US 20220113968A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2022/0113968 A1**

Harthcock (43) **Pub. Date: Apr. 14, 2022**

(54) **FULLY PIPELINED BINARY CONVERSION HARDWARE OPERATOR LOGIC CIRCUIT**

(71) Applicant: **Jerry D. Harthcock**, Boerne, TX (US)

(72) Inventor: **Jerry D. Harthcock**, Boerne, TX (US)

(21) Appl. No.: **17/555,408**

(22) Filed: **Dec. 18, 2021**

Related U.S. Application Data

(62) Division of application No. 16/943,077, filed on Jul. 30, 2020.

(60) Provisional application No. 62/886,570, filed on Aug. 14, 2019.

Publication Classification

(51) **Int. Cl.**

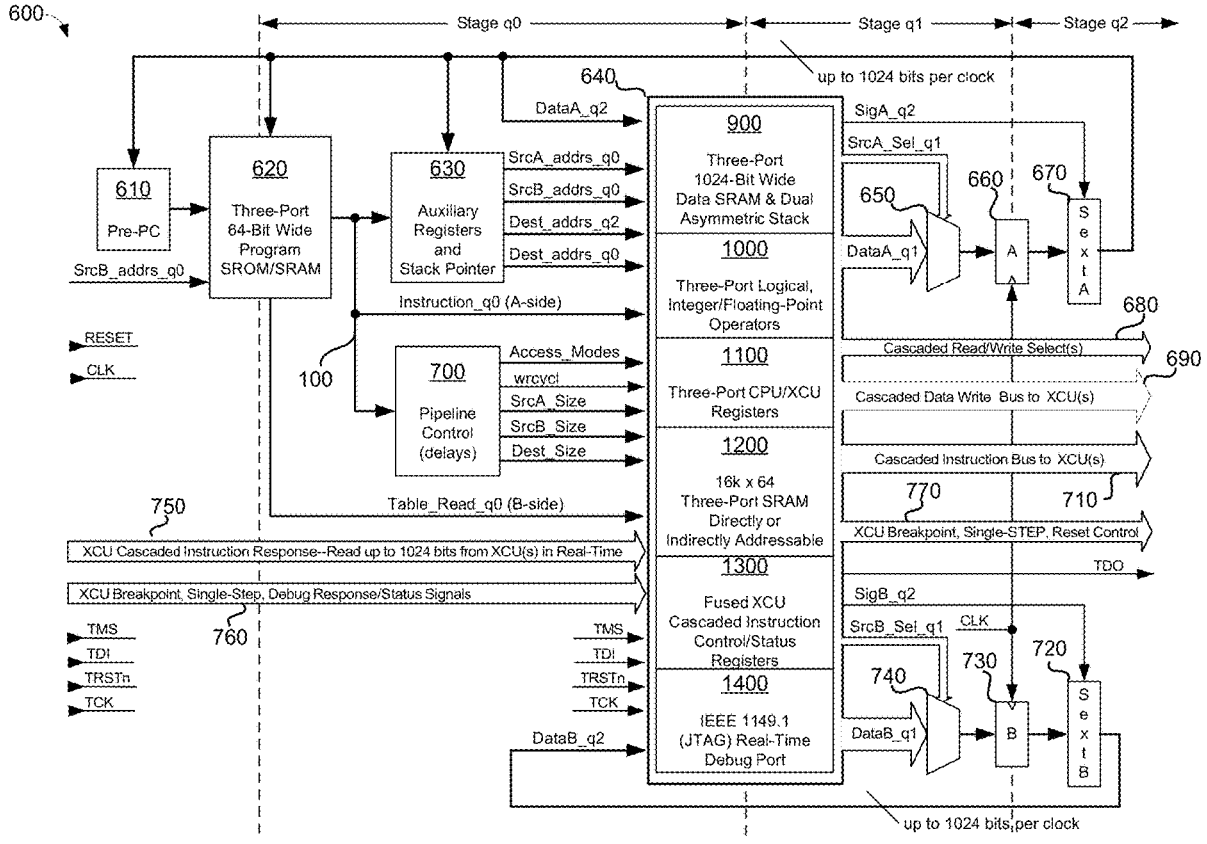
G06F 9/30	(2006.01)
G06F 12/06	(2006.01)
G06F 9/355	(2006.01)
G06F 9/54	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 9/30025*** (2013.01); ***G06F 9/30079*** (2013.01); ***G06F 9/30101*** (2013.01); ***G06F 9/3557*** (2013.01); ***G06F 9/544*** (2013.01); ***G06F 12/063*** (2013.01)

