Patentability Search Report

Simple and Linear Fast Adder

Index

Key Features

The following key features are prepared based on the disclosure/information provided by the client.

Key-Features		Strength of the key-feature
	The invention discloses a general-purpose fast adder in the form of a sequential logic circuit comprising	
KF 1	A fast adder with linear complexity and area is presented in the form of a sequential logic circuit based on a novel mathematical formulation of addition of natural and real numbers in terms of a finite state machine.	**High**
KF 2	The sequential circuitry is linear in which the adding bits to the inputs does not complicate the circuit topology as the growth is linear and the instruction set is constant, and hardware based.	**High**
KF 3	The fast adder describes an architecture of the Arithmetic Logic Unit (ALU) using comparable material resources to carry-lookahead adder (CLA) and providing less power consumption and potentially faster performance.	**High**
KF 4	The ALU architecture supports operands for integer and rational approximations to real numbers and the operations include left/right shift, multiplication/division by addition, signed operations, and other operations.	**Moderate**
KF 5	A level triggered and dual edge flip flop architecture are reasonable adaptations.	**High**
KF 6	Modifications based on choices of clock cycles, flip flop, and super architectures such as carry save adder constitute versions of the simple and linear fast adder.	**Moderate**

Key-Feature Matrix (Prior-Arts v/s Key-Features)

1. 11 Relevant Patent Citations

Sr. No.	Patent Number	KF 1	KF 2	KF 3	KF 4	KF 5	KF 6
1	**US9928211B2**	Y*	N	Y	Y	N	N
2	**US9292474B1**	Y*	N	Y	Y	N	N
3	**US8724001B2**	Y	N	Y	Y	Y*	N
4	**US20200272417A1**	Y	N	Y	Y	N	Y
5	**US20160028350A1**	Y	N	N	N	Y*	N
6	**US10776690B2**	Y*	N	Y	Y	N	N
7	**CN111030687A**	Y	N	Y	Y	N	N
8	**US20170322772A1**	Y	N	Y	Y	N	N
9	**US6957245B2**	Y	N	Y	Y	N	N
10	**US20190271959A1**	Y	N	N	N	N	N
11	**CN112600774A**	Y	N	N	N	N	N

1. Relevant Patent Citations Details

Reference 1: US9928211B2

Patent Citation Number	US9928211B2
Title	Parallel Self-Timed Adder (PASTA)
Assignee	Universiti Malaya
Publication Date ***(YYYY-MM-DD)***	2018-03-27

Mapping/Relevant Text

A parallel self-timed adder (PASTA) is disclosed. It is based on recursive formulation and uses only half adders for performing multi-bit binary addition. Theoretically the operation is parallel for those bits that do not need any carry chain propagation. Thus the new approach attains logarithmic performance without any special speed-up circuitry or look-ahead schema. The corresponding CMOS implementation of the design along with completion detection unit is also presented. The design is regular and does not have any practical limitations of fan-ins or fan-outs or complex interconnections. **Thus it is more suitable for adoption in fast adder implementation in high-performance processors. The performance of the implementation is tested using SPICE circuit simulation tool by linear technology. Simulation results show its superiority over cascaded circuit adders. A constant time carry propagation is also achieved using the proposed implementation by tuning the CMOS parameters.**

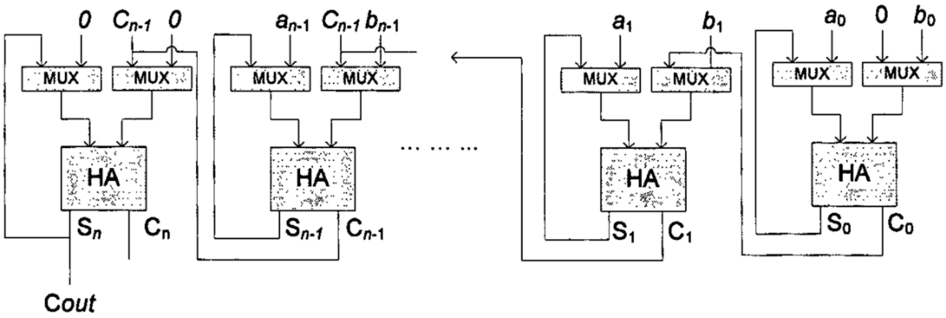
Briefly, the embodiment of this invention is to provide a recursive formulation for Parallel Self-Timed Adder (PASTA). **The design of PASTA is regular and uses Half Adders along with multiplexers with minimum interconnection requirement. Thus the interconnection and area requirement is linear making it easy to fabricate in a VLSI chip.** The design works in truly parallel manner for the number of bits that do not require carry propagation.

A first embodiment of the parallel self-timed adder is presented in FIG. 1. **Multi-bit adders are often constructed from single bit adders using combinatorial and sequential circuits for asynchronous or synchronous design.** The sequential circuits are often serial/chain adders and are not the match for high speed combinatorial adders.

The results clearly indicate the potential of the new PASTA as it performs best among the cascaded logic

designs. It is due to the truly parallel theoretical basis of the design for independent carry chains.

However, the biggest advantage that could be reaped out of the proposed design could possibly be **a truly constant time parallel adder. It is found that the cascading delay for successive carry propagation can be totally avoided by tuning the MOS dimensions.** The timing diagram for single carry propagation for a 32 bit adder circuit for operands A=(FFFF FFFF)16 and B=(1)16 is shown in FIG. 3(d). The completion signal (TERM) and the carry out bit (S32) do not require little more than a single bit delay.



