VLSI

RTL Design and Analysis of an Energy-Efficient 8-Bit Structural ALU Using Source-Synchronous SPI-Driven Clock Gating and Operand Isolation

U. Venkata Surya, S. Aravind, B. Madhan Mohan

Department of Electronics and Communication Engineering, RGUKT RK Valley, Kadapa, Andhra Pradesh, India

Published November 2026

Abstract

Dynamic power caused by continuous clock switching and wide parallel I/O interfaces is a primary constraint in low-power embedded processors, IoT sensor nodes and edge signal-processing systems. Conventional synchronous arithmetic logic units (ALUs) rely on free-running global clocks and parallel operand buses, wasting switching energy during idle periods. Prior parallel structural 8-bit ALU designs on the Xilinx Artix-7 xc7a100tcsg324-1 report 0.595 W of dynamic power, dominated by 0.555 W of I/O dissipation. This paper presents a fully serial, source-synchronous 8-bit structural ALU controlled as an SPI Mode 0 slave using five pins (sclk, reset, cs_n, mosi, miso). Operands, operation select, carry-in and result readback are carried in a single 32-bit SPI frame. Because sclk is inactive while cs_n is high, background clock switching is eliminated. Received bits are assembled in a shift register and transferred to double-buffered operand registers only at packet completion, providing operand isolation. A single-cycle valid flag enables a Xilinx BUFGCE clock buffer so that the gated clock emits exactly one glitch-free pulse per frame. The design was verified by behavioral simulation in Vivado XSim and analyzed with the Vivado Power Analyzer after post-route implementation. The report shows dynamic power below 0.001 W and a total on-chip power of 0.091 W, of which 99% is device static power.

Keywords: Clock gating; Operand isolation; SPI; Structural ALU; Low-power design; Artix-7

1 Introduction

Power dissipation has become a first-order design constraint in battery-powered and energy-harvesting systems such as IoT sensor nodes and edge processors. In CMOS circuits the dynamic component, proportional to switching activity, load capacitance, the square of the supply voltage and clock frequency, usually dominates when logic is active [1]. In FPGA implementations, clock distribution and I/O buffers contribute a large share of this dynamic power.

A conventional synchronous ALU is driven by a continuously running clock and fed through wide parallel operand and result buses. Even when no useful computation is performed, the clock network toggles, and every parallel pin adds I/O switching energy. A previously reported parallel structural 8-bit ALU on the Artix-7 xc7a100tcsg324-1 consumes 0.595 W of dynamic power, of which 0.555 W is I/O power, and it makes no attempt to suppress idle-time clock activity.

This work addresses both sources together. The ALU is wrapped in a serial SPI slave interface that reduces the pin count to five, and the SPI protocol itself is used as the activity indicator for clock

gating: the master-driven sclk is silent whenever the chip-select cs_n is high, and the ALU clock is released for exactly one cycle per completed frame. The main contributions are:

1. a fully serial, five-pin, source-synchronous 8-bit ALU controlled through a single 32-bit SPI Mode 0 transaction;
2. double-buffered operand registers that isolate the ALU inputs from mid-packet shifting and prevent combinational glitching;
3. a BUFGCE-based glitch-free clock-gating scheme that produces one ALU clock pulse per frame; and
4. functional verification and post-implementation power analysis in Vivado for the Artix-7 target.

2 Background

2.1 Clock Gating

Clock gating disables the clock to a register group when its contents need not change, removing the clock-tree and flip-flop switching associated with that group [2]. A naive AND-gate clock gate can produce truncated or spurious pulses if the enable changes while the clock is high. The Xilinx 7-series BUFGCE primitive provides a glitch-free clock enable on a global clock buffer [4], [5], which makes it suitable for FPGA clock gating.

2.2 Operand Isolation

Operand isolation prevents data from propagating into a combinational block while its result is unused. Here, isolation is achieved at the interface: bits arriving serially change only an internal shift register, and the ALU operand registers are updated once, at the end of a frame.

2.3 Serial Peripheral Interface

SPI is a synchronous serial protocol in which the master generates the clock and selects the slave using an active-low chip select [7]. In Mode 0 (CPOL = 0, CPHA = 0), data are sampled on the rising clock edge and the clock idles low. Since the clock is generated only during a transaction, a slave that uses sclk as its working clock inherits an inherently idle-quiet clock.

