Automation and Test Europe (DATE)</u>	<u>2025</u>
<u>General Chair, IEEE Dallas Circuits and Systems Conference (DCAS)</u>	<u>2024</u>
<u>Track Chair, ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)</u>	<u>2024</u>
<u>Track Chair (EDA), IEEE International Conference on Computer Design (ICCD)</u>	<u>2023</u>
<u>Technical Program co-Chair, IEEE Dallas Circuits and Systems Conference (DCAS)</u>	<u>2023</u>
<u>Track Chair (EDA), IEEE International Conference on Computer Design (ICCD)</u>	<u>2022</u>
<u>Workshop co-chair, 7th Workshop on Approximate Computing (AxC22)</u>	<u>2022</u>
<u>Special Sessions Chair, IEEE Dallas Circuits and Systems Conference (DCAS)</u>	<u>2022</u>
<u>Track co-Chair, ACM/IEEE International Conference on CAD (ICCAD)</u>	<u>2021</u>
<u>Track Chair, ACM/IEEE Design Automation Conference (DAC)</u>	<u>2021</u>
<u>Track Chair, ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2021</u>
<u>Track Chair, ACM/IEEE Design Automation Conference (DAC)</u>	<u>2020</u>
<u>Track Chair, ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2020</u>
<u>Track Chair, ACM/IEEE Design Automation Conference (DAC)</u>	<u>2019</u>
<u>Track Chair, ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2019</u>
<u>Local Arrangement Chair, IEEE Midwest Symposium on Circuits and Systems (MWSCAS)</u>	<u>2019</u>

<u>General Chair</u> , ESLsyn conference, co-located with DAC	<u>2014</u>
<u>Member of the Technical Program Committee</u>	
<u>IEEE Conference on Field Programmable Technology (FPT)</u>	<u>2026</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2026</u>
<u>IEEE Reconfigurable Architectures Workshop (RAW)</u>	<u>2026</u>
<u>IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)</u>	<u>2026</u>
<u>IEEE Design Automation Conference (DAC)</u>	<u>2026</u>
<u>ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)</u>	<u>2026</u>
<u>ACM/IEEE International Conference on Computer Aided Design (ICCAD)</u>	<u>2025</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2025</u>
<u>IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)</u>	<u>2025</u>
<u>IEEE Design Automation Conference (DAC)</u>	<u>2025</u>
<u>IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)</u>	<u>2024</u>
<u>IEEE Design Automation and Test Europe (DATE)</u>	<u>2024</u>
<u>ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2023</u>
<u>ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)</u>	<u>2023</u>
<u>ACM/IEEE International Conference on Computer Aided Design (ICCAD)(student competition)</u>	<u>2022</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2022</u>
<u>IEEE Application-Specific Systems, Architectures and Processors (ASAP)</u>	<u>2022</u>
<u>ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)</u>	<u>2022</u>
<u>IEEE International Conference on Computer Design (ICCD)</u>	<u>2021</u>
<u>IEEE Transaction on Computers. TPC for special issue on Compiler Optimizations for FPGAs</u>	<u>2021</u>
<u>ACM/IEEE International Conference on Computer Aided Design (ICCAD)</u>	<u>2021</u>
<u>IEEE Application-Specific Systems, Architectures and Processors (ASAP)</u>	<u>2021</u>
<u>ACM/IEEE Design Automation Conference (DAC)</u>	<u>2021</u>
<u>IEEE Design Automation and Test Europe (DATE)</u>	<u>2021</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2021</u>
<u>ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2021</u>
<u>ACM/IEEE International Conference on Computer Aided Design (ICCAD)</u>	<u>2020</u>
<u>IEEE Application-Specific Systems, Architectures and Processors (ASAP)</u>	<u>2020</u>
<u>ACM/IEEE Design Automation Conference (DAC)</u>	<u>2020</u>
<u>IEEE Design Automation and Test Europe (DATE)</u>	<u>2020</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2020</u>
<u>ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2020</u>
<u>ACM/IEEE International Conference on Computer Aided Design (ICCAD)</u>	<u>2019</u>
<u>IEEE Application-Specific Systems, Architectures and Processors (ASAP)</u>	<u>2019</u>
<u>ACM/IEEE Design Automation Conference (DAC)</u>	<u>2019</u>
<u>ACM/IEEE Design Automation and Test Europe (DATE)</u>	<u>2019</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2019</u>
<u>ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2019</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2018</u>
<u>ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2018</u>
<u>ACM/IEEE Design Automation Conference (DAC)</u>	<u>2018</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2018</u>
<u>ACM Great Lakes Symposium on VLSI (GLSVLSI)</u>	<u>2018</u>
<u>ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)</u>	<u>2017</u>
<u>IEEE Conference on Field Programmable Gate Arrays (FPL)</u>	<u>2017</u>
<u>ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)</u>	<u>2016</u>

<u>Electronic System Level Synthesis Conference (ESLSyn)</u>	<u>2015</u>
<u>ACM/IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)</u>	<u>2015</u>
<u>Electronic System Level Synthesis Conference (ESLSyn)</u>	<u>2014</u>
<u>Electronic System Level Synthesis Conference (ESLSyn)</u>	<u>2013</u>
<u>NSF Reviewer</u>	
<u>NSF Reviewer</u>	<u>2026</u>
<u>NSF Reviewer</u>	<u>2025</u>
<u>NSF Reviewer</u>	<u>2024</u>
<u>NSF Reviewer</u>	<u>2022</u>
<u>NSF Reviewer</u>	<u>2021</u>
<u>NSF Reviewer</u>	<u>2021</u>
<u>NSF Reviewer</u>	<u>2019</u>
<u>External PhD Reviewer</u>	
<u>City University, HKSAR</u>	<u>2024</u>
<u>Columbia University, USA</u>	<u>2022</u>
<u>Universidad Politecnica de Madrid, Spain</u>	<u>2021</u>
<u>Universidad Politecnica de Madrid, Spain</u>	<u>2020</u>
<u>Visvesvaraya Technological University (VTU), India</u>	<u>2018</u>

ACHIEVEMENTS IN ORIGINAL INVESTIGATION

Total: 154 Publications (3 Books, 3 Book chapters, 41 Journals, and 106 Conferences)
First author: 40 Publications (3 Books, 1 Book chapter, 19 Journals, and 17 Conferences)

Books

- [B4] **B. Carrion Schaefer**. *High-Level Synthesis Made Easy (Second Edition)*. 2nd. highX Technologies, 2026, pp. 1–183. URL: <http://www.hlsbook.com>.
- [B4] **B. Carrion Schaefer**. *High-Level Synthesis : Applications*. 1st. highX Technologies, 2025, pp. 1–98. URL: <http://www.hlsbook.com>.
- [B4] **B. Carrion Schaefer**. *High-Level Synthesis Made Easy*. 1st. highX Technologies, 2023, pp. 1–135. URL: <http://www.hlsbook.com>.

Book Chapters

- [BC4] Srinivas Katkoori and Sheikh Ariful Islam. *Source Code Obfuscation of Behavioral IPs: Challenges and Solutions, chapter 3*. 2020.
- [BC4] Wim Vanderbauwhede and Khaled Benkrid. *High-Level Synthesis: From Algorithm to Digital Circuit, chapter 7*. 2013.
- [BC4] Philippe Coussy and Adam Morawiec. *High-Level Synthesis: From Algorithm to Digital Circuit*. 2008.

Journals

- [J43] **B. Carrion Schafer** and Baharealsadat Parchamdar. “Approximate Computing in High-Level Synthesis: From Survey to Practical Implementation”. In: *ACM Computing Surveys* (Dec,

2025), pp. 1–20.

