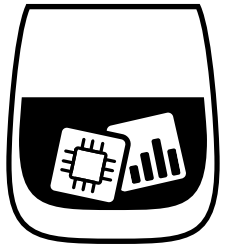


Scotch: Generating FPGA-Accelerators for Sketching at Line Rate

VLDB 2021, Copenhagen, Denmark
2021-08-18

Martin Kiefer, Ilias Poulakis, Sebastian Breß, Volker Markl
martin.kiefer@tu-berlin.de
Technische Universität Berlin / DFKI

Sketching on FPGAs

Sketching

- Constructing stream summaries
- e.g., Count-Min, AGMS, HLL, …
- Various applications (e.g., AQP, ML, Network)

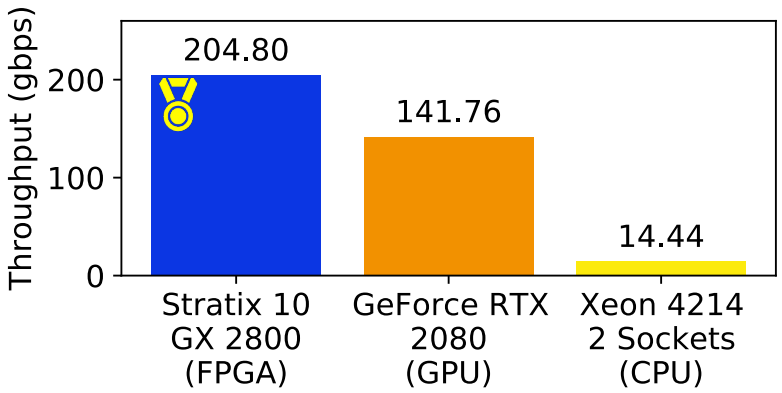
FPGAs

- Custom hardware from software
- Circuit-level parallelism
  - Data / Task / Pipeline

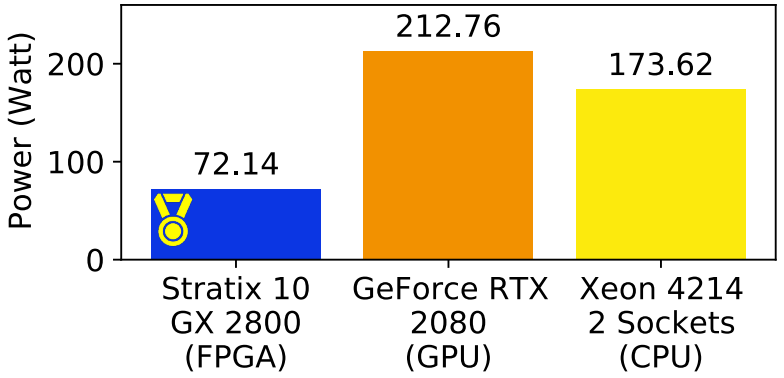
Sketching on FPGAs

- High guaranteed throughput
- Lower power consumption

Count-Min (8 Rows, 40k Columns)



↑ 1.4 - 14x throughput



↑ 2.4 - 2.9x power draw

Outline

Challenges & Scotch’s Approach

Scotch

Evaluation

Summary

Challenges & Scotch's Approach

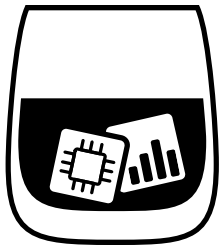
Challenges:

An FPGA expert is required

- Device-, vendor-, interconnect-specific implementations
- Register-Transfer Level (RTL) programming
- Manual tuning
  - Sketch size vs. resources & timing