(57) **ABSTRACT**

A universal floating-point Instruction Set Architecture (ISA) implemented entirely in hardware. Using a single instruction, the universal floating-point ISA has the ability, in hardware, to compute directly with dual decimal character sequences up to IEEE 754-2008 "H=20" in length, without first having to explicitly perform a conversion-to-binary-format process in software before computing with these human-readable floating-point or integer representations. The ISA does not employ opcodes, but rather pushes and pulls "gobs" of data without the encumbering opcode fetch, decode, and execute bottleneck. Instead, the ISA employs stand-alone, memory-mapped operators, complete with their own pipeline that is completely decoupled from the processor's primary push-pull pipeline. The ISA employs special three-port, 1024-bit wide SRAMS; a special dual asymmetric system stack; memory-mapped stand-alone hardware operators with private result buffers having simultaneously readable side-A and side-B read ports; and dual hardware H=20 convertFromDecimalCharacter conversion operators.



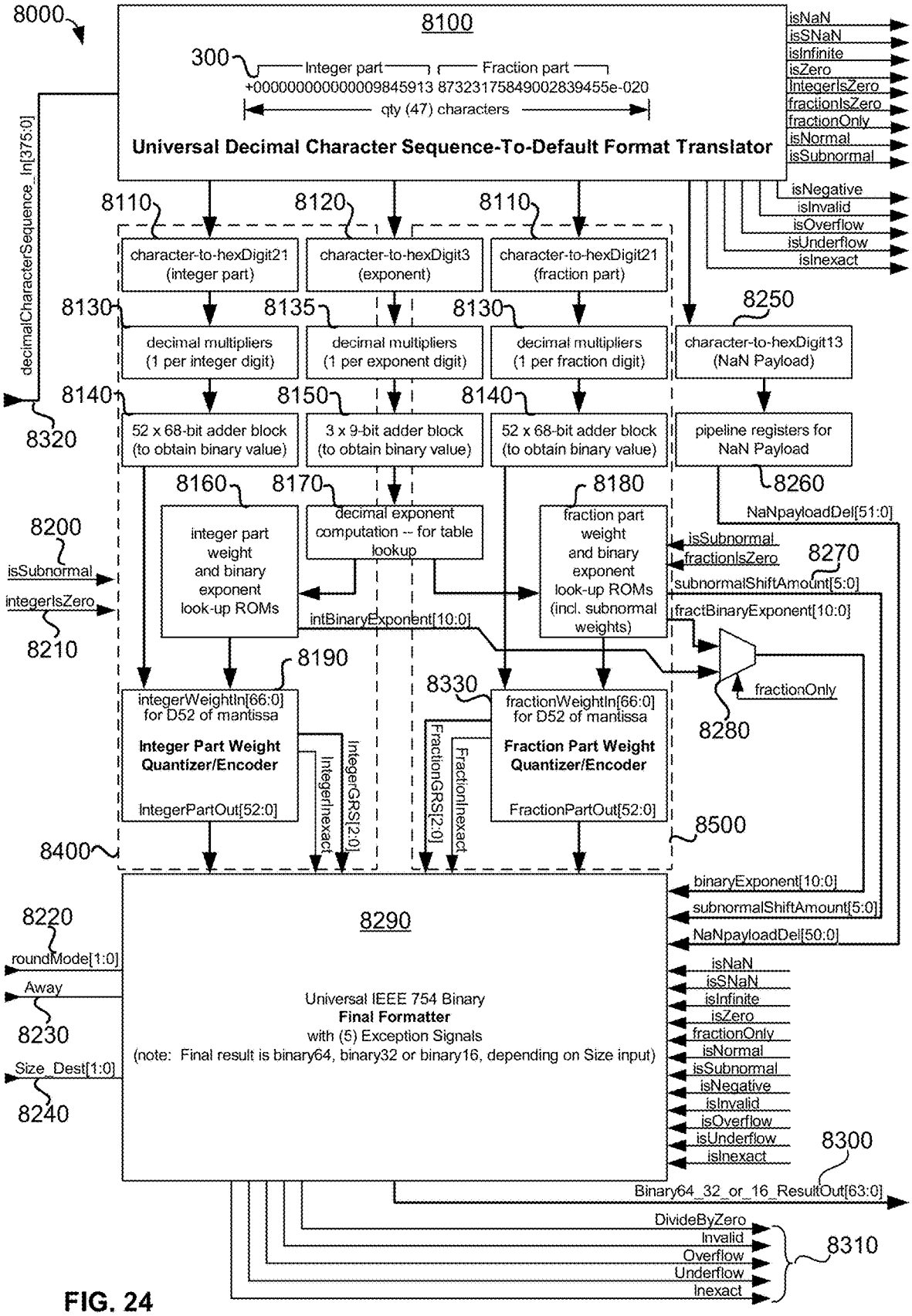

8000
8100
300
Integer part
Fraction part
+00000000000000009845913 8732317584900283945 5e-020
qty (47) characters
Universal Decimal Character Sequence-To-Default Format Translator
decimalCharacterSequence_In[375:0]
isNaN
isSNaN
isInfinite
isZero
IntegerIsZero
fractionIsZero
fractionOnly
isNormal
isSubnormal
isNegative
isInvalid
isOverflow
isUnderflow
isInexact
8110
8120
8110
character-to-hexDigit21 (integer part)
character-to-hexDigit3 (exponent)
character-to-hexDigit21 (fraction part)
8130
8135
8130
decimal multipliers (1 per integer digit)
decimal multipliers (1 per exponent digit)
decimal multipliers (1 per fraction digit)
8250
character-to-hexDigit13 (NaN Payload)
8140
8150
8140
52 x 68-bit adder block (to obtain binary value)
3 x 9-bit adder block (to obtain binary value)
52 x 68-bit adder block (to obtain binary value)
pipeline registers for NaN Payload
8320
8160
8170
8180
8260
NaNpayloadDel[51:0]
integer part weight and binary exponent look-up ROMs
decimal exponent computation -- for table lookup
fraction part weight and binary exponent look-up ROMs (incl. subnormal weights)
8200
isSubnormal
integerIsZero
8210
isSubnormal
fractionIsZero
8270
subnormalShiftAmount[5:0]
fractBinaryExponent[10:0]
intBinaryExponent[10:0]
8190
8330
integerWeightIn[66:0] for D52 of mantissa
Integer Part Weight Quantizer/Encoder
IntegerPartOut[52:0]
fractionWeightIn[66:0] for D52 of mantissa