- [J43] **B. Carrion Schafer** and Chaitali G. Sathe. “From High-Level Synthesis Lite to High-Level Synthesis Full: Unlocking HLS tool Limitations”. In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* (Nov. 2025), pp. 1–16.
- [J43] Tejinder Kaur, Jose Luis Olvera-Cervantes, **B. Carrion Schafer** and Alonso Corona-Chavez. “Identification of lyophilized avocado powder adulteration using the cavity perturbation technique at microwave frequencies”. In: *Journal of Microwave Power and Electromagnetic Energy* 0.0 (2024), pp. 1–14.
- [J43] Qilin Si and **B. Carrion Schafer**. “MOSAIC: Maximizing Resource Sharing in Behavioral Application Specific Processors”. In: *Elsevier Microprocessors and Microsystems (MICPRO)* 65 (2024), pp. 1–10.
- [J43] Md. Imtiaz Rashid and **B. Carrion Schafer**. “Robust and Efficient RTL to C Compiler Optimized for High-Level Synthesis”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42.11 (Nov. 2024), pp. 1–9.
- [J43] T. Kaur, A. Gamez, J.L. Olvera, **B. Carrion Schaefer**, and A. Corona-Chavez. “I-TAINTED: Identification of Turmeric Adulteration using the Cavity Perturbation Technique and Technology optimized Machine Learning”. In: *IEEE Access* (2023), pp. 1–11.
- [J43] Md. Imtiaz Rashid and **B. Carrion Schafer**. “Fast and Inexpensive High-Level Synthesis Design Space Exploration : Machine Learning to the Rescue”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42.11 (Nov. 2023), pp. 3939–3950.
- [J43] Q. Si, P. Chowdhury, R. Sreekumar, and **B. Carrion Schafer**. “Application Specific Approximate Behavioral Processor”. In: *IEEE Transactions on Sustainable Computing* (2022), pp. 1–15.
- [J43] P. Goswami, **B. Carrion Schafer** and D. Bhatia. “Machine Learning Based Fast and Accurate High Level Synthesis Design Space Exploration: From Graph to Synthesis”. In: *Integration* 88 (2022), pp. 116–124.
- [J43] Z. Wang and **B. Carrion Schafer**. “Learning from the Past: Efficient High-Level Synthesis Design Space Exploration for FPGAs”. In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 27.4 (July 2022), pp. 1–23.
- [J43] Z. Wang, F. Lau, and **B. Carrion Schafer**. “SSSL: Secure Search Space Locking of Behavioral IPs”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2021), pp. 1–10.
- [J43] Q. Si, S. Shetty, and **B. Carrion Schafer**. “Building Complete Heterogeneous Systems-on-Chip in C: From Hardware Accelerators to CPUs”. In: *MDPI Electronics* (2021), pp. 1–15.
- [J43] P. Chowdhury and **B. Carrion Schafer**. “Leveraging Automatic High-Level Synthesis Resource Sharing to maximize Dynamical Voltage Overscaling with Error Control”. In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* (Sept. 2021), pp. 1–13.
- [J43] A. Yadav, S. Xu, **B. Carrion Schafer**, and A. Davoudi. “Hardware-assisted Simulation of Voltage-behind-reactance Models of Electric Machines on FPGA”. In: *IEEE Transactions*

on Energy Conversion (2020), pp. 1–10.

- [J43] S. Liu, F. Lau, and **B. Carrion Schafer**. “Predictive Compositional Method to Design and Re-optimize Complex Behavioral Dataflows”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2020), pp. 1–12.
- [J43] **B. Carrion Schafer** and Z. Wang. “High-Level Synthesis Design Space Exploration: Past, Present and Future”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2020), pp. 1–12.
- [J43] Siyuan Xu, Shuangnan Liu, Yidi Liu, Anushree Mahapatra, Mónica Villaverde, Felix Moreno, and **B. Carrion Schafer**. “Design space exploration of heterogeneous MPSoCs with variable number of hardware accelerators”. In: *Microprocessors and Microsystems* 65 (2019), pp. 169–179.
- [J43] S. Xu and **B. Carrion Schafer**. “Toward Self-Tunable Approximate Computing”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.4 (Apr. 2019), pp. 778–789.
- [J43] David Aledo, **B. Carrion Schafer**, and Felix Moreno. “VHDL vs. SystemC: Design of Highly Parameterizable Artificial Neural Networks”. In: *IEICE Transactions on Information and Systems* E102.D.3 (2019), pp. 512–521.
- [J43] A. Mahapatra and **B. Carrion Schafer**. “VeriIntel2C: Abstracting RTL to C to maximize High-Level Synthesis Design Space Exploration”. In: *Integration* 64 (2019), pp. 1–12.
- [J43] John N. Randall, James H. G. Owen, Joseph Lake, Rahul Saini, Ehud Fuchs, Mohammad Mahdavi, S. O. Reza Moheimani, and **B. Carrion Schafer**. “Highly parallel scanning tunneling microscope based hydrogen depassivation lithography”. In: *Journal of Vacuum Science & Technology B* 36.6 (2018), 06JL05.
- [J43] Anushree Mahapatra, Yidi Liu, and **B. Carrion Schafer**. “Accelerating cycle-accurate system-level simulations through behavioral templates”. In: *Integration* 62 (2018), pp. 282–291.
- [J43] S. Xu and **B. Carrion Schafer**. “Exposing Approximate Computing Optimizations at Different Levels: From Behavioral to Gate-Level”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.11 (Nov. 2017), pp. 3077–3088.
- [J43] N. Veeranna and **B. Carrion Schafer**. “S3CBench: Synthesizable Security SystemC Benchmarks for High-Level Synthesis”. In: *J. Hardware and Systems Security* 1.2 (2017), pp. 103–113.
- [J43] N. Veeranna and **B. Carrion Schafer**. “Trust Filter: Runtime Hardware Trojan Detection in Behavioral MPSoCs”. In: *J. Hardware and Systems Security* 1.1 (2017), pp. 56–67.
- [J43] **B. Carrion Schafer**. “Parallel High-Level Synthesis Design Space Exploration for Behavioral IPs of Exact Latencies”. In: *ACM Trans. Des. Autom. Electron. Syst.* 22.4 (May 2017).
- [J43] **B. Carrion Schafer**. “Enabling High-Level Synthesis Resource Sharing Design Space Exploration in FPGAs Through Automatic Internal Bitwidth Adjustments”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36.1 (Jan. 2017), pp. 97–105.
- [J43] N. Veeranna and **B. Carrion Schafer**. “Hardware Trojan Detection in Behavioral Intellectual Properties (IP’s) Using Property Checking Techniques”. In: *IEEE Transactions on Emerging Topics in Computing* 5.4 (Oct. 2017), pp. 576–585.

- [J43] **B. Carrion Schafer**. “Probabilistic Multi-knob High-Level Synthesis Design Space Exploration Acceleration”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.3 (Mar. 2016), pp. 394–406.
- [J43] **B. Carrion Schafer**. “Tunable Multiprocess Mapping on Coarse-Grain Reconfigurable Architectures With Dynamic Frequency Control”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24.1 (Jan. 2016), pp. 324–328.
- [J43] **B. Carrion Schafer**. “Source Code Error Detection in High-Level Synthesis Functional Verification”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24.1 (Jan. 2016), pp. 301–312.
- [J43] **B. Carrion Schafer**. “Hierarchical High-Level Synthesis Design Space Exploration with Incremental Exploration Support”. In: *IEEE Embedded Systems Letters* 7.2 (June 2015), pp. 51–54.
- [J43] **B. Carrion Schafer** and A. Mahapatra. “S2CBench: Synthesizable SystemC Benchmark Suite for High-Level Synthesis”. In: *IEEE Embedded Systems Letters* 6.3 (Sept. 2014), pp. 53–56.
- [J43] **B. Carrion Schafer** and K. Wakabayashi. “Machine learning predictive modelling high-level synthesis design space exploration”. In: *IET Computers Digital Techniques* 6.3 (May 2012), pp. 153–159.
- [J43] **B. Carrion Schafer** and K. Wakabayashi. “Divide and Conquer High-level Synthesis Design Space Exploration”. In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 17.3 (July 2012), 29:1–29:19.
- [J43] **B. Carrion Schafer** and K. Wakabayashi. “Precision tunable RTL macro-modelling cycle-accurate power estimation”. In: *IET Computers Digital Techniques* 5.2 (Mar. 2011), pp. 95–103.
- [J43] **Benjamin Carrion Schafer** and Majid Sarrafzadeh. “Semi-Automatic Control Unit Generation for Complex VLSI Designs”. In: *IPSJ Transactions on System LSI Design Methodology* 5.4 (2010), pp. 234–243.
- [J43] **Benjamin Carrion Schafer**, Yusuke Iguchi, Wataru Takahashi, Shingo Nagatani, and Kazutoshi Wakabayashi. “Fixed Point Data Type Modeling for High Level Synthesis”. In: *IEICE Transactions on Electronics* E93.C.3 (2010), pp. 361–368.
- [J43] **B. Carrion Schafer** and K. Wakabayashi. “Design Space Exploration Acceleration Through Operation Clustering”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29.1 (Jan. 2010), pp. 153–157.
- [J43] **B. Carrion Schafer** and T. Kim. “Autonomous temperature control technique in VLSI circuits through logic replication”. In: *IET Computers Digital Techniques* 3.1 (Jan. 2009), pp. 62–71.
- [J43] **B. Carrion Schafer** and T. Kim. “Hotspots Elimination and Temperature Flattening in VLSI Circuits”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 16.11 (Nov. 2008), pp. 1475–1487.
- [J43] **B. Carrion Schafer**, Steven F. Quigley, and Andrew H.C. Chan. “Acceleration of the Discrete Element Method (DEM) on a reconfigurable co-processor”. In: *Computers and Structures* 82.20 (2004), pp. 1707–1718.

