RED semiconductor

Technical note

November 2022, Windsor, United Kingdom

RED Semiconductor benchmarks ChaCha20 cipher performance using its Vector+1 vectorisation engine with OpenPOWER® true-RISC instruction set

1 New microprocessor architecture reduces memory accesses

RED Semiconductor's innovative microprocessor ('Vantage') and vectorised instruction precursor (Vector+1) have been developed to minimise instruction count for execution of complex mathematical functions like FFT, DCT and matrix multiply, all commonly used in the computation of algorithms developed for encryption, codecs, complex trigonometry and processing of scientific data.

The inspiration for **RED's** vectorisation precursor, which creates a prefix to an existing robust scalar ISA, rather than duplicating scalar instructions with vector ones, is the original Cray vector instruction methodology. Cray set out to exploit instruction vectorisation nearly 50 years ago when memory was expensive and semiconductor technology was unable to deliver the density demanded. With availability of high-density low-cost memory, processor and instruction set architectures didn't need to be memory-efficient, but in cybersecurity there is a pernicious legacy from that intrinsic inefficiency. So half a century on, **RED** has developed a new take on the original Cray concept to perform execution of complex algorithms in Level-1 cache wherever possible, minimising read/write to instruction and data memory. Other industry approaches to complex mathematical processing like SIMD (Single Instruction Multiple Data) and duplication of scalar instructions as vector instructions simply continue the legacy of resource-inefficiency.

RED Semiconductor is a firm advocate of libre hardware and software, and is collaborating with Libre-SOC on the development and commercialisation of their 64-bit processor vector-accelerated architecture. Together we have selected the OpenPOWER® Instruction Set Architecture as the baseline for building our vectorisation engine onto, as it's a robust, well supported true-RISC ISA. (The OpenPOWER® SFFS – Scalar Fixed-point + Floating-point Subset is only 214 instructions). Through membership of the OpenPOWER Foundation, **RED Semiconductor** is influencing further development of vectorisation with our 'Vector+1' methodology.

Why our obsession with minimum or zero memory bus accesses? Every bus access can be leaked or hacked, so by executing only within L1 cache we keep the entire process securely buried in registers and cache. This has the security-critical benefit of ensuring that when executing an encryption algorithm, the cipher information – seed and private key – are never

e info@redsemiconductor.com
w redsemiconductor.com

exposed onto a bus from which they can be accessed or hacked, and the risk of timing-based attacks is mitigated. At **RED** we are developing our vectorisation engine ('Vector+1') and microprocessor architecture (Vantage) to give assurance to customers that their critical information remains secure.

2 ChaCha20 cipher benchmark

ChaCha20 is a stream cipher developed by Daniel J. Bernstein. It uses a new round function that increases diffusion and increases performance on some architectures.
It is built on a remarkably simple pseudorandom function based on add-rotate-XOR (ARX) operations — 32-bit addition, bitwise addition (XOR) and rotation operations. The core function maps a 256-bit key, a 64-bit nonce, and a 64-bit counter to a 512-bit block of the key. This gives ChaCha20 the unusual advantage that the user can efficiently seek to any position in the key stream in constant time. ChaCha20 offers speeds of around 4–14 cycles per byte in software on modern x86 processors, and reasonable hardware performance. It is not patented, and its originator has written several public domain implementations optimized for common architectures. Moreover, ChaCha20 is considered to be more efficient that the prevalent AES encryption algorithm in terms of speed of execution.

ChaCha20 is therefore an excellent benchmark to put **RED Semiconductor's** 'Vector+1' to the test against publicly verifiable competitive solutions.