2.4 Dynamic Power Model

The dynamic power of a CMOS node is commonly written as $P = \alpha \cdot C \cdot V^2 \cdot f$, where α is the switching activity, C the switched capacitance, V the supply voltage and f the clock frequency [1], [3]. Clock gating attacks the product $\alpha \cdot f$ on the clock network and on the registers it feeds, while serializing the interface reduces the number of switching I/O buffers and the capacitance C that each transition drives. The proposed design applies both ideas: the effective clock frequency seen by the ALU is one pulse per frame, and the interface has five pins instead of a wide parallel bus.

2.5 Limitations of Conventional Parallel ALUs

A parallel ALU interface exposes two 8-bit operand buses, a select bus, a carry input and an 8-bit result bus as device pins. Each of these pins requires an I/O buffer, and every transition on them dissipates I/O power, which is why the baseline in this work reports 0.555 W of its 0.595 W dynamic power in the I/O category. In addition, a free-running clock continues to toggle the clock tree and flip-flops during idle intervals, even if the data inputs are static.

3 Proposed architecture

Fig. 1 shows the elaborated schematic of the design. It consists of three blocks: the SPI slave controller (spi_slave_ctrl), the 8-bit structural ALU (alu_8bit), and the clock-gating element (clk_gate) that drives the clock of the ALU output register alu_out_reg [7:0].

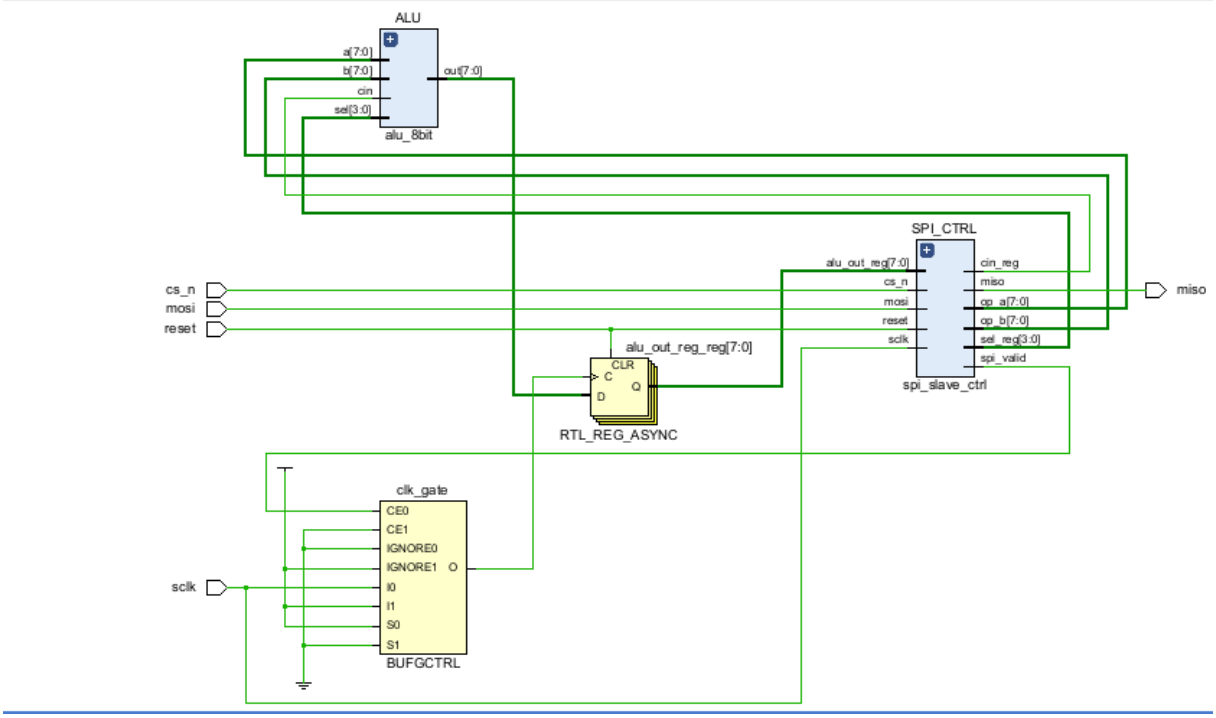


Figure 1. Elaborated schematic: SPI slave controller, 8-bit ALU, gated output register and clock-gate buffer.