Conference Papers

- [C107] Baharealsadat Parchamdar and **B. Carrion Schafer**. “Breaking Behavioral IPs Design Space Lock”. In: *Great Lakes Symposium on VLSI (Best Poster Award)*. GLSVLSI’25. Association for Computing Machinery, 2025, pp. 1–6.
- [C107] Sabrina Ahmed, Khaza Anuarul Hoque and **B. Carrion Schafer**. “Robust and Fault-Aware Design for Resilient Edge Computing AI Accelerators”. In: *Great Lakes Symposium on VLSI*. GLSVLSI’25. Association for Computing Machinery, 2025, pp. 1–6.
- [C107] Victoria Gammenthaler and **B. Carrion Schafer**. “Improving the Quality of the High-Level Synthesis Estimation Results through Multi-Level Predictive Models”. In: *Great Lakes Symposium on VLSI*. GLSVLSI’25. Association for Computing Machinery, 2025, pp. 1–6.
- [C107] C.G. Sathe, Y. Makris, **B. Carrion Schafer**. “FREEDOM: FPGA-based Hardware Redaction Emulator”. In: *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM 2025)*. 2025, pp. 1–5.
- [C107] C.G. Sathe, Y. Makris, **B. Carrion Schafer**. “FPGA-based Hardware Redaction Bitstream Fuzzing Attack”. In: *IEEE Dallas Circuits and Systems (DCAS)*. 2025, pp. 1–5.
- [C107] C.G. Sathe, A. Fowler, C. Sechen, **B. Carrion Schafer**, Y. Makris, M. Zaman and R. Narumi. “What to Redact? Guidelines for Cost-Effective Hardware Intellectual Property Protection (Best Poster Award)”. In: *GOMACTech*. 2025, pp. 1–4.
- [C107] B. Parchamdar and **B. Carrion Schafer**. “ADVISOR: Approximate Computing-friendly High-Level Synthesis Design Space Explorer”. In: *ACM/IEEE Design Automation Conference (DAC)*. June 2025, pp. 1–6.
- [C107] M. Imtiaz Rashid and **B. Carrion Schafer**. “Making Legacy Hardware Robust against Side Channel Attacks via High-Level Synthesis”. In: *Asia and South Pacific Design Automation (ASP-DAC)*. Jan. 2025, pp. 1–6.
- [C107] C.G. Sathe, Y. Makris, and **B. Carrion Schafer**. “Efficient and Secure Cloud-based Split Logic Synthesis”. In: *Asia and South Pacific Design Automation (ASP-DAC)*. Jan. 2025, pp. 1–6.
- [C107] Q. Si and **B. Carrion Schafer**. “HAMMER: Hardware-aware Runtime Program Execution Acceleration through runtime reconfigurable CGRAs”. In: *Asia and South Pacific Design Automation (ASP-DAC)*. Jan. 2025, pp. 1–6.
- [C107] M. Imtiaz Rashid and **B. Carrion Schafer**. “VeriPy: A Python-Powered Framework for Parsing Verilog HDL and High-Level Behavioral Analysis of Hardware”. In: *IEEE Dallas Circuits and Systems (DCAS)*. Apr. 2024, pp. 1–6.
- [C107] B. Parchamdar and **B. Carrion Schafer**. “Investigating the Effect of Hyper-Parameter Settings on Simulated Annealing-based High-Level Synthesis Design Space Exploration”. In: *IEEE Dallas Circuits and Systems (DCAS)*. Apr. 2024, pp. 1–5.
- [C107] B. Parchamdar and **B. Carrion Schafer**. “Finding Bugs in RTL Descriptions: High-Level Synthesis to the Rescue”. In: *ACM/IEEE Design Automation Conference (DAC)*. June 2024, pp. 1–6.
- [C107] S. Shetty and **B. Carrion Schafer**. “Facilitating the Design of complete System-on-Chip

- through High-Level Synthesis”. In: *GOMACTech*. 2024, pp. 1–4.
- [C107] S. Yu and **B. Carrion Schafer**. “Temperature Control through dynamic Approximation”. In: *GOMACTech*. 2024, pp. 1–4.
- [C107] **B. Carrion Schafer** and C.G. Sathe. “Circumventing Restrictions in commercial High-Level Synthesis Tools”. In: *Design, Automation, and Test in Europe (DATE)*. Mar. 2024, pp. 1–2.
- [C107] Qilin Si and **B. Carrion Schafer**. “PEPA: Performance Enhancement of Embedded Processors through HW Accelerator Resource Sharing”. In: *Great Lakes Symposium on VLSI GLSVLSI’23*. Association for Computing Machinery, 2023, pp. 1–6.
- [C107] Qilin Si and **B. Carrion Schafer**. “ADVICE: Automatic Design and Optimization of Behavioral Application Specific Processors”. In: *Great Lakes Symposium on VLSI GLSVLSI’23*. Association for Computing Machinery, 2023, pp. 1–6.
- [C107] Md Imtiaz Rashid, A.H. Torabi and **B. Carrion Schafer**. “CERTIFY: Automatic Measuring the Quality of High-Level Synthesis”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2023, pp. 1–5.
- [C107] Md. Imtiaz and **B. Carrion Schafer**. “MIRROR: Maximizing the Re-usability of RTL through RTL to C Compiler”. In: *Design, Automation, and Test in Europe (DATE)*. Mar. 2023, pp. 1–6.
- [C107] C.G. Sathe, Y. Makris, and **B. Carrion Schafer**. “MANTIS: Machine Learning-Based Approximate Modeling of Redacted Integrated Circuits”. In: *Design, Automation, and Test in Europe (DATE)*. Mar. 2023, pp. 1–6.
- [C107] A. Hossein Md I. Rashid and **B. Carrion Schafer**. “Automatic Modernization of Hardware Assets”. In: *GOMACTech*. 2023, pp. 1–5.
- [C107] C. Sathe Md I. Rashid and **B. Carrion Schafer**. “ROPE: Re-usability Lock of Behavioral Intellectual Property”. In: *GOMACTech*. 2023, pp. 1–5.
- [C107] Prattay Chowdhury, Jorge Castro Godinez, and **B. Carrion Schafer**. “Approximating HW Accelerators through Partial Extractions onto shared Artificial Neural Networks”. In: *Asia and South Pacific Design Automation (ASP-DAC)*. Jan. 2023, pp. 1–6.
- [C107] P. Chowdhury, C.G. Sathe, **B. Carrion Schafer**. “Predictive Model Attack for Embedded FPGA Logic Locking”. In: *International Symposium on Low Power Electronics and Design (ISLPED)*. ISLPED ’22. ACM/IEEE, 2022, pp. 1–6.
- [C107] C.G. Sathe, Y. Makris and **B. Carrion Schafer**. “Investigating the Effect of different eFPGAs fabrics on Logic Locking through HW Redaction”. In: *Dallas Circuits and Systems Conference (DCAS)*. DCAS’22. IEEE, 2022, pp. 1–6.
- [C107] Q. Si and **B. Carrion Schafer**. “Optimizing Behavioral Near On-Chip Memory Computing Systems”. In: *International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. ASAP’22. IEEE, 2022, pp. 1–7.
- [C107] Md Imtiaz Rashid and **B. Carrion Schafer**. “Fast Parallel High-Level Synthesis Design Space Explorer: Targeting FPGAs to accelerate ASIC Exploration”. In: *Great Lakes Symposium on VLSI (Best Paper Award)*. GLSVLSI ’22. Association for Computing Machinery, 2022, pp. 1–6.