Reference 2: US9292474B1

Patent Citation Number	US9292474B1
Title	Configurable Hybrid Adder Circuitry
Assignee	Altera Corporation
Publication Date ***(YYYY-MM-DD)***	2016-03-22

Mapping/Relevant Text

Particularly when performing computations on wide data words, carry look-ahead adders may be faster than ripple carry adders and carry select adders. However, conventional carry look-ahead adders operate on data words of fixed widths and are not configurable to handle input data words of different desired widths as is sometimes required on programmable integrated circuits.

Carry select adders perform addition more rapidly than ripple carry adders. An 8-bit carry select adder is shown in FIG. 6. As shown in FIG. 6, carry select adder 600 receives inputs A and B and produces corresponding sum and carry out signals. Carry select adder 600 has three adders (A1, A2 and A3) and has a multiplexer MUX. Adders A2 and A1 are used to compute two different versions of the carry and sum signals for the adder, one based on a fixed carry input signal “0” and one based on a fixed carry input signal “1” received on inputs 601. Multiplexer MUX receives corresponding precomputed sum signals on lines 603. Logic gates 605 receive precomputed carry out signals from adders A1 and A2 and provide carry out signal C8. The carry select adder may receive a carry-in signal C0 from a preceding adder. This carry in signal C0 is used in producing C4 at the control input of multiplexer MUX and is used in selecting which of the sum signals at the input to multiplexer MUX should be used as the sum output of adder 60. Because the sum signals at the inputs to multiplexer MUX are precomputed in parallel by adders A2 and A1, addition operations can be performed more rapidly than with ripple carry adders.

Configurable hybrid adders in accordance with embodiments of the present invention may include adder components of different types (e.g., ripple carry, carry select, carry look-ahead, etc.). These adders may perform addition more rapidly than other configurable adder arrangements (e.g., configurable adders based on combinable ripple carry adder blocks), particularly when processing wide inputs (e.g., inputs with bit widths of about 40 bits or more as an example).

Programmable logic 18 may include combinational and sequential logic circuitry. The programmable logic 18 may be configured to perform a custom logic function. The programmable interconnects 16 may be considered to be a type of programmable logic 18.

As the FIG. 10 example demonstrates, it is not necessary to provide two multiplexers on the output of each logic element. **Rather, one of the two multiplexers in each ALE of FIG. 9 (i.e., multiplexer 901B) can be eliminated, leaving only a single sum multiplexer 901 in each ALE. By eliminating the carry multiplexer from each ALE, resource consumption is reduced. The carry signal in this type of configuration can be handled by the last ALE in the chain. This last ALE is configured (by suitable programmable logic programming of look-up table logic connected to the ALE inputs as an example) so that signals A and B on its inputs are both a logic low ("0").**

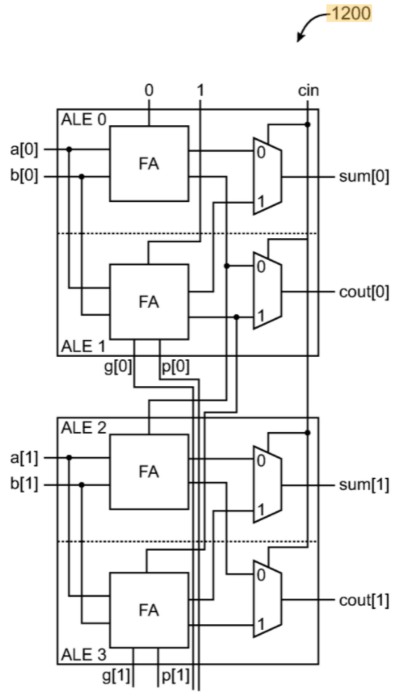


FIG. 12

Reference 3: US8724001B2

Patent Citation Number	US8724001B2
Title	Analog-to-Digital Converter Using a Ramp Generator
Assignee	University of Idaho
Publication Date ***(YYYY-MM-DD)***	2014-05-13