3.1 SPI Slave Controller

The controller exposes the five-pin interface sclk, reset, cs_n, mosi and miso. While cs_n is low, incoming mosi bits are shifted into an internal register. Operands, operation-select bits and carry-in are packed together with the serial result readback into one 32-bit frame. When the frame completes, the shift register contents are transferred to the operand registers op_a, op_b, sel_reg and cin_reg, and a single-cycle spi_valid_flag is asserted.

3.2 Structural 8-Bit ALU

The ALU is modeled structurally in Verilog HDL as eight cascaded 1-bit slices built from full adders, discrete logic gates and 16-to-1 multiplexers. The 4-bit select sel_reg chooses the function at the multiplexer of each slice, and alu_cout reports the carry out of the most significant slice.

3.3 Clock Gating

spi_valid_flag drives the enable of a Xilinx BUFGCE global clock buffer, which is mapped by Vivado onto a BUFGCTRL site (Fig. 1). The output gated_clk therefore carries exactly one glitch-free pulse per frame. This pulse updates alu_out_reg [7:0], an asynchronously cleared register, before the result is shifted out on miso. Between frames the gated clock is held inactive, so the output register and its clock branch do not toggle.

3.4 Frame Format and Timing

Every operation is carried out in one 32-bit SPI frame. The master pulls cs_n low, clocks 32 bits on sclk while presenting operands, select bits and carry-in on mosi, and reads the 8-bit result back on miso, and then releases cs_n. At an sclk period of 100 ns the frame lasts 32 × 100 ns = 3.2 µs. After the last bit has been received, the transfer of the shift register contents to the operand registers and the single gated-clock pulse take place, so the operation time seen by the master is determined by the frame length and not by the internal ALU delay.

3.5 Reset and Initialization

The reset input places the controller and the asynchronously cleared output register alu_out_reg [7:0] in a known state before the first frame. As seen in Fig. 2, reset is applied at the start of the simulation, after which all registers are initialized to zero and the first frame can be accepted.

3.6 Operand Isolation Benefit

If the serial bits were applied directly to the ALU inputs, the combinational network would see 32 intermediate input patterns during every frame, and each pattern would cause toggling in the adders and multiplexers. With double-buffered registers, the ALU inputs change exactly once per frame. The result is stable by the time the gated clock pulse arrives, and the output register captures a settled value instead of a transient one.

4 Implementation methodology

The design, verification, synthesis, post-route implementation and timing-constrained vectorless power analysis were all carried out in the Xilinx Vivado Design Suite targeting the Artix-7 xc7a100tcsg324-1. Functional behavior was checked in Vivado XSim with a testbench acting as the SPI master, using an sclk period of 100 ns (10 MHz) and a series of 32-bit frames covering the supported operations. Power was then profiled in the Vivado Power Analyzer [6] to obtain the dynamic, I/O and total on-chip power.

4.1 Tool Flow

The flow followed in Vivado consists of RTL entry, behavioral simulation, synthesis, implementation (placement and routing) and power estimation. The Verilog structural description of the ALU slices, the SPI controller and the BUFGCE instance form the top-level design. Timing constraints were applied before the power estimate so that the clock definition used by the analyzer corresponds to the intended sclk frequency.

4.2 Testbench and Stimulus

The testbench behaves as an SPI master. It applies reset, then issues successive frames with different operand pairs and select codes, toggling cs_n and sclk according to Mode 0 timing. Operand values were chosen so that carry propagation, bitwise logic and inversion functions each produce a distinct and easily checked result, and alu_cout is monitored for carry or borrow behavior.

4.3 Power Analysis Setup

Power was estimated after place and route with timing constraints applied. No switching-activity file was used; the analyzer therefore applies default toggle-rate assumptions to the design, which is the meaning of vectorless analysis [6].

5 Results and discussion

5.1 Functional Verification

Fig. 2 shows the behavioral simulation. The cs_n signal goes low once per frame and sclk pulses only during those windows; outside them it remains idle. The operand and select registers change only after a frame completes, and spi_valid_flag and gated_clk each produce a single narrow pulse per frame. The ALU result alu_result [7:0] is captured into alu_out_reg [7:0] on that pulse, and the output register holds its value until the next frame.