Fraction Part Weight Quantizer/Encoder
FractionPartOut[52:0]
fractionOnly
8280
IntegerInexact
IntegerGRS[2:0]
FractionGRS[2:0]
FractionInexact
8500
8400
binaryExponent[10:0]
subnormalShiftAmount[5:0]
NaNpayloadDel[50:0]
8220
8290
roundMode[1:0]
Away
8230
Size_Dest[1:0]
8240
Universal IEEE 754 Binary
Final Formatter
with (5) Exception Signals
(note: Final result is binary64, binary32 or binary16, depending on Size input)
isNaN
isSNaN
isInfinite
isZero
fractionOnly
isNormal
isSubnormal
isNegative
isInvalid
isOverflow
isUnderflow
isInexact
8300
Binary64_32_or_16_ResultOut[63:0]
DivideByZero
Invalid
Overflow
Underflow
Inexact
8310

FIG. 24

the triangles. During this time, each XCU brings its DONE bit inactive low, to signal it is busy. At this time the parent is monitoring its XCU status register for the DONE bits to be brought back to active high, indicating XCU processing is complete. When complete, the parent XCU then pulls the transformed triangles from each XCU result memory and pushes them back out to external memory.

[0598] If, at the initial stages described above, the parent CPU determines there are no XCUs available to perform the transform, the parent CPU performs the entire 3D transform solo (without the use of any XCU) and pushes the transformed triangles back out to external memory when complete. It should be understood that the parent CPU and the child XCUs (if any) execute the same instruction set. In this implementation, the 3D transform routine physically resides in the parent CPU program memory space. Thus, in this instance, when the parent performs a "push-all" of the required routine and parameters, it pulls the routine code out of its program memory space and pushes it into all XCUs program memories simultaneously. If the required program is not resident in the CPU's program space, it could alternatively pull it in from any child XCU program memory space or external memory space.

[0599] FIG. **51** is an actual wire-frame "Before" and "After" rendering **9955** of a simple "olive" 3D model in .STL file format performed by from 1 to 16 child XCUs or solo parent CPU using the scale, rotate, and translate parameters shown for each axis.

[0600] In the drawings and specification, there have been disclosed typical preferred embodiments of the disclosure and, although specific terms are employed, they are used in a generic and descriptive sense only and not for purposes of limitation, the scope of the invention being set forth in the following claims.