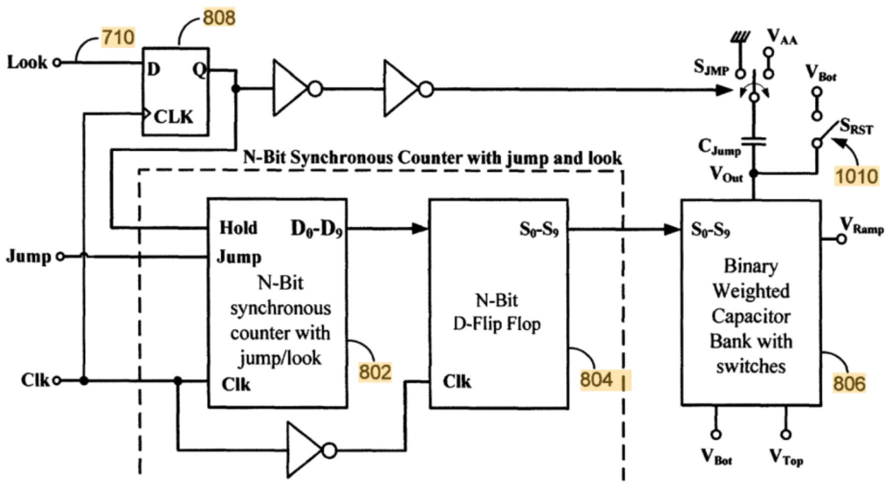
Mapping/Relevant Text
Four control signals determine the finite state machine's (FSM) states: operation reset ("rst"), jump signal from event detector ("jump"), internal counter done signal ("done"), and master clock signal ("mclk"). If the rst=1, state machine stays at S0 state. FSM changes its state at rising edge of the master clock signal conditionally or unconditionally. Unconditional state changes only exist from states S4 to S5 and from S2 to S3. Other state transitions depend on value of the jump, done and reset signals. If the reset is set high, the state machine goes to state S0 and waits until the reset signal is cleared to move to state S1. Done signal is generated in the CONT unit by a 7-bit synchronous counter and comparator units, as shown in the Block diagram of the SSLAR ADC controller unit (CONT) (FIG. 18). It composes of CONT FSM, 7-bit synchronous counter, and 7-bit digital comparator. Block diagram of the ramp-count generator unit is shown in FIG. 21. It generates the analog ramp signal and the associated 8-bit digital counter words. **The unit includes two multiplexers, one continuous time digital carry-look ahead (CLA) full-adder and latch (FAL) block, two continuous time carry-look ahead digital subtractors and one 8-bit binary weighted charge scaling ramp generator blocks.** Look ahead, jump and fall back operations are controlled through the proper timing of the blocks without having physical counter units in the RCG unit. Only clocked unit is the CLA-FAL unit which allow programmable look-ahead, jump or fall back operations. A block diagram CLA full-adder and latch (CLA-FAL) unit is shown in the FIG. 22. It is a part of look-ahead and jump operation. Latch clock (Lclk) is generated and feed to the unit from the CONT unit. 8-bit inputs (A[7:0]) to CLA full-adder are provided by the 4:1 multiplexer unit. Other inputs (B[7:0]) come from the 8-bit latch outputs. CLA full adder adds these two inputs and generates the CLA-FAL block outputs. **Output of the CLA full-adder block is also feed to the 8-bit latch unit. Rising edge triggered D-type flip-flops are used in the latch unit. When Lclk signal is asserted from low to high, latch block holds the 8-bit outputs of the CLA-FAL**

unit.

The CLA type full adder was used due to the fact that other digital adders, such as the ripple carry adders, produce non ideal transition of the output values. In ripple carry adder, adder output for the LSB bit comes first, while the MSB bit becomes available later after certain delay time. This nonlinear delay between adder output bits cause glitches in binary ramp generator (BRG) block. It is because both adder and the binary ramp generator work in continuous time domain. Any delay among adder outputs is reflected at the output of the ramp generator.

2-to-1 multiplexer passes the half step programming word if C2='1' or the zero ('0') to the subtractor #2. It implements the half and full step counter increment operations of the SSLAR ADC algorithm. Half of the step code word is attained by taking upper 6 bit of the original step program word as 2-to-1 multiplexer inputs. C2 signal is generated by the CONT unit. CONT FSM was designed such a way that half step word is not subtracted from the first subtraction unit outputs (W4) during the first look-ahead operation at which CAL-FAL outputs (W1) equal to "00000001" and first subtractor output (W4) is "00000000".

FIG. 8



Reference 4: US20200272417A1

Patent Citation Number	US20200272417A1
Title	Apparatus and Method of Fast Floating-Point Adder Tree for Neural Networks
Assignee	Dinoplusai Holdings Ltd
Publication Date ***(YYYY-MM-DD)***	2020-08-27

Mapping/Relevant Text
A computing device to implement fast floating-point adder tree for the neural network applications is disclosed. The fast float-point adder tree comprises a data preparation module, a fast fixed-point Carry-Save Adder (CSA) tree, and a normalization module. The floating-point input data comprises a sign bit, exponent part and fraction part. The data preparation module aligns the fraction part of the input data and prepares the input data for subsequent processing. **The fast adder uses a signed fixed-point CSA tree to quickly add a large number of fixed-point data into 2 output values and then uses a normal adder to add the 2 output values into one output value. The fast adder uses for a large number of operands is based on multiple levels of fast adders for a small number of operands. The output from the signed fixed-point Carry-Save Adder tree is converted to a selected floating-point format.** **FIG. 3 illustrates an example of traditional multiplication-addition solution. An array of inputs is multiplied with corresponding weights using multiple multipliers (310). Each operand of the multiplier (310) corresponds to an fp16 number.** The output is represented in the fp32 format. An adder tree based on 2-to-1 adders is used to add up the multiplication results. The adder tree comprises multiple layers. The pipeline delay needs many registers (e.g. flip-flops) to buffer the intermediate values, which results in larger circuit size and higher power consumption. **In order to overcome the high cost and high power consumption associated with the conventional floating-point adder tree, the present invention introduces a floating-point adder tree solution that requires much smaller circuits and runs much faster. Furthermore, the adder tree according to the present invention also consumes much less power.** **Fixed point Carry-Save Adder (CSA) technique is well-known in the field of computing architecture and device for its high speed characteristics. The Carry-Save Adder (CSA) technique can reduce the delay substantially.** The idea is to add multiple numbers together and convert it into 2 numbers corresponding to

carry (C) and sum (S) of the addition result of the multiple operands. The CSA method explicitly calculates the carry and sum without the need for dealing with the carry propagation. The CSA postpones the carry propagation till the stage after the CSA.

