ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

32-Bit Wave Pipelined Sparse Tree Adder

Lakki Reddy Naresh
M.Tech,Embedded System And VLSI Design.
Department of Electronics,
Mallareddy College of Engineering.

N.Vijayalakshmi
Assistant Professor,
Department of Electronics,
Mallareddy College of Engineering.

Abstract:

In this paper, we discuss the architecture, design, and testing of the first 32-bit asynchronous wave-pipelined sparse-tree superconductor rapid single flux quantum adder implemented. Compared to the Kogge–Stone adder, our parallel-prefix sparse-tree adder has better energy efficiency with significantly reduced complexity (at the expense of latency) and almost no decrease in operation frequency.

INTRODUCTION:

One of the most universal digital circuits for almost any application is an added. It is the fundamental building block of Arithmetic Logic Units (ALUs) in general-purpose and special-purpose digital signal microprocessors. Currently, in the CMOS domain, the design space of adder structures has been nearly exhausted, with only minimal improvements shown over previous designs. In contrast, emerging digital circuit technologies such as superconducting Rapid Single Flux Quantum (RSFQ) logic opens a way for researchers to explore new design methodologies for extremely fast, energy-efficient adders. In RSFQ logic, most adder designs demonstrated to date are bit-serial or digit-serial architectures which operate on a single bit or a small group of bits sequentially at a very high processing rate . Such designs allow for simple clocking and compact structures.

However, the latency of serial adders scales O(n), where n is the number of bits per operand, which leads to long latencies for 32-/64-bit operations in general purpose processors. In the past, parallel architectures in RSFQ have been limited to small data widths or relatively long latency ripple-carry adders. One study evaluated 32-/64-bit parallel Kogge-Stone RSFQ adders using co-flow clocking

In the effort of realizing scalable, high-performance, fully parallel designs, a new technique of asynchronous hybrid wave-pipelining for RSFQ circuits has been developed at Stony Brook University (SBU) . Later, as a result of the collaboration between the SBU and HYPRES designers, an8-bit wave-pipelined ALU was successfully designed, fabricated, and demonstrated correct operation at the rate of 20 GHz[13], [14].In this paper, we present the design of the first 16-bit asynchronous parallel adder implemented in RSFQ logic. It builds upon the proven hybrid wave-pipelining techniques to provide16-bit wide processing and synchronization. It incorporates an energy efficient, low complexity sparse-tree structure with very high processing rate. The work is based on a design study fora scalable 32-bit wave-pipelined sparse-tree adder conducted at SBU.

FORMULATION:

The micro architecture of the adder has two main features asynchronous hybrid wave-pipelined processing and a prefix sparse-tree carry generate-propagate structure for arithmetic.

Hybrid wave-pipelining:

The use of data and clock waves in two-dimensional RSFQ circuits such as multipliers with counter-flow clocking was first proposed in [16]. In data-driven wave-pipelining, data waves “self-propagate” through combinational (non-clocked)logic gates without any need for clock signals. The data waves are followed by reset waves that “clean up” the residual logic states of the gates before the next data wave arrival. The maximum clock rate is limited by (1) the time differences in data propagation paths through combinational circuits, and (2) the minimum time gap between data and reset signals.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

In co-flow clocking, data are "pushed through" the circuits by clock signals, so the computation process involves setup time overhead at each stage. In contrast, wave-pipelined circuits are data-driven, not clock-driven, and as a result, they have no setup time contribution to the latency of operations. In processor design, the most viable approach is to use asynchronous wave-pipelining for units with regular structure, and co-flow clocking for irregular circuits such as control logic.

PREFIX SPARSE-TREE CORE:

High-performance parallel adders typically use prefix trees which generate carries in log 2(n) time, where n is the number of bits of the data-path. The Kogge-Stone adder (KSA) [18] is considered to be the fastest among parallel-prefix adders. AKSA adder was successfully used in the 20 GHz 8-bit RSFQALU [14]. However, KSAs have very high complexity and a tremendous amount of wiring congestion.