1-18. (canceled)

19. A fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit configured to convert one or more human-readable decimal character sequence floating-point representations to IEEE 754-2008 binary floating-point representations every clock cycle, said hardware operator logic circuit comprising:

a hardware-implemented Decimal Character Sequence Format Translator logic front end that separates an integer part, a fraction part, and an exponent part of a character sequence and places the integer, fraction, and exponent parts into respective assigned character positions of a default predetermined character sequence format and delivers an integer part character sequence to a hardware-implemented integer part mantissa logic, a fraction part character sequence to a hardware-implemented fraction part mantissa logic, and an exponent part character sequence to both a hardware-implemented integer part exponent conversion logic and a hardware-implemented fraction part exponent conversion logic;

wherein the integer part mantissa logic is configured to convert a delivered integer part character sequence into an equivalent decimal value representing an integer part mantissa and to deliver a converted decimal value representing the integer part mantissa to an integer part weight quantizer/encoder logic;

wherein the fraction part mantissa logic is configured to convert a delivered fraction part character sequence into an equivalent decimal value representing a fraction part mantissa and to deliver a converted decimal value representing the fraction part mantissa to a fraction part weight quantizer/encoder logic;

wherein the integer part exponent conversion logic is configured to convert the delivered exponent part character sequence into an equivalent decimal value representing an integer part exponent and to deliver a converted exponent decimal value representing an integer part exponent to a hardware-implemented integer part exponent look-up table/ROM and interpolation logic;

wherein the fraction part exponent conversion logic is configured to convert the delivered exponent part character sequence into an equivalent decimal value representing a fraction part exponent and to deliver a converted exponent decimal value representing a fraction part exponent to a hardware-implemented fraction part exponent look-up table/ROM and interpolation logic;

wherein the integer part exponent look-up table/ROM and interpolation logic is configured to receive from the integer part exponent conversion logic, a decimal value representing the integer part exponent and to deliver an equivalent binary value representing the integer part exponent to a hardware-implemented selection logic;

wherein the fraction part exponent look-up table/ROM and interpolation logic is configured to receive from the fraction part exponent conversion logic, a decimal value representing the fraction part exponent and to deliver an equivalent binary value representing the fraction part exponent to the selection logic;

wherein the selection logic is configured to select the delivered equivalent binary value representing the fraction part exponent when the original human-readable decimal character sequence is fraction-only, or to select the delivered equivalent binary value representing the integer part when the original human-readable decimal character sequence is not fraction-only, wherein the selection logic then delivers a selected equivalent binary value exponent to a hardware-implemented final IEEE 754 formatter logic;

a hardware-implemented integer part greatest weight look-up table/ROM and interpolation logic configured to receive from the integer part exponent conversion logic, a decimal value representing the integer part exponent and to deliver to an integer part quantizer logic, a binary value representing an integer part greatest weight corresponding to a decimal value representing the integer part exponent;

a hardware-implemented fraction part greatest weight look-up table/ROM and interpolation logic configured to receive from the fraction part exponent conversion logic, a decimal value representing the fraction part exponent and to deliver to a fraction part quantizer logic, a binary value representing a fraction part greatest weight corresponding to a decimal value representing the fraction part exponent;

wherein the integer part quantizer logic is configured to receive from the integer part greatest weight look-up table/ROM and interpolation logic, a binary value representing the integer part greatest weight and to receive from the integer part mantissa logic, a delivered binary value representing an equivalent binary value

representing the integer part mantissa and to deliver a quantized/encoded integer part value to a final IEEE 754 formatter logic;

wherein the fraction part quantizer logic is configured to receive from the fraction part greatest weight look-up table/ROM and interpolation logic, a binary value representing the fraction part greatest weight and to receive from the fraction part mantissa logic, a delivered binary value representing an equivalent binary value representing the fraction part mantissa and to deliver a quantized/encoded fraction part value to a final IEEE 754 formatter logic; and

a final IEEE 754 formatter logic that accepts the selected equivalent binary value exponent from the selection logic, the quantized/encoded integer part value from the integer part quantizer logic, and the quantized/encoded fraction part value from the fraction part quantizer logic and outputs an IEEE 754 binary floating-point format final result.

20. The fully pipelined convertToBinaryFromDecimalCharacter hardware operator as recited in claim **19**, further comprising hardware logic enabling the hardware operator to convert human-readable decimal character sequence floating-point representations that also include a token exponent.

21. A fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit configured to convert a human-readable decimal character sequence floating-point representation having an integer part and a fraction part to an IEEE 754-2008 binary floating-point representation every clock cycle, said hardware operator logic circuit comprising:

a hardware universal decimal character sequence format translator configured to:

receive the human-readable decimal character sequence floating-point representation up to IEEE 754-2008 "H=20" in length,

separate, in character format, the integer part and the fraction part of the human-readable decimal character sequence floating-point representation,

place the integer part in a first assigned character position of a predetermined character sequence format, and

place the fraction part in a second assigned character position of the predetermined character sequence format;

computational hardware logic configured to convert the integer part and the fraction part to weighted binary values for the integer part and the fraction part; and

a hardware final formatter configured to receive the weighted binary values for the integer part and the fraction part and to format the weighted binary values for a significand part of the IEEE 754-2008 binary floating-point representation.