As mentioned above, the weighted sum calculation plays an important role in neural networks and deep learning. Accordingly, the present invention discloses a high-speed floating-point adder tree capable of handling a large number of operands (e.g. 128, 256 or 512) for the neural network applications. For example, an embodiment according to the present invention can add up 128 operands in the fp32 format into two results in about one clock cycle, where each fp32 operand corresponds to the multiplication result of an input signal and a weight in the fp16 format.

Reference 5: US20160028350A1

Patent Citation Number	US20160028350A1
Title	Modified Flying Adder Architecture
Assignee	BAE Systems Information and Electronic Systems Integration Inc
Publication Date *(YYYY-MM-DD)*	2016-01-28

Mapping/Relevant Text

Embodiments of the present disclosure provide a system for a flying adder circuit. Briefly described, in architecture, one embodiment of the system, among others, can be implemented as follows. **The flying adder circuit includes a fine pulse clock and a coarse pulse clock. A rising edge triggered output circuit is connected to both the fine pulse clock and the coarse pulse clock to provide a pulse train. An adder is provided. A register/accumulator is situated to receive a signal from said adder and said pulse train. A single bit register is situated to receive a signal from said rising edge triggered output.**

In order to solve the phase jitter problem, the modified flying adder architecture may utilize a triggered fine/coarse clock output involving clock edge selection partitioning in order to achieve jitter free synthesis. The clock edge selection includes a reference clock for establishing the leading or rising edge of a coarse pulse coupled to the enable input of a rising edge triggered one-shot circuit. A fine pulse clock is coupled to the trigger input of this one shot circuit. **The rising edge of the fine clock pulse is enabled by the rising edge of the coarse pulse clock that triggers through the fine clock pulse such that the rising edge of a pulse from the fine clock is time coincident with the rising edge of the coarse pulse clock pulse, thus to eliminate phase jitter**. The result is that there does not need to be any averaging for the frequency synthesis. This type of system may allow for all of the necessary clock phases to effectively be selectable from a reduced set of reference phases and results in all equal output periods of the exact desired frequency.

In one embodiment, the frequency that the coarse pulse clock multiplexer outputs is determined by the most significant bits from the register. The frequency that the fine pulse clock multiplexer outputs is determined by the middle bits of the register and the adjustable delay is determined by the least significant bits from the register. **It is noted that not only is the frequency jitter problem solved, in that one can provide phase coherent pulse trains, but the improved flying adder architecture can be utilized in a low-energy integrated**

circuit.

It will be appreciated that multiplexer 12 has an output signal 26, with the register contents applied over line 28. **The output of the frequency synthesizer is available from a D flip-flop 30 which is a one bit register having as an input the output of multiplexer 12. The output of D flip-flop 30 is illustrated by reference character 32 and is a square wave of the desired frequency.** As pointed out, the phase coherency of this output is in question which makes its use for phase concurrent frequency synthesis problematical.

The middle bits of the register 40 (acc(7:4)) represent a fine clock selection. In the illustrated embodiment, the clock pulse trains feeding into the multiplexer 46 have a 256 ps period (3.90625 GHz) and 16 ps edge to edge spacing. The coarse and fine clocks are combined with logic that is enabled when the selected coarse clock is high and triggered on the rising edge of the selected fine clock. **This circuit is a rising edge triggered one-shot circuit 48 that is enabled by the output of multiplexer 42 over line 50, with this circuit being triggered by a pulse from multiplexer 46 as illustrated at 52. The output of the rising edge triggered one-shot circuit is the equivalent of a traditional 12 b flying adder with the 4 LSBs truncated.**

Reference 6: US10776690B2

Patent Citation Number	US10776690B2
Title	Neural Network Unit with Plurality of Selectable Output Functions
Assignee	VIA Alliance Semiconductor Co Ltd
Publication Date ***(YYYY-MM-DD)***	2020-09-15

Mapping/Relevant Text

A neural network unit includes a register programmable with a control value, a plurality of neural processing units (NPU), and a plurality of activation function units (AFU). **Each NPU includes an arithmetic logic unit (ALU) that performs arithmetic and logical operations on a sequence of operands to generate a sequence of results and an accumulator into which the ALU accumulates the sequence of results as an accumulated value.** Each AFU includes a first module that performs a first function on the accumulated value to generate a first output, a second module that performs a second function on the accumulated value to generate a second output, the first function is distinct from the second function, and a multiplexer that receives the first and second outputs and selects one of the two outputs based on the control value programmed into the register.

In one embodiment, **the NPUs 126 and sequencer 128 comprise combinatorial logic, sequential logic, state machines, or a combination thereof.** An architectural instruction (e.g., MFNN instruction 1500) loads the contents of the status register 127 into one of the GPR 116 to determine the status of the NNU 121, e.g., that the NNU 121 has completed a command or completed a program the NNU 121 was running from the program memory 129, or that the NNU 121 is free to receive a new command or start a new NNU program.