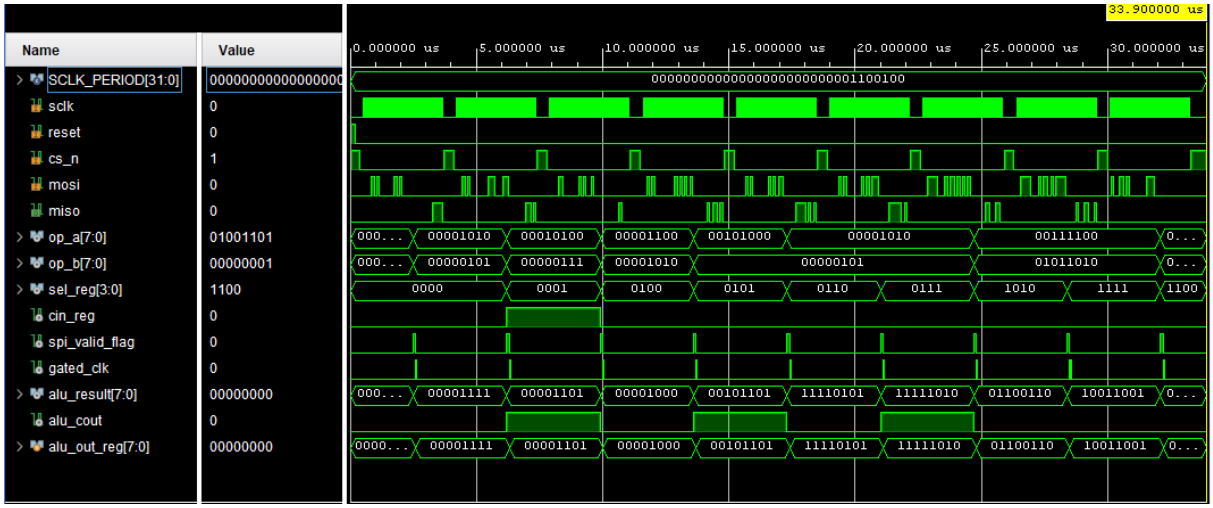


Figure 2. XSim waveform showing frame-based SPI activity, double-buffered operands, and one gated-clock pulse per frame.

Table 1 lists the operations exercised in the simulation. For example, 0x0A + 0x05 returns 0x0F for the add operation, and 0x14 − 0x07 returns 0x0D for the subtract operation. The logic functions agree with their expected bitwise results.

Table 1. Operations observed in simulation.

sel	Function	Operands (A, B)	Result
0000	ADD	0Ah, 05h	0Fh
0001	SUB	14h, 07h	0Dh
0100	AND	0Ch, 0Ah	08h
0101	OR	28h, 05h	2Dh
0110	NOT A	0Ah, 05h	F5h
0111	NOT B	0Ah, 05h	FAh
1010	XOR	3Ch, 5Ah	66h
1111	XNOR	3Ch, 5Ah	99h

5.2 Power Analysis

The post-implementation power report is shown in Fig. 3. The dynamic power of the design is below 0.001 W, which is under 1% of the total, and the device static power of 0.091 W accounts for the remaining 99%. Within the dynamic part, clocks contribute 79%, signals 7%, logic 4% and I/O 10%.

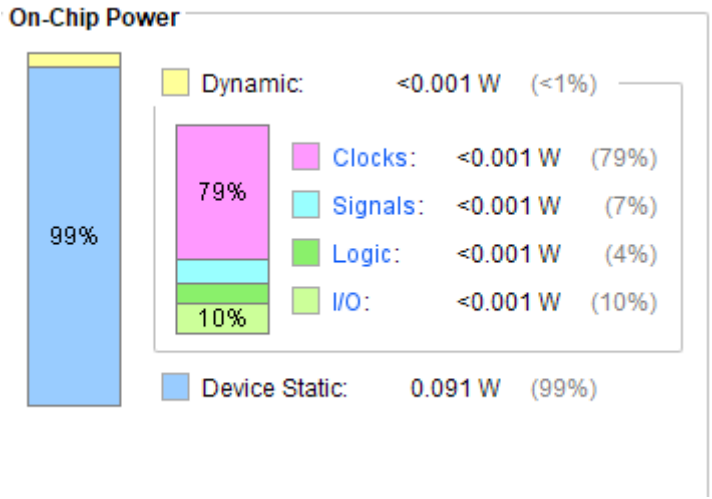


Figure 3. Vivado on-chip power report for the proposed design.