- [C107] Md Imtiaz Rashid, Qilin Si and **B. Carrion Schafer**. “Modernizing Hardware Circuits through High-Level Synthesis”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2022, pp. 1–5.
- [C107] **B. Carrion Schafer**. “Hotspot Mitigation through Multi-Row Thermal-aware Re-Placement of Logic Cells based on High-Level Synthesis Scheduling”. In: *Asia and South Pacific Design Automation (ASP-DAC)*. Jan. 2022, pp. 1–6.
- [C107] Md Imtiaz Rashid and **B. Carrion Schafer**. “Improving the Quality of Hardware Accelerators through automatic Behavioral Input Language Conversion in HLS”. In: *Asia and South Pacific Design Automation (ASP-DAC)*. Jan. 2022, pp. 1–6.
- [C107] Z. Wang, S. O. Mohammed, Y. Makris and **B. Carrion Schafer**. “Functional Locking through Omission: From HLS to Obfuscated Design”. In: *IEEE 39th International Conference on Computer Design (ICCD)*. Nov. 2021, pp. 1–8.
- [C107] P. Chowdhury and **B. Carrion Schafer**. “ADAPT: ANN-Controlled System-Level Runtime Adaptable APproximate CompuTing”. In: *IEEE 39th International Conference on Computer Design (ICCD)*. Nov. 2021, pp. 1–4. (invited paper).
- [C107] Prattay Chowdhury and **B. Carrion Schafer**. “BEACON : BEst Approximations for Complete BehaviOral HeterogeNeous SoCs”. In: *International Symposium on Low Power Electronics and Design (ISLPED)*. ISLPED ’21. ACM/IEEE, 2021, pp. 1–6.
- [C107] Qilin Si, Imtiaz Rashid, and **B. Carrion Schafer**. “Micro-architecture Tuning for Dynamic Frequency Scaling in Coarse-Grain Runtime Reconfigurable Arrays with Adaptive Clock Domain Support”. In: *IEEE Computer Society Annual Symposium on VLSI. ISVLSI ’21*. IEEE, 2021, pp. 1–6.
- [C107] Prattay Chowdhury and **B. Carrion Schafer**. “Unlocking Approximations through Selective Source Code Transformations”. In: *Great Lakes Symposium on VLSI. GLSVLSI ’21*. Association for Computing Machinery, 2021, pp. 1–6.
- [C107] S. Shetty and **B. Carrion Schafer**. “Enabling the Design of Behavioral Systems-on-Chip”. In: *ACM/IEEE Design Automation Conference (DAC)*. July 2021, pp. 1–6.
- [C107] Z. Zhu and **B. Carrion Schafer**. “Reducing the Complexity of Fault-Tolerant System amenable to Approximate Computing”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2021, pp. 1–5.
- [C107] Y. Gao and **B. Carrion Schafer**. “Effective High-Level Synthesis Design Space Exploration through a Novel Cost Function Formulation”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2021, pp. 1–5.
- [C107] P. Kalimuthu, K. Basu, and **B. Carrion Schafer**. “Efficient Hierarchical Post-Silicon Validation and Debug”. In: *International Conference on VLSI Design (VLSID) (Honorable Men- tion Award)*. Feb. 2021, pp. 1–6.
- [C107] J. Chen and **B. Carrion Schafer**. “Watermarking of Behavioral IPs: A Practical Approach”. In: *Design, Automation, and Test in Europe (DATE)*. Feb. 2021, pp. 1–4.
- [C107] Z. Wang and **B. Carrion Schafer**. “Locking the Re-usability of Behavioral IPs: Discriminating the Search Space through Partial Encryptions”. In: *Design, Automation, and Test in Europe (DATE)*. Feb. 2021, pp. 1–4.

- [C107] J. Chen and **B. Carrion Schafer**. “Area Efficient Functional Locking through Coarse Grained Runtime Reconfigurable Architectures”. In: *Asia and South Pacific Design Automation (ASP- DAC)*. Jan. 2021, pp. 1–6.
- [C107] M. Shah and **B. Carrion Schafer**. “Flexible Runtime Reconfigurable Computing Overlay Architecture and Optimization for Dataflow Applications”. In: *Hierarchical Parallelism for Exascale Computing (HiPar)*. Nov. 2020, pp. 1–6.
- [C107] R. Sreekumar, P. Chowdhury, and **B. Carrion Schafer**. “Bespoke Approximate Behavioral Processors”. In: *IEEE 38th International Conference on Computer Design (ICCD)*. Nov. 2020, pp. 1–4.
- [C107] A. Balachandran and **B. Carrion Schafer**. “Efficient Functional Locking of Behavioral IPs”. In: *IEEE International Midwest Symposium on Circuits and Systems (MWCAS)*. Aug. 2020, pp. 1–4.
- [C107] Z. Wang and **B. Carrion Schafer**. “Machine Learning to Set Meta-Heuristic Specific Parameters for High-Level Synthesis Design Space Exploration”. In: *ACM/IEEE Design Automation Conference (DAC)*. July 2020, pp. 1–6.
- [C107] J. Chen, M. Zaman, Y. Makris, S. Blanton, S. Mitra, and **B. Carrion Schafer**. “DECOY: DEflection-Driven HLS-Based Computation Partitioning for Obfuscating Intellectual Property”. In: *ACM/IEEE Design Automation Conference (DAC)*. July 2020, pp. 1–6.
- [C107] Z. Zhu and **B. Carrion Schafer**. “Light-Weight Soft-Errors Detection Mechanism in High-Level Synthesis”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2020, pp. 1–5.
- [C107] M. M. Shihab, J. Tian, G. R. Reddy, B. Hu, W. Swartz, **B. Carrion Schafer**, C. Sechen, and Y. Makris. “A Transistor-Level Fabric for Design Obfuscation,” in: *GOMACTech*. Mar. 2020, pp. 1–4.
- [C107] Z. Wang, J. Chen, and **B. Carrion Schafer**. “Efficient and Robust High-Level Synthesis Design Space Exploration through offline Micro-kernels Pre-characterization,” in: *Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2020, pp. 1–6.
- [C107] Bo Hu, Mustafa Shihab, William Swartz, Yiorgos Makris, **B. Carrion Schafer**, and Carl Sechen. “OAn Efficient MILP-Based Aging-Aware Floorplanner for Multi-Context Coarse-Grained Runtime Reconfigurable FPGAs”. In: *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2020, pp. 1–6.
- [C107] S. Xu and **B. Carrion Schafer**. “On the Design of High Performance HW Accelerator through High-level Synthesis Scheduling Approximations”. In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2020, pp. 1–6.
- [C107] Bo Hu, Mustafa Shihab, William Swartz, Yiorgos **B. Carrion Schafer** Makris, and Carl Sechen. “Extending the Lifetime of Coarse-grained Runtime Reconfigurable FPGAs by Balancing Processing Element Usage,” in: *International Conference on Field-Programmable Technology (FPT)*. Dec. 2019.
- [C107] F.N. Taher, A. Balajandran, and **B. Carrion Schafer**. “Learning-based Diversity Estimation: Leveraging the Power of High-level Synthesis”. In: *IEEE 37th International*