However, rather than performing all the multiplies associated with all the connection inputs and then adding all the products together as in a conventional manner, **advantageously each neuron is configured to perform, in a given clock cycle, the weight multiply operation associated with one of the connection inputs and then add (accumulate) the product with the accumulated value of the products associated with connection inputs processed in previous clock cycles up to that point.** Assuming there are M connections to the neuron, after all M products have been accumulated (which takes approximately M clock cycles), the neuron performs the activation function on the accumulated value to generate the output, or result. **This has the advantage of requiring fewer multipliers and a smaller, simpler and faster adder circuit (e.g., a 2-input adder) in the neuron than an adder that would be required to add all, or even a subset of, the products associated with**

all the connection inputs. This, in turn, has the advantage of facilitating a very large number (N) of neurons (NPUs 126) in the NNU 121 so that after approximately M clock cycles, the NNU 121 has generated the output for all of the large number (N) of neurons. Finally, the NNU 121 constructed of such neurons has the advantage of efficiently performing as an artificial neural network layer for a large number of different connection inputs. That is, as M increases or decreases for different layers, the number of clock cycles required to generate the neuron outputs correspondingly increases or decreases, and the resources (e.g., multipliers and accumulators) are fully utilized; whereas, in a more conventional design, some of the multipliers and a portion of the adder may not be utilized for smaller values of M. **Thus, the embodiments described herein have the benefit of flexibility and efficiency with respect to the number of connection inputs to the neurons of the NNU 121, and provide extremely high performance.**

As discussed above, advantageously the NNU 121 operates on integer data rather than floating-point data. This has the advantage of simplifying each NPU 126, or at least the ALU 204 portion. For example, the ALU 204 need not include adders that would be needed in a floating-point implementation to add the exponents of the multiplicands for the multiplier 242. Similarly, the ALU 204 need not include shifters that would be needed in a floating-point implementation to align binary points of the addends for the adder 234. As one skilled in the art will appreciate, floating point units are generally very complex; thus, these are only examples of simplifications to the ALU 204, and other simplifications are enjoyed by the instant integer embodiments with hardware fixed-point assist that enable the user to specify the relevant binary points. **The fact that the ALUs 204 are integer units may advantageously result in a smaller (and faster) NPU 126 than a floating-point embodiment, which further advantageously facilitates the incorporation of a large array of NPUs 126 into the NNU 121.** The AFU 212 portion deals with scaling and saturating the accumulator 202 value 217 based on the, preferably user-specified, number of fractional bits desired in the accumulated value and number of fractional bits desired in the output value. **Advantageously, any additional complexity and accompanying increase in size, power consumption and/or time in the fixed-point hardware assist of the AFUs 212 may be amortized by sharing the AFUs 212 among the ALU 204 portions, as described with respect to the embodiment of FIG. 11, for example, since the number of AFUs 1112 may be reduced in a shared embodiment.**

Reference 7: CN111030687A

Patent Citation Number	CN111030687A
Title	All-Digital Phase-Locked Loop Based on Fast Full Adder and Phase-Locked Control Method
Assignee	University of South China
Publication Date *(YYYY-MM-DD)*	2020-04-17

Mapping/Relevant Text

The all-digital phase-locked loop comprises a digital phase discriminator module, a digital loop filter module, a buffer register and a numerically controlled oscillator module. **The digital loop filter module and the numerical control oscillator module respectively comprise a quick full adder, the quick full adder comprises a plurality of carry look ahead adders, the plurality of carry look ahead adders are connected in an internal carry look ahead cascade mode, and the design of each module circuit is completed in a top-down mode by utilizing an electronic design automation technology. The circuit structure of the all-digital phase-locked loop is optimized by adopting a method based on a fast full adder, so that the phase-locking range of the all-digital phase-locked loop circuit is expanded, and the phase-locking frequency is improved. The invention has the advantages of high phase locking speed, wide locking frequency range, low power consumption and the like, and has important significance for improving the performance of electronic devices and developing semiconductor process technology by being embedded into different system chips as a phase locking circuit module.**

The carry generation signal g and the carry propagation signal p of the four-bit carry look-ahead adder of the first carry look-ahead adder 2-1-1 are both input to the four-bit carry-parallel part CLA, the carry input signals of the first four-bit carry look-ahead adder and the four-bit carry-parallel part CLA are both the carry input signals of the first carry look-ahead adder 2-1-1, and the four-bit carry-parallel part CLA generates the carry input signals of the other 3 four-bit carry look-ahead adders and the carry output signal Cout4 of the first carry look-ahead adder 2-1-1.

The invention can lock the input signals of different frequencies by changing the parameters of the phase-locked frequency control word, expands the phase-locked range of the all-digital phase-locked loop circuit, **obtains the output signal with the same frequency and frequency multiplication as the input signal,** expands the application range of the phase-locked loop and does not increase the logic resource in the FPGA chip.

The numerically controlled oscillator module comprises a third fast full adder and a third register; the third fast full adder comprises a third carry look ahead adder and a fourth carry look ahead adder, the third carry look ahead adder is composed of 4 four-bit carry look ahead adders and 1 four-bit parallel carry part CLA, the 4 four-bit carry look ahead adders are all connected with the four-bit parallel carry part CLA, the fourth carry look ahead adder is composed of 3 four-bit carry look ahead adders and 1 three-bit parallel carry part CLA, the 3 four-bit carry look ahead adders are all connected with the 1 three-bit parallel carry part CLA, and the third carry look ahead adder is connected with the fourth carry look ahead adder.