Table 2 compares the proposed design with the parallel structural ALU baseline on the same device. Dynamic power falls from 0.595 W to below 0.001 W, a reduction of more than 99%. The I/O component falls from 0.555 W to below 0.001 W, which is consistent with replacing the wide parallel interface by five pins. The remaining on-chip power is dominated by device static power, which is a property of the FPGA and is not affected by the architecture.

Table 2. Power comparison on xc7a100tcsg324-1.

Metric	Parallel ALU	Proposed
Dynamic power (W)	0.595	< 0.001
I/O power (W)	0.555	< 0.001
Device static (W)	—	0.091
Interface pins	Parallel	5 (SPI)
Idle clock activity	Yes	No (gated)

The trade-off is throughput: each operation needs a full 32-bit serial frame, i.e. 32 sclk cycles (3.2 µs at 10 MHz), compared with a single cycle for a parallel ALU. The architecture is therefore best suited to low-duty-cycle, energy-constrained applications. Also, the power figures come from vectorless analysis, whose toggle-rate assumptions are set by the tool; they indicate relative improvement rather than measured board-level consumption.

5.3 Discussion of the Power Distribution

Clocks account for the largest share (79%) of the small dynamic component, followed by I/O (10%), signals (7%) and logic (4%). The ordering shows that, once the I/O interface is serialized, the clock network is the main remaining dynamic contributor, which is the reason for gating the clock of the

ALU output register. The logic share is small because the combinational ALU changes its inputs only once per frame.

Since the reported dynamic power is below 0.001 W, an upper bound on the dynamic energy of one operation can be obtained by multiplying this value by the 3.2 µs frame time, giving less than 3.2 nJ per operation even if the design were active continuously. This is a bound derived from the tool report, not a measured energy value.

5.4 Suitability for Embedded Applications

The five-pin interface suits microcontrollers, sensor hubs and FPGA-based edge nodes, where SPI is already available and pins are scarce. Because the ALU clock is released only after a complete frame, the block imposes no clock activity on the system between requests, which matters in duty-cycled IoT sensors that spend most of their time idle.

5.5 Limitations

Three limitations should be noted. First, the power numbers are vectorless estimates and should be confirmed by measurement on hardware. Second, device static power (0.091 W) dominates the total and is independent of the architecture, so the benefit of the design is visible mainly in the dynamic component. Third, serial operation reduces throughput compared with a single-cycle parallel ALU.

6 Conclusion

This paper presented an 8-bit structural ALU controlled through a five-pin SPI Mode 0 interface, with double-buffered operand registers and BUFGCE-based clock gating. Simulation confirmed correct operation, with one gated-clock pulse per 32-bit frame. Post-implementation analysis on the Artix-7 xc7a100tcsg324-1 reports dynamic power below 0.001 W and total on-chip power of 0.091 W, compared with 0.595 W of dynamic power for a parallel baseline. Future work includes hardware validation on a development board with a measured power comparison, support for wider operand sizes, and burst-mode frames to improve throughput.

Acknowledgment

The authors thank the Department of Electronics and Communication Engineering, RGUKT RK Valley, for providing the laboratory and software facilities used in this work.

References

[1] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low-power CMOS digital design,” *IEEE J*. Solid-State Circuits, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[2] Q. Wu, M. Pedram, and X. Wu, “Clock-gating and its application to low power design of sequential circuits,” *IEEE Trans*. Circuits Syst. I, Fundam. Theory Appl., vol. 47, no. 3, pp. 415–420, Mar. 2000.

[3] N. H. E. Weste and D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Boston, MA, USA: Addison-Wesley, 2011.

[4] Xilinx, Inc., “7 Series FPGAs Clocking Resources User Guide,” *UG472*.

[5] Xilinx, Inc., “7 Series FPGAs Libraries Guide for HDL Designs,” *UG768*.

[6] Xilinx, Inc., “Vivado Design Suite User Guide: Power Analysis and Optimization,” *UG907*.

[7] Motorola, Inc., “SPI Block Guide,” *S12SPIV3/D*, Document V03.06.