- Conference on Computer Design (ICCD). Nov. 2019.
- [C107] J. Chen and **B. Carrion Schafer**. “Low Power Design through Frequency-Optimized Runtime Micro-architectural Adaptation”. In: *IEEE 37th International Conference on Computer Design (ICCD)*. Nov. 2019.
- [C107] J. Chen and **B. Carrion Schafer**. “Exploiting the Benefits of High-level Synthesis for Thermal-aware VLSI Design”. In: *IEEE 37th International Conference on Computer Design (ICCD)*. Nov. 2019.
- [C107] S. Xu and **B. Carrion Schafer**. “Low Power Design of Runtime Reconfigurable FPGAs through Contexts Approximations”. In: *IEEE 37th International Conference on Computer Design (ICCD)*. Nov. 2019.
- [C107] S. Xu and **B. Carrion Schafer**. “Approximating Behavioral HW Accelerators through Selective Partial Extractions onto Synthesizable Predictive Models”. In: *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. Nov. 2019, pp. 1–8.
- [C107] M. R. Babu, F. N. Taher, A. Balachandran, and **B. Carrion Schafer**. “Efficient Hardware Acceleration for Design Diversity Calculation to Mitigate Common Mode Failures”. In: *IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. Apr. 2019, pp. 267–270.
- [C107] Jianqi Chen and **B. Carrion Schafer**. “Thermal Fingerprinting of FPGA Designs through High-Level Synthesis”. In: *Great Lakes Symposium on VLSI. GLSVLSI '19*. Tysons Corner, VA, USA: Association for Computing Machinery, 2019, pp. 331–334.
- [C107] Zhiqi Zhu, Farah Naz Taher, and **B. Carrion Schafer**. “Exploring Design Trade-Offs in Fault-Tolerant Behavioral Hardware Accelerators”. In: *Great Lakes Symposium on VLSI. GLSVLSI '19*. Tysons Corner, VA, USA: Association for Computing Machinery, 2019, pp. 291–294.
- [C107] Bo Hu, Jingxiang Tian, Mustafa Shihab, Gaurav Rajavendra Reddy, William Swartz, Yiorgos Makris, **B. Carrion Schafer**, and Carl Sechen. “Functional Obfuscation of Hardware Accelerators through Selective Partial Design Extraction onto an Embedded FPGA”. In: *Great Lakes Symposium on VLSI. GLSVLSI '19*. Tysons Corner, VA, USA: Association for Computing Machinery, 2019, pp. 171–176.
- [C107] S. Liu, F. C. Lau, and **B. Carrion Schafer**. “Accelerating FPGA Prototyping through Predictive Model-Based HLS Design Space Exploration”. In: *56th ACM/IEEE Design Automation Conference (DAC)*. June 2019, pp. 1–6.
- [C107] A. Mahapatra and **B. Carrion Schafer**. “Optimizing RTL to C Abstraction Methodologies to Improve HLS Design Space Exploration”. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2019, pp. 1–5.
- [C107] Z. Wang and **B. Carrion Schafer**. “Partial Encryption of Behavioral IPs to Selectively Control the Design Space in High-Level Synthesis”. In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2019, pp. 642–645.
- [C107] M. M. Shihab, J. Tian, G. R. Reddy, B. Hu, W. Swartz, **B. Carrion Schafer**, C. Sechen, and Y. Makris. “Design Obfuscation through Selective Post-Fabrication Transistor-Level Programming”. In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2019, pp. 528–533.

- [C107] F. N. Taher, M. Joslin, A. Balachandran, Z. Zhu, and **B. Carrion Schafer**. “Common-Mode Failure Mitigation: Increasing Diversity through High-Level Synthesis”. In: *Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2019, pp. 1563–1566.
- [C107] Z. Zhu, J. Callenes-Sloan, and **B. Carrion Schafer**. “Control Flow Checking Optimization Based on Regular Patterns Analysis”. In: *IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*. Dec. 2018, pp. 203–212.
- [C107] S. Xu and **B. Carrion Schafer**. “DEEP: Dedicated Energy-Efficient Approximation for Dynamically Reconfigurable Architectures”. In: *IEEE 36th International Conference on Computer Design (ICCD)*. Oct. 2018, pp. 587–594.
- [C107] S. Xu and **B. Carrion Schafer**. “Autonomous Temperature Management through Selective Control of Exact-Approximate Tiles”. In: *IEEE 36th International Conference on Computer Design (ICCD)*. Oct. 2018, pp. 346–349.
- [C107] F. N. Taher, M. Kishani, and **B. Carrion Schafer**. “Design and Optimization of Reliable Hardware Accelerators: Leveraging the Advantages of High-Level Synthesis”. In: *IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*. July 2018, pp. 232–235.
- [C107] Shuangnan Liu, Francis Lau, and **B. Carrion Schafer**. “Investigation and Optimization of Pin Multiplexing in High-Level Synthesis”. In: *Great Lakes Symposium on VLSI. GLSVLSI '18*. Chicago, IL, USA: Association for Computing Machinery, 2018, pp. 427–430.
- [C107] Farah Naz Taher, Joseph Callenes-Sloan, and **B. Carrion Schafer**. “A Machine Learning Based Hard Fault Recuperation Model for Approximate Hardware Accelerators”. In: *55th Annual Design Automation Conference. DAC '18*. San Francisco, California: Association for Computing Machinery, 2018.
- [C107] S. Xu, J. Chen, and **B. Carrion Schafer**. “HW/SW co-design experimental framework using configurable SoCs”. In: *International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. Dec. 2017, pp. 1–6.
- [C107] S. Xu, **B. Carrion Schafer**, and Y. Liu. “Configurable SoC In-Situ Hardware/Software Co-Design Design Space Exploration”. In: *IEEE International Conference on Computer Design (ICCD)*. Nov. 2017, pp. 509–512.
- [C107] S. Xu and **B. Carrion Schafer**. “Approximate Reconfigurable Hardware Accelerators: Adapting the Micro-architecture to Dynamic Workloads”. In: *IEEE International Conference on Computer Design (ICCD)*. Nov. 2017, pp. 1–7.
- [C107] S. Liu and **B. Carrion Schafer**. “Learning-based interconnect-aware dataflow accelerator optimization”. In: *27th International Conference on Field Programmable Logic and Applications (FPL)*. Sept. 2017, pp. 1–7.
- [C107] **B. Carrion Schafer**, David Aledo, and Félix Moreno. “Application Specific Behavioral Synthesis Design Space Exploration: Artificial Neural Networks. A Case Study”. In: *Euromicro Conference on Digital System Design, DSD*. 2017, pp. 129–136.
- [C107] Y. Liu, M. Villaverde, F. Moreno, and **B. Carrion Schafer**. “Characterization and optimization of behavioral hardware accelerators in heterogeneous MPSoCs”. In: *12th*

- International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC). July 2017, pp. 1–8.
- [C107] Muhammad Yasin, Abhrajit Sengupta, **B. Carrion Schafer** Schafer, Yiorgos Makris, Ozgur Sinanoglu, and Jeyavijayan (JV) Rajendran. “What to Lock? Functional and Parametric Lock- ing”. In: *Great Lakes Symposium on VLSI 2017. GLSVLSI ’17*. Banff, Alberta, Canada: Association for Computing Machinery, 2017, pp. 351–356.
- [C107] N. Veeranna and **B. Carrion Schafer**. “Efficient behavioral intellectual properties source code obfuscation for high-level synthesis”. In: *2017 18th IEEE Latin American Test Symposium (LATS)*. Mar. 2017, pp. 1–6.
- [C107] N. Veeranna and **B. Carrion Schafer**. “Hardware Trojan avoidance and detection for dynamically re-configurable FPGAs”. In: *International Conference on Field-Programmable Technology (FPT)*. Dec. 2016, pp. 193–196.
- [C107] Dong Liu and **B. Carrion Schafer**. “Efficient and reliable High-Level Synthesis Design Space Explorer for FPGAs”. In: *26th International Conference on Field Programmable Logic and Applications (FPL)*. Aug. 2016, pp. 1–8.
- [C107] A. Balachandran, N. Veeranna, and **B. Carrion Schafer**. “On Time Redundancy of Fault Tol- erant C-Based MPSoCs”. In: *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. July 2016, pp. 631–636.
- [C107] Y. Liu and **B. Carrion Schafer**. “Optimization of behavioral IPs in multi-processor system- on-chips”. In: *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. Jan. 2016, pp. 336–341.
- [C107] I. Llamas-Garro, J. Bas, J. M. F`abrega, M. S. Moreolo, **B. Carrion Schafer**, R. Torres-Torres, M. R. T. de Oliveira, M. T. de Melo, Jung-Mu Kim, and D. Vukobratovic. “Recent trends and considerations for high speed data in chips and system interconnects”. In: *SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*. Nov. 2015, pp. 1–6.
- [C107] Xiaotong Li and **B. Carrion Schafer**. “Temperature-triggered behavioral IPs HW Trojan detection method with FPGAs”. In: *25th International Conference on Field Programmable Logic and Applications (FPL)*. Sept. 2015, pp. 1–4.
- [C107] Yidi Liu and **B. Carrion Schafer**. “Adaptive combined macro and micro-exploration of con- current applications mapped on shared bus Reconfigurable SoC”. In: *Electronic System Level Synthesis Conference (ESLsyn)*. June 2015, pp. 11–16.
- [C107] **B. Carrion Schafer**. “Process selection for maximum resource sharing in High-Level Synthesis”. In: *Electronic System Level Synthesis Conference (ESLsyn)*. June 2015, pp. 35–40.
- [C107] Y. Liu and **B. Carrion Schafer**. “HW acceleration of multiple applications on a single FPGA”. In: *International Conference on Field-Programmable Technology (FPT)*. Dec. 2014, pp. 284–285.
- [C107] **B. Carrion Schafer**. “Time sharing of Runtime Coarse-Grain Reconfigurable Architectures processing elements in multi-process systems”. In: *International Conference on Field- Programmable Technology (FPT)*. Dec. 2014, pp. 76–82.
- [C107] A. Mahapatra and **B. Carrion Schafer**. “Machine-learning based simulated annealer method