The digital phase discriminator module, the digital loop filter module, **the buffer register and the digital controlled oscillator module are sequentially connected,** and the output of the digital controlled oscillator module is fed back to the digital phase discriminator module to be used as the input of the digital phase discriminator module.

Reference 8: US20170322772A1

Patent Citation Number	US20170322772A1
Title	Method for A Stage Optimized High Speed Adder
Assignee	Intel Corporation
Publication Date ***(YYYY-MM-DD)***	2017-11-09

Mapping/Relevant Text
A method for fast parallel adder processing. the method includes receiving parallel inputs from a communications path, wherein each input comprises one bit, adding the inputs using a parallel structure, wherein the parallel structure is optimized to accelerate the addition by utilizing a characteristic that the inputs are one bit each, and transmitting the resulting outputs to a subsequent stage. **To find the new positions of the remapping, the outputs of these logical functions are used to feed a parallel adder , and this parallel adder produces the final remapping of the virtual requests into the new positions.** In this manner, the circuit performs the two actions described above where two or more of those inputs (e.g., virtual requests) can be combined, merged or grouped to form a merged request. The other action can be splitting or fragmenting the virtual request or packet into two or more requests. FIG. 17 shows a diagram of a circuit performing an evaluation of the cumulative sum in accordance with one embodiment of the present invention. FIG. 17 shows how the terms of an evaluation equation are processed by hardware components of evaluation circuit. **This diagram also shows a stage optimized high-speed adder circuit (e.g., a parallel adder 1700) that is used to solve the module to function depicted in a single clock cycle. The circuit is particularly suited to any application where a plurality of one bit inputs need to be quickly added to produce a one or two bit output. Details of the parallel adder 1700 are further described in FIG. 21 below.** **The first observation is that when unrolling the recursive sum equation as shown in the figure, the duplicate sum elements will zero out under modulo 2 arithmetic.** The second observation is how addition elements and multiplication elements within the recursive sum equation shown in the figure behave under modulo 2 arithmetic. **Under such conditions, addition elements become XOR functions and multiplication elements become AND functions. This allows the recursive sum equation to be mapped to logical circuits as shown in**

the figure. Elements within the parentheses are multiplication and are thus operated on by the AND gates. Elements outside of the parentheses are additions and are thus operated on by the XOR gates. The equation now becomes completely unrolled in space instead of being serialized.

FIG. 20 shows a diagram depicting a parallel carry save adders in accordance with one embodiment of the present invention. **As shown in FIG. 20, traditional parallel 4:2 carry save adders are used to perform summation of parallel inputs, such as ready bits in a processor scheduling used to choose instructions for dispatch, or to map instructions selected for dispatch to a corresponding dispatch port, or for example, being used to sum valid bits for counting valid sparse entries and assigning allocation write ports to them.**

Reference 9: US6957245B2

Patent Citation Number	US6957245B2
Title	Ultra-Fast Adder
Assignee	Oracle America Inc
Publication Date ***(YYYY-MM-DD)***	2005-10-18

Mapping/Relevant Text

A method and apparatus are disclosed that increase the speed of carry look-ahead (CLA) adders by reducing the stack height of their first stage logic circuits. In accordance with the present invention, a CLA adder capable of adding (or subtracting) two input signals includes first stage logic having a plurality of carry-create and carry-transmit logic circuits each coupled to receive one or more bits of each input signal. Each carry-create circuit generates a novel carry-create signal in response to corresponding first bit-pairings of the input signals, and each carry-transmit circuit generates a novel carry-transmit signal in response to corresponding second bit-pairings of the input signals. The carry-create and carry-transmit signals are combined in CLA logic to generate accumulated carry-create signals, which are then used to select final sum bits.

In addition, adders of the present invention may be readily used to perform arithmetic subtraction operations. Accordingly, the present invention is not to be construed as limited to specific examples described herein but rather includes within its scope all embodiments defined by the appended claims.

For one embodiment, each carry-create circuit is coupled to receive 3 bit-pairings of input signals A and B, and generates a corresponding carry-create signal according to the logical expression $J[z\rightarrow x]=(Az|Bz)+(Ay|By)+(Ax|Bx)$, where $|$ represents the logical AND operation, + indicates the logical OR operation, and x, y, and z represent bit positions in the input signals A and B. The carry-create circuit implements three 2-input AND terms, and thus has a stack height of two. Each carry-transmit circuit is coupled to receive 3 bit-pairings of input signals A and B, and generates a corresponding carry-transmit signal according to the logical expression $T[z\rightarrow x]=(Az+Bz)|[(Ay+By)|(Ax+Bx)+(Ax+Bx)]$. **The carry-transmit circuit implements a 3-input AND term, and thus has a stack height of three. By comparison, prior art carry-propagate and carry-generate circuits have stack heights of three and four, respectively. Thus, because the first stage carry-create and carry-transmit circuits of the present invention have lower stack heights than do**

prior art first stage carry-propagate and carry-generate circuits, adders that incorporate Applicant's first stage carry-create and carry-transmit circuits are faster than prior art adders that utilize conventional carry-propagate and carry-generate circuits.