Further enhancements to the KSA prefix structure such as the sparse-tree configuration have been proposed and used in high-performance Intel processors [19].In our 16-bit RSFQ adder design, we chose the sparse-tree structure to reduce the number of Josephson junctions (JJs)needed for its implementation without any significant effect on its processing rate. As a side effect, this will also lead to a more energy-efficient design by reducing the total bias current and power consumption.

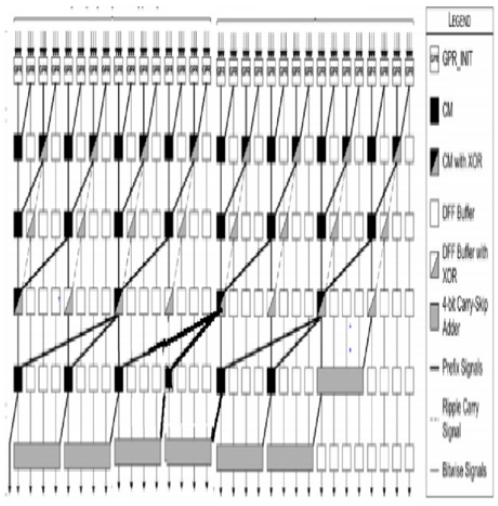

Fig. 1 Structural diagram of the 32-bit sparse-tree adder

It consists of the following three stages: Initialization, Prefix-Tree and Summation. The Initialization stage receives two 16-bit data operands A and B to create bit-wise Generate (G) and Propagate (P) signals which will be merged in a logarithmic manner in the Prefix-Tree stage.

The Prefix-Tree stage consists of Carry-Merge (CM) blocksto merge the prefix signals and provides a group carry to each4-bit summation block. In contrast, the Kogge-Stone prefix-tree provides a carry to every individual bit of the adder.DFF (D flip-flop) buffers appropriately delay prefix and bit-wise P signals until they are ready to be merged or processed at the Summation stage, respectively. The first three levels of the Prefix-Tree also perform the ripple-carry addition within each 4-bit group before data arrive at the Summation stage. The Summation stage computes the final sum with 4-bitcarry-skip adders [20]. The lower-half of the adder (bits 7:0) can start the Summation stage early because all appropriate signals are ready. The upper-half of the adder (bits 15:8) must wait until carries for this upper half are calculated by the very last level of the Prefix-Tree stage.

The Initialization stage consists of GPR_INIT logic blocks, one for each bit. The GPR_INIT creates the bit-wise prefix functions described as Gi = Ai • Bi and Pi = Ai ⍰ Bi where i is the bit index column ranging from 15 down to 0 in the 16-bitadder. These functions are easily realized through clocked AND and XOR gates in a co-flow clocking arrangement. The clock isthe Rdy signal provided to all bits through a distribution tree built with passive transmission lines (PTLs). Additionally, it is necessary to create the trailing reset signal R which will be used to reset the asynchronous elements in the Prefix-Tree. Signal Ris a copy of the Rdy signal for each bit with JJ-based delay lines to ensure data signals are processed before reset follows in the asynchronously wave-pipelined Prefix-Tree. The Prefix-Tree stage is built with CM blocks to merge the prefix signals as shown in Fig. 1.

Merging of the prefix signalsis described in [18]. It is implemented with CFFs (resettable Muller C-flip-flop gates based on the Muller C-element [22],[23]) and confluence buffers used as asynchronous OR gates. The CFFs provide the following functions. First, they behave as asynchronous AND gates.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Second, they are used as keyre-synchronization elements for wave-pipelining allowing data waves to wait until all their appropriate signals arrive. Due to the encoding of the prefix signals, confluence buffers can be safely used as asynchronous OR gates without any danger of violating the time separation requirement of their input pulses. The Summation stage has a 4-bit carry-skip adder block [20]for each 4-bit group. In our carry-skip adders, the generation of a carry-in to the two most significant bits of the group is done in parallel with the calculation of the two least significant sum bits in the group. For each group, the signal Propagate-Propagate (PP) obtained from the Prefix-Tree denotes whether a carry-in will propagate through the lower 2 bits of the group or not. This propagation is done by a clocked AND gate which creates the correct carry-in of the upper 2 bits of the group, thus "skipping" the slower ripple-carry produced from the lower 2 bits.