- for high level synthesis design space exploration”. In: *Proceedings of the 2014 Electronic System Level Synthesis Conference (ESLsyn)*. May 2014, pp. 1–6.
- [C107] **B. Carrion Schafer**. “Allocation of FPGA DSP-macros in multi-process high-level synthesis systems”. In: *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*. Jan. 2014, pp. 616–621.
- [C107] **B. Carrion Schafer**. “Automatic partitioning of behavioral descriptions for high-level synthesis with multiple internal throughputs”. In: *Electronic System Level Synthesis Conference (ESLsyn)*. May 2013, pp. 1–6.
- [C107] Shahin Golshan, Eli Bozorgzadeh, **B. Carrion Schafer**, Kazutoshi Wakabayashi, Houman Homayoun, and Alex Veidenbaum. “Exploiting Power Budgeting in Thermal-Aware Dynamic Placement for Reconfigurable Systems”. In: *Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design. ISLPED 2010*. Austin, Texas, USA: ACM, 2010, pp. 49–54.
- [C107] **B. Carrion Schafer**, Takashi Takenaka, and Kazutoshi Wakabayashi. “Adaptive Simulated Annealer for high level synthesis design space exploration”. In: *2009 International Symposium on VLSI Design, Automation and Test*. Apr. 2009, pp. 106–109.
- [C107] **B. Carrion Schafer**, Y. Lee, and T. Kim. “Temperature-Aware Compilation for VLIWProcessors”. In: *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007)*. Aug. 2007, pp. 426–431.
- [C107] **B. Carrion Schafer** and T. Kim. “Thermal-Aware Instruction Assignment for VLIW Processors”. In: *11th Workshop on Interaction between Compilers and Computer Architectures (INTERACT-11)*. Feb. 2007.
- [C107] **B. Carrion Schafer**, S. F. Quigley, and A. H. C. Chan. “Scalable Implementation of the Discrete Element Method on a Reconfigurable Computing Platform”. In: *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 925–934.
- [C107] **B. Carrion Schafer**, S. F. Quigley, and A. H. C. Chan. “Implementation of the Discrete Element Method using reconfigurable computing (FPGAs),” in: *15th Engineering Mechanics Conference (EM2002)*. 2002.
- [C107] **B. Carrion Schafer**, S. F. Quigley, and A. H. C. Chan. “Analysis and implementation of the discrete element method using a dedicated highly parallel architecture in reconfigurable computing”. In: *10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*. 2002, pp. 173–181.
- [C107] **B. Carrion Schafer**, S. F. Quigley, and A. H. C. Chan. “Description of a Dedicated Hardware Architecture for the Discrete Element Method (DEM) implemented on a Field Programmable Gate Array (FPGA),” in: *10th Annual conference of the Association for Computational Mechanics in Engineering (ACME)*. 2002, pp. 51–54.
- [C107] **B. Carrion Schafer**, Steven F. Quigley, and Andrew H. C. Chan. “Evaluation of an FPGA Implementation of the Discrete Element Method”. In: *Field-Programmable Logic and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 306–314.
- [C107] **B. Carrion Schafer**, S. F. Quigley, and A. H. C. Chan. “Numeric Modeling of the Mechanical interaction between non-biological particles using reconfigurable computing”. In:

9th Annual conference of the Association for Computational Mechanics in Engineering (ACME). 2001, pp. 53–56.

Patents

[P2] **B. Carrion Schafer.** Method and Apparatus for Incremental Design Space Exploration. U.S. Patent 8,977,996 B2.

[P2] **B. Carrion Schafer.** Method and Apparatus for Design Space Exploration. JP Patent 2012-522281.

RESEARCH IMPACT AND CITATIONS

h-index (Google Scholar January 2025): 26

i10-index (Google Scholar January 2025): 45

Citations (Google Scholar January 2025): 2402

SOFTWARE ARTIFACTS RELEASED

FREEDOM: FPGA-based Hardware Redaction Emulator

Hardware redaction emulation framework that allows to quickly prototype ASIC+eFPGA designs onto a single low-cost FPGA. The emulator is available at <https://github.com/chaitalisathe/FREEDOM>. The framework only requires a Terasic DE1-SoC standard board.

DSE2Frame: DSE Explorer Framework

DSEFrame is a QT-based open source development framework to create and visualize High-Level Synthesis (HLS) design space exploration results. The development framework is freely available online at <https://github.com/DARCLabHao/DSEframe>.

S2CBench: Synthesizable SystemC Benchmark Suite

S2CBench is collection of over 20 SystemC benchmarks from different domain that comply with the latest Accelera's SystemC synthesizable subset. The benchmark suite is freely available online at <https://sourceforge.net/projects/s2cbench/> from where it has been downloaded over 1,000 times.

S3CBench: Security Synthesizable SystemC Benchmark Suite

S3CBench is the extension of the S2CBench benchmarks instrumented with malicious alterations a.k.a. Hardware Trojan that when triggered lead to deviation of the benchmark from its original intended behavior. The benchmark suite is freely available online at <https://sourceforge.net/projects/s3cbench/> from where it has been downloaded over 300 times.

UNIVERSITY COURSES TAUGHT

At the University of Texas at Dallas

<u>Year</u>	<u>Semester</u>	<u>Code</u>	<u>Type</u>	<u>Title, Credits, Students</u>
2025	Fall	EEDG/CE6370	Grad/elec	Reconfigurable Systems, 3, 48
2024	Fall	EEDG/CE6370	Grad/elec	Reconfigurable Systems, 3, 43
2024	Summer	EE/CE4304	Under/core	Computer Architecture, 3, 14
2023	Fall	EEDG/CE6304	Grad/core	Computer Architecture, 3, 60
2023	Fall	EEDG/CE6370	Grad/elec	Reconfigurable Systems, 3, 48
2023	Summer	EE/CE4304	Under/core	Computer Architecture, 3, 16