As discussed above, adder 600 may be faster than conventional CLA adders such as, for example, adder 100 of FIG. 1, because the first stage logic circuits (e.g., J3 circuit 700 and T3 circuit 800) of adder 600 have lower stack heights than first stage logic circuits (e.g., P3 circuit 200 and G3 circuit 300) of conventional CLA adders. Referring again to FIGS. 7 and 8, each discharge path of J3 circuit 700 includes two stacked input transistors coupled between nodes N1 and N2, and thus J3 circuit 700 has a stack height of two, and each discharge path of T3 circuit 800 includes three stacked input transistors coupled between nodes N1 and N2, and thus T3 circuit 800 has a stack height of three. In contrast, prior art P3 circuit 200 of FIG. 2 has a stack height of three, and prior art G3 circuit 300 of FIG. 3 has a stack height of four. Accordingly, because the first stage carry-create and carry-transmit circuits of present embodiments have lower stack heights than prior art first stage carry-propagate and carry-generate circuits, respectively, adders configured in accordance with the present invention may be faster than conventional CLA adders that employ prior art carry-propagate and carry-generate logic circuits.

Reference 10: US20190271959A1

Patent Citation Number	US20190271959A1
Title	Sequential Logic Circuitry with Reduced Dynamic Power Consumption
Assignee	21 Inc
Publication Date ***(YYYY-MM-DD)***	2020-02-11

Mapping/Relevant Text

Digital systems formed on integrated circuits may include sequential logic circuitry. **The sequential logic circuitry may form at least part of a finite state machine that records different logical states. The sequential logic circuitry may include a first latching circuit and a second latching circuit that each latch bits onto their respective outputs when clocked at different levels**. The first latching circuit may output a first bit. Combinational logic circuitry may be distributed on both sides of the first latching circuit such that a combinational logic circuit interposed between the first and second latching circuits generates a second bit based on at least the first bit. The first and second bits may record one of two possible finite logical states of the sequential logic circuitry. **By distributing combinational logic circuity on two sides of a given latching circuit, dynamic power consumption by the sequential logic circuitry may be optimized.**

In this scenario, the combinational logic 18-1 of FIG. 1 may be divided (distributed) on both sides of latch circuit 24-2 to collectively perform the same logical operations as logic 18-1 formed on a single side of flip flop 12-2 in FIG. 1. The components of combinational logic 18-1 may be divided between combinational logic 30-1 and 30-2 such that, if logic 18-1 has a logical depth of N, logic 30-1 and 30-2 each have a logical depth of approximately N/2. In this way, circuitry 22 may encode two different system states (e.g., state “A” and state “B”) using two bits S1’ and S1”, whereas the same states may be stored using a single bit S1 in an arrangement as shown in FIG. 1. By distributing the encoding of the two system states between two bits S1’ and S1”, logic circuity 22 may have reduced logic depth for combinational circuits 30-1 and 30-2 relative to the combinational logic 18-1 of FIG. 1. **Circuitry 22 may thereby require less dynamic power and may incur less dynamic power glitching than circuitry 10 of FIG. 1. By shifting data through circuitry 22 using alternating positive and negative level sensitive latches, circuitry 22 may ensure that the same states as recorded using bit S1 of FIG. 1 are recorded using bits S1’ and S1” of FIG. 2 (while further reducing dynamic power glitching and consumption in the sequential logic circuitry).**

Processing cores 220 may include digital logic circuitry and any desired circuit elements for performing desired processing operations. For example, core circuits 220 may include, but are not limited to, structures such as metal-oxide-semiconductor field-effect transistors (MOSFETs), bipolar junction transistors (BJTs), diode structures, fuses, memory elements, resistors, capacitors, inductors, intellectual property (IP) blocks, **digital logic circuitry such as adders, exclusive OR (XOR) gates, AND gates, and other suitable integrated circuit processing/storage components.**

Reference 11: CN112600774A

Patent Citation Number	CN112600774A
Title	Equalizer of High-Speed Interface Circuit and Control Method Thereof
Assignee	Basaltic Semiconductor Wuhan Co ltd
Publication Date ***(YYYY-MM-DD)***	2021-04-02

Mapping/Relevant Text

The technical scheme adopted by the invention is as follows: **an equalizer of a high-speed interface circuit comprises a linear equalizer module, an adder compensation module, a decision delay module and a coefficient adjustment module, wherein the linear equalizer module is used for receiving and primarily balancing input signals, the linear equalizer module is sequentially connected with the adder compensation module, the decision delay module and the coefficient adjustment module,** the adder compensation module combines input signals after primary balancing of the linear equalizer module with multi-path balanced signals to perform addition processing and then outputs the signals through channel selection, the decision delay module performs decision delay on signals output by the adder compensation module according to control signals and then outputs the signals to the adder compensation module through channel selection, and the coefficient adjustment module performs coefficient weighting adjustment on decision delay signals of the decision delay module and then outputs the decision delay signals to the adder compensation module.

Thus, the linear equalizer module 1 receives the input signal, performs primary equalization, and outputs a first signal, thereby removing a preamble component of inter-symbol interference (ISI) in the input signal and providing a certain high frequency gain. The linear equalizer module 1 adopts a source negative feedback differential amplifier, and the equalizing method is not limited to a resistor-capacitor negative feedback technology, a negative capacitor technology, a low-voltage zero generator and the like.