The summation in the upper and lower halves of the 4-bit groups is done by T1 (toggle flip-flops with clocked output [24]) and clocked XOR gates. Trailing reset signals are used to clock theT1 and XOR gates in the Summation blocks. To facilitate high-speed on-chip testing, we designed 3 supplemental circuits, namely: a clock generator, input shift register, and output compressor. The clock generator uses a 16 pulse-train design where a single clock pulse is sequentially split 16 times and fed back into a ladder structure of confluence buffers. An appropriate number of JTL delays are inserted to achieve a particular range of clock frequencies.

Two such structures are integrated into the clock generator to provide two frequency ranges:15–25.5 GHz, and 23.5–41 GHz. By adjusting the bias voltage of the clock generator, these two ranges cover the full spectrum of clock frequencies that the 16-bit adder is expected to operate at. The clock generator characteristics were obtained through Verilog simulation using gate timing parameters from JJ circuit-level modeling. An additional one-input pulse, one-output pulse low-frequency mode is also available. A 16-bit parallel-load/ parallel-output shift register provides test vectors to the inputs of the adder at high speed. The high-speed outputs of the adder are captured by the output compressor to create an XOR signature of the results using T1gates. Verilog testbench simulations have been written to verifyfunctional correctness and obtain frequency-dependent DC bias margins.

Fig. 2 shows that the 16-bit adder can operate up to38.5 GHz. It has +20%/ – 16% DC bias margins at the target clock rate of 30 GHz.

RESULTS:

Simulation:

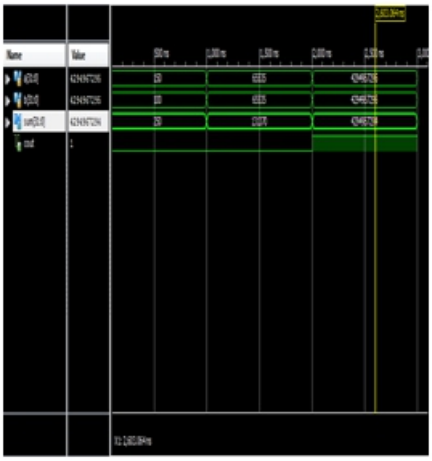

Fig 2 32-bit simulation result

Area report

```
Device utilization summary:
---------------------------

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice LUTs:                   58  out of   9112    0%
    Number used as Logic:                58  out of   9112    0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     58
   Number with an unused Flip Flop:      58  out of     58  100%
   Number with an unused LUT:             0  out of     58    0%
   Number of fully used LUT-FF pairs:     0  out of     58    0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                          97
 Number of bonded IOBs:                  97  out of    232   41%

Specific Feature Utilization:
```

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

Timing report:

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 23.444ns

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 828 / 33
-------------------------------------------------------------------------
Delay:               23.444ns (Levels of Logic = 17)
  Source:            a<1> (PAD)
  Destination:       sum<19> (PAD)