<u>2022</u>	<u>Fall</u>	<u>EE/CE4307</u>	<u>Under/core</u>	<u>Embedded Systems, 3, 37</u>
<u>2022</u>	<u>Fall</u>	<u>EEDG/CE6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 55</u>
<u>2022</u>	<u>Summer</u>	<u>EE/CE4304</u>	<u>Under/core</u>	<u>Computer Architecture, 3, 10</u>
<u>2022</u>	<u>Spring</u>	<u>EE/CE4307</u>	<u>Under/core</u>	<u>Embedded Systems, 3, 60</u>
<u>2021</u>	<u>Fall</u>	<u>EEDG/CE6331</u>	<u>Grad/elec</u>	<u>High-Level Synthesis, 3, 8</u>
<u>2021</u>	<u>Fall</u>	<u>EEDG/CE6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 60</u>
<u>2021</u>	<u>Spring</u>	<u>EE/CE4307</u>	<u>Under/core</u>	<u>Embedded Systems, 3, 55</u>
<u>2020</u>	<u>Spring</u>	<u>EEDG7V81</u>	<u>Grad/elec</u>	<u>High-Level Synthesis, 3, 7</u>
<u>2020</u>	<u>Fall</u>	<u>EEDG/CE6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 43</u>
<u>2020</u>	<u>Spring</u>	<u>EE/CE4307</u>	<u>Under/core</u>	<u>Embedded Systems, 3, 60</u>
<u>2019</u>	<u>Fall</u>	<u>EEDG7V81</u>	<u>Grad/elec</u>	<u>High-Level Synthesis, 3, 7</u>
<u>2019</u>	<u>Fall</u>	<u>EEDG/CE6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 43</u>
<u>2019</u>	<u>Spring</u>	<u>EE/CE4307</u>	<u>Under/core</u>	<u>Embedded Systems, 3, 49</u>
<u>2018</u>	<u>Fall</u>	<u>EEDG7V81</u>	<u>Grad/elec</u>	<u>High-Level Synthesis, 3, 15</u>
<u>2018</u>	<u>Fall</u>	<u>EEDG6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 13</u>
<u>2018</u>	<u>Spring</u>	<u>EEDG6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 13</u>
<u>2018</u>	<u>Spring</u>	<u>EE/CE4370</u>	<u>Under/core</u>	<u>Embedded Systems, 3, 43</u>
<u>2017</u>	<u>Fall</u>	<u>EEDG/CE6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 16</u>
<u>2017</u>	<u>Spring</u>	<u>EEDG/CE6304</u>	<u>Grad/core</u>	<u>Computer Architecture, 3, 16</u>

At the Hong Kong Polytechnic University

<u>Year</u>	<u>Semester</u>	<u>Code</u>	<u>Type</u>	<u>Title, Credits, Students</u>
<u>2016</u>	<u>Spring</u>	<u>EIE4110</u>	<u>Under/elect</u>	<u>VLSI&CAD, 3, 31</u>
<u>2015</u>	<u>Fall</u>	<u>EIE511</u>	<u>Grad/elect</u>	<u>VLSI System design, 3, 15</u>
<u>2014</u>	<u>Spring</u>	<u>EIE4110</u>	<u>Under/elect</u>	<u>VLSI&CAD, 3, 63</u>
<u>2014</u>	<u>Spring</u>	<u>EIE3361</u>	<u>Under/core</u>	<u>Computer Fund, 3, 62</u>
<u>2014</u>	<u>Fall</u>	<u>EIE511</u>	<u>Grad/elect</u>	<u>VLSI System design, 3, 26</u>
<u>2014</u>	<u>Fall</u>	<u>EIE2105</u>	<u>Under/core</u>	<u>Digital Design, 3, 28</u>
<u>2014</u>	<u>Spring</u>	<u>EIE4110</u>	<u>Under/elect</u>	<u>VLSI&CAD, 3, 29</u>
<u>2014</u>	<u>Spring</u>	<u>EIE3361</u>	<u>Under/core</u>	<u>Computer Fund, 3, 74</u>
<u>2013</u>	<u>Fall</u>	<u>EIE2105</u>	<u>Under/elect</u>	<u>Digital Design, 3, 35</u>
<u>2013</u>	<u>Spring</u>	<u>EIE305</u>	<u>Under/elect</u>	<u>Digital Circuits, 3, 18</u>
<u>2013</u>	<u>Spring</u>	<u>EIE361</u>	<u>Under/core</u>	<u>Computer Fund, 3, 56</u>
<u>2013</u>	<u>Spring</u>	<u>EIE410</u>	<u>Under/elect</u>	<u>VLSI&CAD, 3, 22</u>

STUDENT SUPERVISION SUMMARY

<u>Ph.D. students currently supervised:</u>	<u>6</u>
<u>M.S. students currently supervised:</u>	<u>3</u>
<u>Ph.D. students graduated:</u>	<u>15</u>
<u>M.S. students graduated:</u>	<u>20</u>
<u>Undergraduate researchers supervised:</u>	<u>9</u>
<u>High School Students supervised:</u>	<u>3</u>

GRADUATE STUDENTS CURRENTLY SUPERVISED

Doctoral Students

<u>Ph.D. Candidate</u>	<u>Research Area</u>	<u>Years</u>
<u>Sabrina Ahmed</u>	<u>Sustainable HW Design</u>	<u>2023–present</u>
<u>Victoria Gammenthaler</u>	<u>Hardware Security</u>	<u>2024–present</u>

Daksith Chandrasekera	LLM-based EDA	2025–present
Salam Kadhim Al-Khammasi	LLM-based VLSI System Generation	2026–present
Adam Kayyal	LLM-based Design Optimization	2026–present
Emmanuel Llanes	TBD (starting in fall)	2026–present

Master Students

<u>MS</u>	<u>Research Area</u>	<u>Years</u>
Mark Ivey	Efficient LLM implementations on FPGAs	2025–present
Sathwik Sudhindranath	Complex DSP applications on FPGAs	2026–present
Samarth Nandeesh	Custom FPGA fabrics for HLS generated Designs	2026–present

GRADUATE STUDENTS PREVIOUSLY SUPERVISED

Doctoral Advisement/Direction

<u>Name</u>	<u>Year</u>	<u>Dissertation (compact)</u>	<u>Job</u>
Baharealsadat Parchamdar	2026	HLS optimization/applications	=
Chaitali Sathe	2025	HW IP security attacks/defense	=
Md Imtiaz Rashid	2024	HLS circuit modernization	Univ. of Florida (Postdoc)
Qilin Si	2023	Behavioral processors	NVIDIA
Prattay Chowdhury	2022	Approximate computing systems	Qualcomm
Zi Wang	2021	HLS design space search	Cadence
Bo Hu	2021	Thermal/security HLS design	Apple
Jianqi Chen	2020	HLS security/low-power design	Cirrus Logic
Anjana Balachandran	2020	Fault-tolerant circuits	Huawei
Zhiqi Zhu	2020	Fault-tolerant accelerators	Nubis Comm.
Farah Naz Taher	2019	HW fault detection/mitigation	Raytheon
Siyuan Xu	2019	Robust approximate computing	MathWorks
Shuangnan Liu	2019	Behavioral dataflow design	Cadence
Ansuhee Mahapatra	2018	RTL abstraction for HLS	ASTRI
Nandeesh Veeranna	2017	BIP protection	NTU (Postdoc)

Masters Advisement/Direction

<u>Name</u>	<u>Year</u>	<u>Dissertation (short)</u>	<u>Job After Graduation</u>
Adam Kayyal	2026	Synthesis-aware HLS DSE	UT Dallas (PhD), USA
Valliyappan Senthilkumar	2023	ANN optimization HW/SW	Marvell Tech., USA
Amir H. Torabi	2023	HLS quality estimation	UT Dallas (PhD), USA
Santosh Shetty	2020	Behavioral SoC generation	AMD, India
Akshay R. Krisnani	2020	CNN in-situ training	Intel, USA
Yiheng Gao	2020	HLS design space search	Nokia, USA
Rohit Sreekumar	2020	Bespoke processors	Amazon, USA
Himanshu Patra	2019	HLS language conversion	Intel, USA
Maheswaran R. Babu	2019	FPGA diversity accel.	Intel, USA
Mihir Shah	2018	Partial reconfig. architecture	Varex Imaging, USA
Monica Gowda	2018	HLS exploration	MicroChip, USA
Vinay Dasandi	2018	Bitwidth optimization	Delphi, USA
Songseok Choi	2018	FPGA acceleration automation	SK Hynix, Korea
Susmitha Gogineni	2017	OpenCL accel.	Texas Instruments, USA
Siyuan Xu	2015	SoC design exploration	PhD, UT Dallas
Zhendong Gao	2015	FPGA voice viz.	PCCW, Hong Kong

<u>Jieshi Chen</u>	<u>2015</u>	<u>(not specified)</u>	<u>DJI, China</u>
<u>Yidi Liu</u>	<u>2015</u>	<u>Parallel HW acceleration</u>	<u>Huawei</u>
<u>Yu Li</u>	<u>2014</u>	<u>FPGA gaming console</u>	<u>ZTE, China</u>
<u>Xiaotong Li</u>	<u>2014</u>	<u>HW Trojan detection</u>	<u>TCL, China</u>