Here's an implementation of ChaCha20 in C/C++:

```
#define ROTL(a,b) (((a) << (b)) | ((a) >> (32 - (b))))
#define QR(a, b, c, d) (                   \
        a += b,  d ^= a,  d = ROTL(d,16), \
        c += d,  b ^= c,  b = ROTL(b,12), \
        a += b,  d ^= a,  d = ROTL(d, 8), \
        c += d,  b ^= c,  b = ROTL(b, 7))
#define ROUNDS 20

void chacha_block(uint32_t out[16], uint32_t const in[16])
{
        int i;
        uint32_t x[16];

        for (i = 0; i < 16; ++i)
                x[i] = in[i];
        // 10 loops × 2 rounds/loop = 20 rounds
        for (i = 0; i < ROUNDS; i += 2) {
                // Odd round
                QR(x[0], x[4], x[ 8], x[12]); // column 0
                QR(x[1], x[5], x[ 9], x[13]); // column 1
                QR(x[2], x[6], x[10], x[14]); // column 2
                QR(x[3], x[7], x[11], x[15]); // column 3
                // Even round
                QR(x[0], x[5], x[10], x[15]); // diagonal 1 (main diagonal)
                QR(x[1], x[6], x[11], x[12]); // diagonal 2
                QR(x[2], x[7], x[ 8], x[13]); // diagonal 3
                QR(x[3], x[4], x[ 9], x[14]); // diagonal 4
        }
        for (i = 0; i < 16; ++i)
                out[i] = x[i] + in[i];
}
```

3 Benchmark results

The simplicity of **RED's** Vector+1 approach started with the speed of development and optimisation. With a working knowledge of the OpenPOWER® ISA, it took one of our software engineer a matter of a few hours to optimise the full 20 rounds of ChaCha to a minimum instruction count. The combination of Vector+1 and OpenPOWER® ISA means that the entire software implementation uses general-purpose instructions, with no need for special instruction set extensions.

The result? ChaCha20 all rounds executed 100% deterministically in only 10 instructions. Moreover, the execution fitted in only a single line of L1 Instruction Cache.

By comparison, we reviewed third-party implementations for both x86 and ARM ISAs which took around 500 scalar instructions to perform the same algorithm, and where there were extensive read/writes to SRAM memory. Furthermore, a third-party investigation into optimising RISC-V to run ChaCha20 efficiently showed a best-case requirement for about 500 instructions which included specially devised instructions as extensions to the already available RISC-V instructions. Thus the result was a complex development cycle without competitive performance gain.

Why such a big difference? 10 instructions vs. 500 instructions! The main reason is that other instruction sets require loop-unrolling to achieve Deterministic Behaviour (which utilises hundreds of instructions). The **RED** 'Vector+1' vectorisation engine uses Vertical-First Vector Mode and Deterministic Branch Loops, completely avoiding non-deterministic behaviour. The loops are so small that micro-coded parallelism is practical. Constant time is achievable due to both loops for all 20 rounds being 100% Deterministic. L1 I-Cache misses are avoided which could leak information which is a problem in other ISAs due to massive loop-unrolling. There is a further benefit in **RED Semiconductor's** silicon implementation – the huge gain in instruction efficiency means that power consumption is also greatly reduced without compromising performance.

Summary of RED 'Vector+1' benefits:

	RED Semiconductor 'Vector+1' precursor and OpenPOWER® ISA	Traditional ISA (x86, ARM)
Number of Instructions to execute 20 rounds of ChaCha cipher:	10 instructions	>500 instructions
Execution time determinism:	100% deterministic constant timing	Non-deterministic timing – depends on cache availability
Execution memory requirements:	Fits in single L1 I-cache line – zero cache misses	Requires multiple cache lines – risk of cache misses and data leakage
Power consumption to execute:	Lowest power consumption for stream cipher encryption	Significantly greater power consumption
Versatility:	Cipher upgrades easily accommodated	Software redesign required for cipher upgrades
Ease of implementation:	Straightforward and quick	Complex and time-consuming

4 Hardware implementation

RED Semiconductor is working with industry partners to accelerate the implementation of a 1-core ‘Vantage’ cipher-accelerator microprocessor chipset that will deliver to our customers a rapid means of integrating the most secure encryption into their systems. Planned for first-customer shipments in early 2024, the ‘Vantage1’ chip will be targeted at data security and privacy applications, in particular Secure IOT, secure communications, and critical infrastructure market sectors.

For further details and to be included in our lead customer programme please contact:

James Lewis
CEO, **RED Semiconductor** Ltd
james.lewis@redsemiconductor.com
+44 (0)7903 849974