  Data Path: a<1> to sum<19>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
```

RTL diagram:

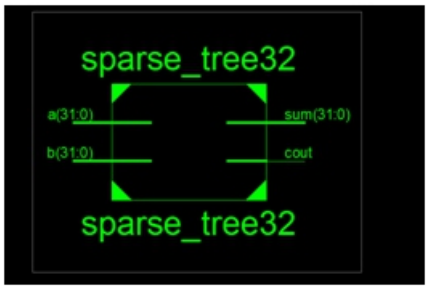

Fig 3 RTL diagram

CONCLUSION:

All test vectors in our test plan successfully produced the expected output for the low frequency testing of the first 32-bit adder chip. In the low frequency mode of operation of the second chip with high-frequency testing circuits, the serial output of the input shift register showed several but not all bits functioning correctly for operand A. We have designed and tested the first 16-bit wave-pipelined sparse-tree adder.

REFERENCES:

[1] M. Tanaka, H. Akaike, A.Fujimaki, Y. Yamanashi, N. Yoshikawa,S. Nagasawa, K. Takagi, and N. Takagi, "100-GHz single-flux-quantumbit-serial adder based on 10-kA/cm2 niobium process," IEEE Trans. Appl.Super-cond., vol. 21, no. 3, pp. 792–796, Jun. 2011.

[2] A. F. Kirichenko and O. A. Mukhanov, "Implementation of novel "push-forward" RSFQ Carry-Save Serial Adders," IEEE Trans. Appl. Super-cond., vol. 5, no. 2, pp. 3010–3013, Jun. 1995.

[3] A. Y. Kidiyarova-Shevchenko, K. Y. Platov, E. M. Tolkacheva, andI. A. Kataeva, "RSFQ asynchronous serial multiplier and spreading codesgenerator for multiuser detector," IEEE Trans. Appl. Supercond., vol. 13,no. 2, pp. 429–432, Jun. 2003.

[4] S. V. Polonsky and A. V. Rylyakov, "RSFQ arithmetic blocks for DSPapplications," IEEE Trans. Appl. Super-cond., vol. 5, no. 2, pp. 2823–2826,Jun. 1995.

[5] H. Park, Y. Yamanashi, N. Yoshikawa, M. Tanaka, and A. Fujimaki,"Design of fast digit-serial adders using SFQ logic circuits," IEICE Electronics Express, vol. 6, no. 19, pp. 1408–1413, 2009.

[6]S.V.Polonsky,V.K.Semenov,P.I.Bunyk,A.F.Kirichenko,A. Y. Kidiyarov-Shevchenko, O. A. Mukhanov, P. N. Shevchenko, D. F. Schneider, D. Y. Zinoviev, and K. K. Likharev, "New RSFQ circuitsJosephson junction digital devices," IEEE Trans. Appl. Supercond.,vol.3,no. 1, pp. 2566–2577, Mar. 1993.

[7] J. Y. Kim, S. Kim, and J. Kang, "Construction of an RSFQ 4-bit ALU withhalf adder cells," IEEE Trans. Appl. Supercond., vol. 15, no. 2, pp. 308–311, Jun. 2005.

[8] Q. P. Herr, N. Vukovic, C. A. Mancini, K. Gaj, V. Adler, E. G. Friedman,A. Krasniewski, M. F. Bocko, and M. J. Feldman, "Design and low speedtesting of a four-bit RSFQ multiplier-accumulator," IEEE Trans. Appl.Super-cond., vol. 7, no. 2, pp. 3168–3171, Jun. 1997.

[9] R. Nakamoto, S. Sakuraba, T. Onomi, S. Sato, and K. Nakajima, "4-bitSFQ Multiplier Based on Booth Encoder," IEEE Trans. Appl. Supercond.,vol. 21, no. 3, pp. 852–855, Jun. 2011.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Monthly Peer Reviewed Open Access International e-Journal

[10] P. Bunyk and P. Litskevitch, "Case study in RSFQ design: Fast pipelinedparallel adder," IEEE Trans. Appl. Supercond., vol. 9, no. 2, pp. 3714–3720, Jun. 1999.

[11] M. Dorojevets, C. Ayala, and A. Kasperek, "Development and evaluationof design techniques for high-performance wave-pipelined wide data path RSFQ processors," in Proc. 12th Int. Supercond.Electron. Conf.,Fukuoka, Japan, 2009, SP-P46.

[12] M. Dorojevets, C. L. Ayala, and A. K.Kasperek, "Data-flow microarchitecture for wide datapath RSFQ processors: Design study," IEEE Trans.Appl. Supercond., vol. 21, no. 3, pp. 787–791, Jun. 2011.

[13] T. Filippov, M. Dorojevets, A. Sahu, A. Kirichenko, C. Ayala, andO. Mukhanov, "8-bit asynchronous wave-pipelined RSFQ arithmetic-logic unit," IEEE Trans. Appl. Supercond., vol. 21, no. 3, pp. 847–851,Jun. 2011.