VISITING GRADUATE STUDENTS SUPERVISED

<u>Takafumi Miyazaki (Visiting from Ritsumeikan University, Japan)</u>	<u>09/2018-10/2018</u>
<u>Shuangnan Liu (Visiting from Hong Kong Polytechnic University, HKSAR)</u>	<u>09/2017-12/2018</u>
<u>Nandeesh Veeranna (Visiting from Hong Kong Polytechnic University, HKSAR)</u>	<u>04/2017-07/2017</u>
<u>Monica Villaverde (Visiting from Universidad Politecnica de Madrid, Spain)</u>	<u>01/2017-04/2017</u>

UNDERGRADUATE AND HIGH SCHOOL STUDENTS SUPERVISED

Undergraduate Students

<u>Juan Diego Rosero Rodriguez</u>	<u>Spring 2026</u>
<u>Omar Abiola Abioye</u>	<u>Spring 2025</u>
<u>Shruthigna Chandupatla</u>	<u>Spring 2024</u>
<u>Axel Gamez (Qualcomm Research Award Recipient)</u>	<u>Fall 2022</u>
<u>Preston T. Glenn</u>	<u>Summer and Fall 2021</u>
<u>Kimberly Klein (Qualcomm Research Award Recipient)</u>	<u>Spring 2021</u>
<u>Michael Nelson</u>	<u>Fall and Spring 2017/2018</u>
<u>Rodolfo Martinez</u>	<u>Summer 2018</u>
<u>Ryota Watanabe</u>	<u>Summer 2018</u>
<u>Kristen Nguyen</u>	<u>Summer 2018</u>
<u>Sean Kennedy</u>	<u>Summer 2018</u>

High School Students

<u>Ishaan Javali</u>	<u>Summer 2019</u>
<u>Alondra Ramos</u>	<u>Summer 2018</u>
<u>Natasha Trayers</u>	<u>Summer 2018</u>

KEYNOTES, INVITED TALKS AND SHORT COURSES TAUGHT

B. Carrion Schaefer, “Fast and Efficient Domain Specific Hardware Design”, The University of Houston, USA, August, 2024.

B. Carrion Schaefer, “C-based VLSI Design: Making Hardware Design Easier”, Hiroshima City University, Japan, July, 2024.

B. Carrion Schaefer, “Career Development: Hiring and Promotion”, DAC Early Career Workshop, San Francisco, USA, June, 2024.

B. Carrion Schaefer and Jeff Roane, “HLS Adoption and Key Technologies: Rethinking the use of High-Level Synthesis: From FPGA Prototyping to ASIC Design”, Cadence Lunch and Learn, Dallas, USA, August, 2023.

B. Carrion Schaefer, “Techniques for Finding Security Vulnerabilities in SoCs”, ACM/IEEE Design Automation Conference (DAC), San Francisco, USA, June, 2018.

B. Carrion Schafer, “Behavioral IPs Micro-architectural Diversity and its Applications”, ESLSyn, San Francisco, Co-located with DAC, June 2015,

B. Carrion Schaefer and J.H. Anderson, “FPGA high-level synthesis: from software to programmable

hardware”, tutorial delivered at the ACM/IEEE Asia-South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January 2015.

A. Mahapatra and B. Carrion Schaefer, “S2CBench:Synthesizable SystemC Benchmark Suite for HighLevel Synthesis”, 20th meeting of the North American SystemC Users Group,San Francisco, June, 2014.

B. Carrion Schaefer,“The ESL Hotspot - Where Software and Hardware Meet”, Design Automation Conference (DAC), 10th annual ESL Symposium, San Francisco, (invited panelist), 2012.

B. Carrion Schaefer,“High-Level Synthesis Production Deployment: Are We Ready?”, Design Au- tomation Conference (DAC), San Francisco, (invited speaker), 2012.

B. Carrion Schaefer,“Challenges and Opportunities of Behavioral Level SoC Design”, ESLSyn, Co- located with DAC, San Francisco, (Keynote speaker), 2012.

GRANTS, CONTRACTS, GIFTS AND DONATIONS

<u>Sponsor Type</u>	<u>Total Funding (\$)</u>	<u>My Share (\$)</u>
<u>U.S. Federal (NSF, ONR, AFRL/AFWERX)</u>	<u>3,347,139</u>	<u>1,211,799</u>
<u>Industry (Boeing, Huawei, Texas Instruments)</u>	<u>224,000</u>	<u>107,000</u>
<u>International Government (NATO, HK UGC, NSFC China)</u>	<u>443,000</u>	<u>134,333</u>
<u>University / Internal (JSRI)</u>	<u>20,000</u>	<u>10,000</u>
<u>Equipment / Software Grants</u>	<u>25,000</u>	<u>25,000</u>
<u>Total</u>	<u>4,059,139</u>	<u>1,488,132</u>

Office of Naval Research (ONR), Self-aware Computing Through AI (Sole PI). 2025-2027

Total amount : \$169,729

NSF, Advancing Innovation in Master of Science and Engineering Scholars Career Development (co-PI w. PI Pearson, UTD).

2025-2030

Total amount : \$2,000,000

NSF CHEST I/UCRC, Investigating the Weaknesses of IP Protection through eFPGA- based Hardware Redaction (PI w. co-PI Makris). Continuation grant. 2024-2025

Total amount : \$75,000

NSF, Collaborative Research: DESC: Type I: SEED: Sustainability-aware Reliable and Reusable AI Hardware Design (co-PI w. PI Hoque, University of Missouri).

2023-2026

Total amount : \$600,000

Amount UTD: \$260,00

NSF CHEST I/UCRC, Investigating the Weaknesses of IP Protection through eFPGA- based Hardware Redaction (PI w. co-PI Makris).

2023-2024

Total amount : \$65,000

Jonsson School Research Initiative (JSRI), UTD, Methods for Practical Design for Trust of Complex System-on-Chip (PI w. Co-PI Wong)

2023-2024

Total amount : \$20,000

NSF CHEST I/UCRC,CAD Flow Development for Design Obfuscation through Post-Fabrication Transistor-Level Programming(co-PI w. PI-Makris and Sechen).

<u>Continuation grant</u>	<u>2022-2023</u>
Total amount : \$125,000	
<u>NSF CHEST I/UCRC,CAD Flow Development for Design Obfuscation through Post-Fabrication Transistor-Level Programming(co-PI w. PI-Makris and Sechen).</u>	
<u>Continuation grant</u>	<u>2021-2022</u>
Total amount : \$125,000	
<u>AFWERX, Microelectronics Challenge Phase III (lead-PI w. Makris and Sechen)</u>	<u>2020</u>
Total amount : \$200,000	
<u>NSF CHEST I/UCRC,CAD Flow Development for Design Obfuscation through Post-Fabrication Transistor-Level Programming(co-PI w. PI-Makris and Sechen)</u>	<u>2020-2021</u>
Total amount : \$100,000	
<u>Boeing/AFRL (co-PI w. PI-Makris and Sechen)</u>	<u>2020</u>
Total amount : \$150,000	
<u>AFWERX, Microelectronics Challenge Phase II (lead-PI w. Makris and Sechen)</u>	<u>2019</u>
Total amount : \$88,410	
<u>NATO, Nerve Agent Detection, (co-PI w. PI-Llamas and Mung)</u>	<u>2019-2022</u>
Total amount : \$236,000	
<u>Huawei USA, Based Band SoC Approximate Computing (Sole PI)</u>	<u>2018</u>
Total amount : \$60,000	
<u>Texas Instruments (Sole PI)</u>	<u>2018</u>
Total amount : \$14,000	
<u>NEC Corporation, Software Grant (CyberWorkBench Software license donation)</u>	<u>2017</u>
Value: \$10,000	
<u>Altera Equipment Grant (Arria SoC and DE1 FPGA Boards)</u>	<u>2017</u>
Value: \$7,000	
<u>Xilinx Equipment Grant (Virtex-5 and Zynq FPGA Boards)</u>	<u>2017</u>
Total amount : \$8,000	
<u>Early Career Scheme, University Grant Council, Hong Kong (Sole PI)</u>	<u>2014</u>
Total amount : HKD \$677,000 (\$87,000)	
<u>National Natural Science Foundation of China (co-PI w. PI-Lau and Ho)</u>	<u>2014</u>
Total amount : RMB \$750,000 (\$120,000)	

PROFESSIONAL MEMBERSHIPS AND LICENSES

IEEE Senior Member since 2010

Last updated May 4, 2026.