Jsilicon

- Author: JunHyeok Seo (mirseo)
- Description: 8-bit ALU with UART transmission built in Verilog, designed for TinyTapeout shuttle.
- Language: Verilog

JSilicon v0.1 – Minimal 8-bit CPU Core

JSilicon is an **8-bit ALU core** designed and implemented from scratch during my mandatory military service in South Korea (2025).

This project integrates an ALU, FSM (Finite State Machine) - based control logic, and UART output module, demonstrating the feasibility of a fully functional silicon design even under highly constrained development conditions.

The JSilicon core accepts two **4-bit operands** and internally performs **8-bit arithmetic and logic operations**. the final result is generated as 16bits, with the lower 8 bits exposed through the output pins.

This design marks the beginning of the **JSilicon series**, an initiative inspired by the simplicity of JavaScript and the philosophy of accessible silicon design.

Overview

– **ALU (Arithmetic Logic Unit)** - Performs basic arithmetic and logic operation ($+$, -, *, /, %, $==$, $>$, $<$)

– **FSM (Finite State Machine)** - Controls ALU execution and UART transmission sequence

– **UART_TX (Universal asynchronous receiver-transmitter)** - Outputs the computed result over serial, enabling easy connection to microcontroller or PCs.

Pinout

Pin	Direction	Description
clk	Input	System clock input (12 MHz typical)
rst_n	Input	Active-low reset
ui_in[7:4]	Input	Operand A (4 bits)
ui_in[3:0]	Input	Operand B (4 bits)
uio_in[2:0]	Input	Opcode selection (000:+, 001:-, 010:*, 011:/, 100:%, 101:==, 110:
uo_out[7:0]	Output	ALU result (lower 8 bits)
uio_out[0]	Output	UART TX status (mirrors tx)
tx	Output	UART TX output (9600 bps serial)

How to test

1. **Provide operands**
   Connect ui_in[7:4] for **operand A** and ui_in[3:0] for **operand B**.
2. **Choose operation**
   Set the operation using uio_in[2:0]:
   - 000: A + B
   - 001: A - B
   - 010: A * B
   - 011: A / B
   - 100: A % B
   - 101: A == B
   - 110: A > B
   - 111: A < B
3. **Read the result**
   The ALU result (8 bits) appears on uo_out[7:0].
4. **Serial output (optional)**
   The same result is sent via UART on the tx pin.
   - Connect a USB-to-serial adapter (9600 bps, 8N1) to read it on a PC or MCU.
   - uio_out[0] reflects the UART TX line state for monitoring.
5. **Reset the design**
   Drive rst_n low to reset the FSM and ALU state, then bring it high again to start a new computation.

Notes

- **Clock** : Design expects a 12Mhz input clock. (TinyTapeout standard)
- **Logic Levels** : All I/O pins use 3.3 V CMOS Logic
- **Bidirectional Pins** : Only `uio_in[2:0]` and `uio_out[0]` are actually used; others are reserved

Vision

JSilicon is not just a chip - it's a story of building silicon under constraints.

This first version (v0.1) was created entirely during mandatory military service in South Korea, demonstrating that hardware innovation is possible even in the most limited environments. Future versions will expand JSilicon into a more capable CPU core with instruction memory, register files, and possibly RISC-like capabilities.

License

Pinout

#	Input	Output	Bidirectional
0	Operand B bit 0	ALU result bit 0	Opcode bit 0 (000:+, 001:-, 010:*, 011:/, 100:%, 101:
1	Operand B bit 1	ALU result bit 1	Opcode bit 1
2	Operand B bit 2	ALU result bit 2	Opcode bit 2
3	Operand B bit 3	ALU result bit 3	
4	Operand A bit 0	ALU result bit 4	
5	Operand A bit 1	ALU result bit 5	
6	Operand A bit 2	ALU result bit 6	
7	Operand A bit 3	ALU result bit 7	UART TX output (mirrored on bit 0 of uio_out)