

AssemblyScript in C

If I had known that the project would be as difficult as it turned out, I just would have done the whole thing in C.

It's too late for that now, and I can't break 100 MIPS on a baseline system (GeekBench i7-12700K). Well, that isn't entirely true. This is the story of how I broke the 100 MIPS barrier.

The Best AssemblyScript can do

After several attempts at optimizing instructions I managed to get LDA to 95 MIPS, STA to 80 MIPS, DEC to 95 and NOP to 114. The NOP was easy as it's basically just a function call.

Next I tried the typical 'leader function' approach:

```
export function cpu_step(): void {

    const IP_now = _IP;
    let opcode = fetch_byte();

    // leader function
    let c_result = c_cpu_step(opcode);
    if (c_result === 0) {
        // C handled it successfully, we're done
        return;
    }

    switch(opcode) {
    //////////////////////////////////////
        // LDR/STR load and store architecture
    //
    //////////////////////////////////////
        case OP.LD_IMM: {
            ...
        }
    }
}
```

The results?

Pre-C NOP was 114 MIPS. Post-C... wait for it... 105 MIPS. Not bad! Then I looked at the leader function. There's an IF in the hot path now. It also damages every other instruction! So for example, if I look at LDA now it's no longer a 98 MIPS operation - it's 85. This strongly implies that the IF is also damaging POST-C's operational speed by some number. Likely 10-15 just like in AssemblyScript. The question is then, can POST-C LDA or ADD or any other compete with it's Pre-C version? And, is it worth it to convert *everything* into C?

First, the ISA has to be solid. Feature freeze. Then we move everything into C. The more we move, hopefully, the faster it gets. This is on the back burner for now, while we work on Pao Alto Tiny Basic

(see INT 05h documentation).

Observations

Not done yet

Conclusions

Not done yet

From:

<https://www.appledog.ca/wiki/> - **Appledog**

Permanent link:

https://www.appledog.ca/wiki/doku.php?id=sd:assemblyscript_in_c

Last update: **2026/01/21 01:59**