The invention relates to a control method of an equalizer of a high-speed interface circuit, which receives and primarily balances input signals through a linear equalizer module, combines the input signals after the primary balance of the linear equalizer module with multi-path equalization signals through an adder compensation module to perform addition processing, **then outputs the signals through channel selection, performs decision delay on the signals output by the adder compensation module through a decision delay module according to control signals, outputs the signals to the adder compensation module through the**

channel selection, performs coefficient weighting adjustment on the decision delay signals of the decision delay module through a coefficient adjustment module, and outputs the signals to the adder compensation module, thereby reducing the data jitter of high-speed signals.

BROAD REFERENCES

1. 12 Patent Citations

Sr. No.	Patent Number	Title	Assignee/ Applicant
1.	EP2113836B1	Flexible Adder Circuits with Fast Carry Chain Circuitry	Altera Corporation
2.	CN110210612B	Integrated Circuit Acceleration Method and System Based on Self-Adaptive Piecewise Linear Approximation Curve	Linyi Zhongke Ruihe Intelligent Technology Co.,Ltd.
3.	US10950299B1	System and Method for Cryogenic Hybrid Technology Computing and Memory	SeeQC Inc
4.	US10908879B2	Fast Vector Multiplication and Accumulation Circuit	Neuchips Corporation
5.	CN112783472A	Multi-Value Logic Wide-Bit High-Speed Adder	He Qun
6.	US7444366B2	Faster Shift Value Calculation using Modified Carry-Lookahead Adder	Intel Corporation, Hewlett Packard Development Co LP
7.	US7007059B1	Fast Pipelined Adder/Subtractor Using Increment/Decrement Function with Reduced Register Utilization	Intellectual Ventures II LLC
8.	US20200210146A1	Binary Parallel Adder and Multiplier	Micron Technology Inc
9.	US11165613B2	High-Speed Signaling Systems with Adaptable Pre-Emphasis and Equalization	Rambus Inc
10.	US10768897B2	Arithmetic Logic Unit for Single-Cycle Fusion Operations	International Business Machines Corporation
11.	WO2019113603A1	State Machine Block for High-Level Synthesis	Shadi ASSADIKHOMAMI

12.	US20210150413A1	Data Processing System Configured for Separated Computations for Positive and Negative Data	Mentium Technologies Inc

2. 3 Non-Patent Citation

Sr. No.	Title
1.	Implementation of Regular Linear Carry Select Adder with Binary to Excess-1 Converter
2.	Comparative Analysis of Fast Adder Circuit
3.	Speeding Up Processing with Approximation Circuits

Search Details

Search Keywords

- fast
- speed
- rapid
- quick
- powerful
- efficient
- adder
- flip flop
- arithmetic
- multiplexer
- arithmetic logic unit
- ALU
- logical unit
- arithmetic
- operation
- carry input
- carry lookahead
- floating point
- auxiliary ALU
- hardware
- sequential
- circuit
- topology
- sequential-logic
- circuit
- logic circuitry
- linear
- simple
- straight
- rectilinear
- addition
- division
- subtraction
- multiplication
- input
- left
- right
- cyclic
- d-flipflop
- dual flip flop
- trigger
- level-trigger
- edge-trigger
- architecture
- clock cycle
- linear adder
- real number
- integer
- rational
- approximation
- signed
- operation
- less
- lower
- reduce
- minimum
- power
- performance
- optimize
- improve
- constant
- bit
- edge

Search Strings

Google Patents Search Strings	
1.	((FAST NEAR ADDER) AND LINEAR AND (SEQUENTIAL NEAR LOGIC NEAR CIRCUITRY))
2.	((FAST NEAR ADDER) AND (LEVEL NEAR TRIGGERED) AND (DUAL NEAR EDGE NEAR FLIP NEAR FLOP))
3.	((CARRY NEAR SAVER NEAR ADDER) AND (FLIP NEAR FLOP) AND (CLOCK NEAR CYCLE) AND (FAST NEAR ADDER) AND LINEAR)
4.	((FAST NEAR ADDER) AND (SEQUENTIAL NEAR LOGIC NEAR CIRCUIT) AND (FINITE NEAR STATE NEAR MACHINE))
5.	((FAST NEAR ADDER) AND (SEQUENTIAL NEAR LOGIC NEAR CIRCUIT) AND ((CARRY NEAR LOOKAHEAD NEAR ADDER) OR CLA))
6.	(((LESS OR MINIMUM OR REDUCE OR LOWER) NEAR POWER CONSUMPTION) AND ((SPEED OR RAPID OR EFFICIENT OR FASTER) NEAR PERFORMANCE) AND (FAST NEAR ADDER))
7.	((FAST NEAR ADDER) AND (SEQUENTIAL NEAR LOGIC NEAR CIRCUIT) AND (FINITE NEAR STATE NEAR MACHINE) AND ((CARRY NEAR LOOKAHEAD NEAR ADDER) OR CLA) AND ((ARITHMETIC NEAR LOGIC NEAR UNIT) OR ALU) AND (LEVEL NEAR TRIGGERED) AND (DUAL NEAR EDGE NEAR FLIP NEAR FLOP))
NPL Search Strings	
1.	FAST ADDER AND LINEAR COMPLEXITY AND SEQUENTIAL LOGIC CIRCUIT
2.	LEVEL TRIGGERED AND DUAL EDGE FLIP FLOP ARCHITECTURE AND FAST ADDER
3.	FAST ADDER AND LINEAR AND SEQUENTIAL LOGIC AND (CARRY LOOK-AHEAD ADDER OR CLA) AND (FINITE STATE MACHINE OR FSM)