22. The fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit as recited in claim **21**, wherein the hardware final formatter is configured to selectively output a binary**16**, binary**32**, or binary**64** IEEE 754-2008 binary floating-point representation depending on a Size input.

23. The fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit as recited in claim **21**, wherein the human-readable decimal character sequence floating-point representation also includes an exponent part, and the hardware universal decimal character sequence format translator is further configured to separate the exponent part and place the exponent part in a third assigned character position of the predetermined character sequence format.

24. The fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit as recited in claim **23**, wherein the computational hardware logic is further configured to:

convert the exponent part to an exponent part binary value;

adjust the exponent part binary value for use as an index; and

utilize the index to look up weights for the integer part and the fraction part.

25. The fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit as recited in claim **21**, wherein the human-readable decimal character sequence floating-point representation has no explicit exponent part, and the hardware universal decimal character sequence format translator is further configured to create a character sequence exponent for the human-readable decimal character sequence floating-point representation and place the character sequence exponent in a third assigned character position of the predetermined character sequence format.

26. The fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit as recited in claim **21**, wherein the human-readable decimal character sequence floating-point representation also includes a token exponent part, and the hardware universal decimal character sequence format translator is further configured to create a character sequence exponent for the human-readable decimal character sequence floating-point representation and place the character sequence exponent in a third assigned character position of the predetermined character sequence format.

27. A hardware circuit, comprising:

a first memory-mapped, fully pipelined, hardware operator that, with a first single instruction, is configured to convert one or more received human readable decimal character sequence floating-point representations up to IEEE 754-2008 "H=20" in length into resulting IEEE 754 binary floating-point representations and to automatically store the resulting IEEE 754 binary floating-point representations along with any IEEE 754 exceptional signals produced by the first memory-mapped, fully pipelined hardware operator during conversion;

wherein the resulting IEEE 754 binary floating-point representations and IEEE 754 exceptional signals, if any, are stored in a first randomly accessible result buffer dedicated to the first memory-mapped, fully pipelined, hardware operator at a direct or indirect destination address specified in the first single instruction; and

wherein the first memory-mapped, fully pipelined hardware operator is configured to accept new human-readable decimal character sequence floating-point representations every clock cycle until the first dedicated result buffer becomes full.

28. The hardware circuit as recited in claim **27**, further comprising:

a second memory-mapped, fully pipelined, hardware operator that, with a second single instruction, is configured to convert one or more received IEEE 754 binary floating-point representations into resulting human readable decimal character sequence floating-

point representations and to automatically store the resulting human-readable decimal character sequence floating point representations along with any IEEE 754 exceptional signals produced by the second memory-mapped, fully pipelined hardware operator during conversion;

wherein the resulting human-readable decimal character sequence floating point representations and IEEE 754 exceptional signals, if any, are stored in a second randomly accessible result buffer dedicated to the second memory-mapped, fully pipelined, hardware operator at a direct or indirect destination address specified in the second single instruction; and

wherein the second memory-mapped, fully pipelined hardware operator is configured to accept new IEEE 754 binary floating-point representations every clock cycle until the second dedicated result buffer becomes full.

* * * * *

AMENDMENTS TO THE SPECIFICATION

Please replace the title with the following:

FULLY PIPELINED BINARY CONVERSION
HARDWARE OPERATOR LOGIC CIRCUIT

Please replace paragraph [0001] on page 1 with the following replacement paragraph:

[0001] This application is a divisional of co-pending U.S. Patent Application No. 16/943,077 filed July 30, 2020, which claims the priority benefit under 35 U.S.C. §119(e) of U.S. Provisional Application No. 62/886,570 filed on August 14, 2019, the ~~disclosure~~ disclosures of which [[is]] are incorporated herein by reference in [[its]] their entirety.

Please replace the abstract with the following rewritten new abstract.

ABSTRACT OF THE DISCLOSURE:

A fully pipelined convertToBinaryFromDecimalCharacter hardware operator logic circuit configured to convert one or more human-readable decimal character sequence floating-point representations to IEEE 754-2008 binary floating-point representations every clock cycle. The circuit converts decimal character sequence floating-point representations up to 28 decimal digits in length to IEEE 754 binary64, binary32, or binary16 formats.