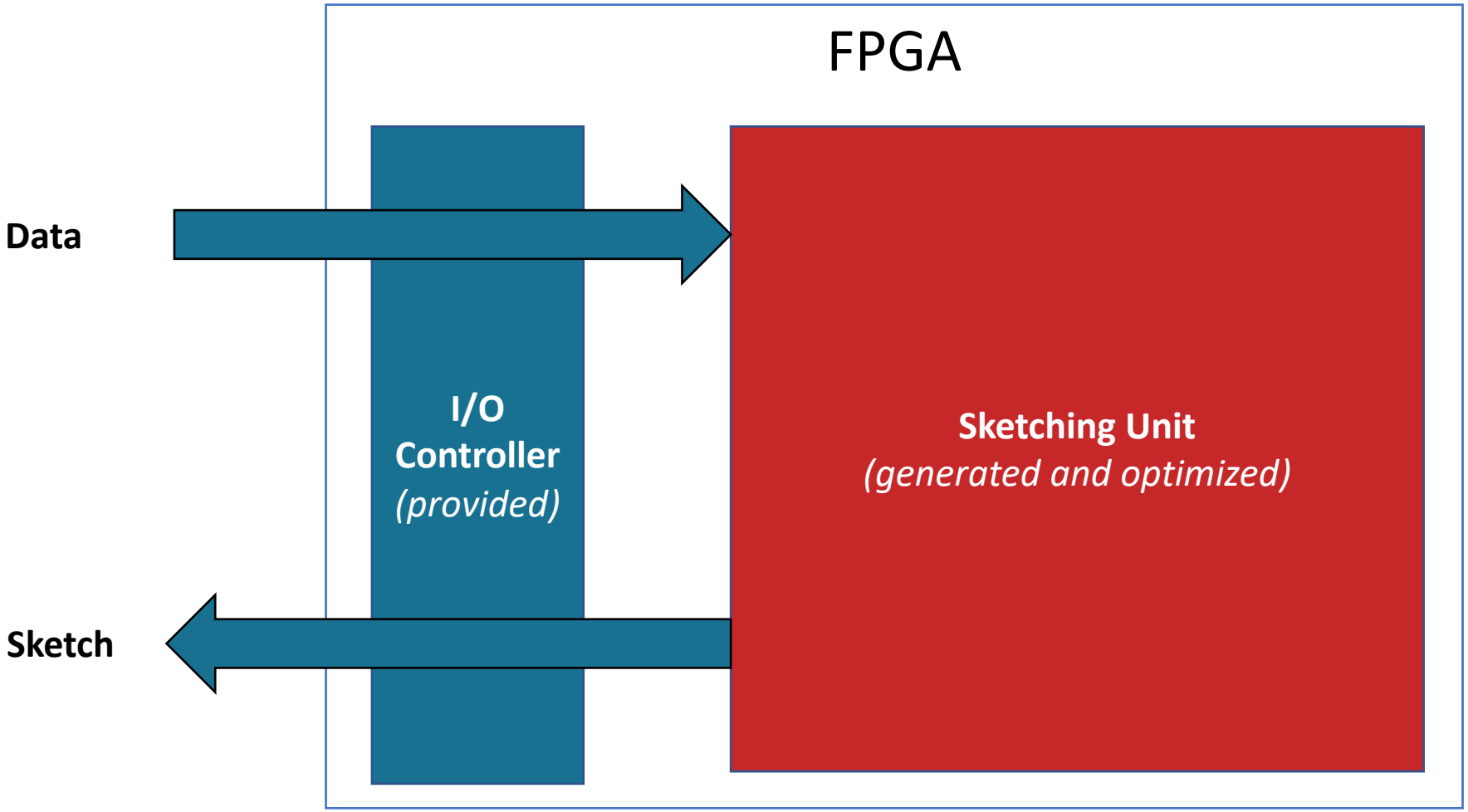
Scotch's Approach:

Abstract and Automate

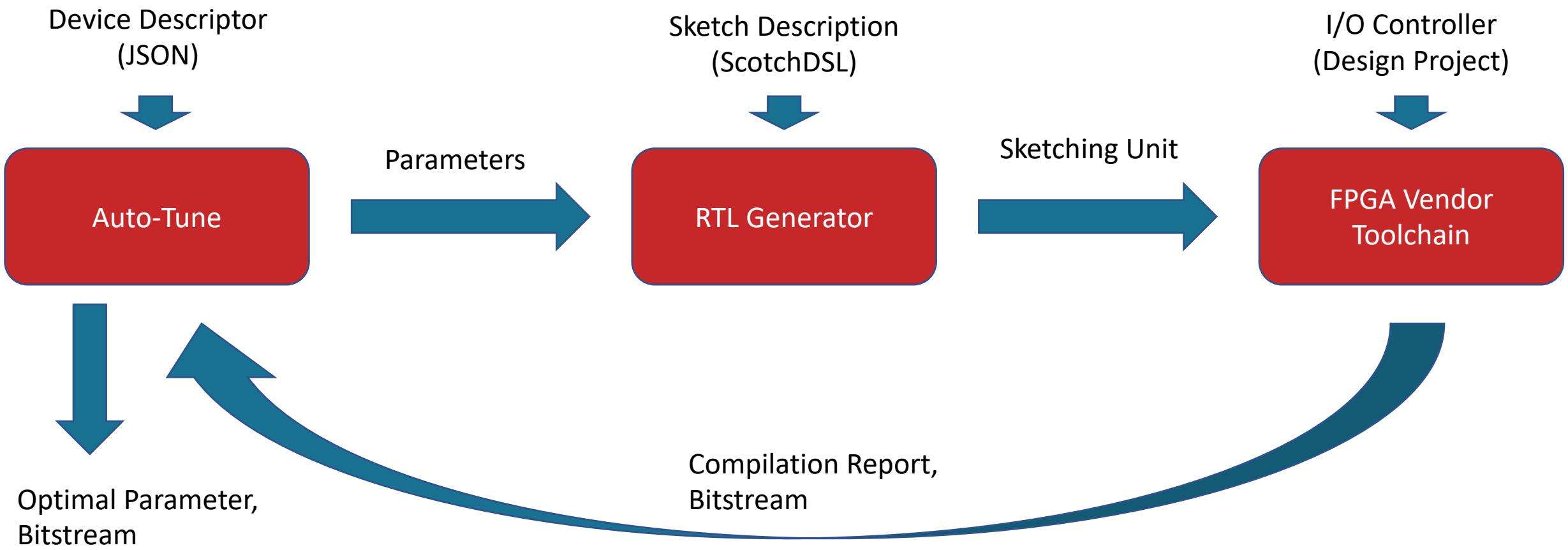


- Device and I/O agnosticism
- Lightweight sketch specification
- RTL generation
- Automated tuning

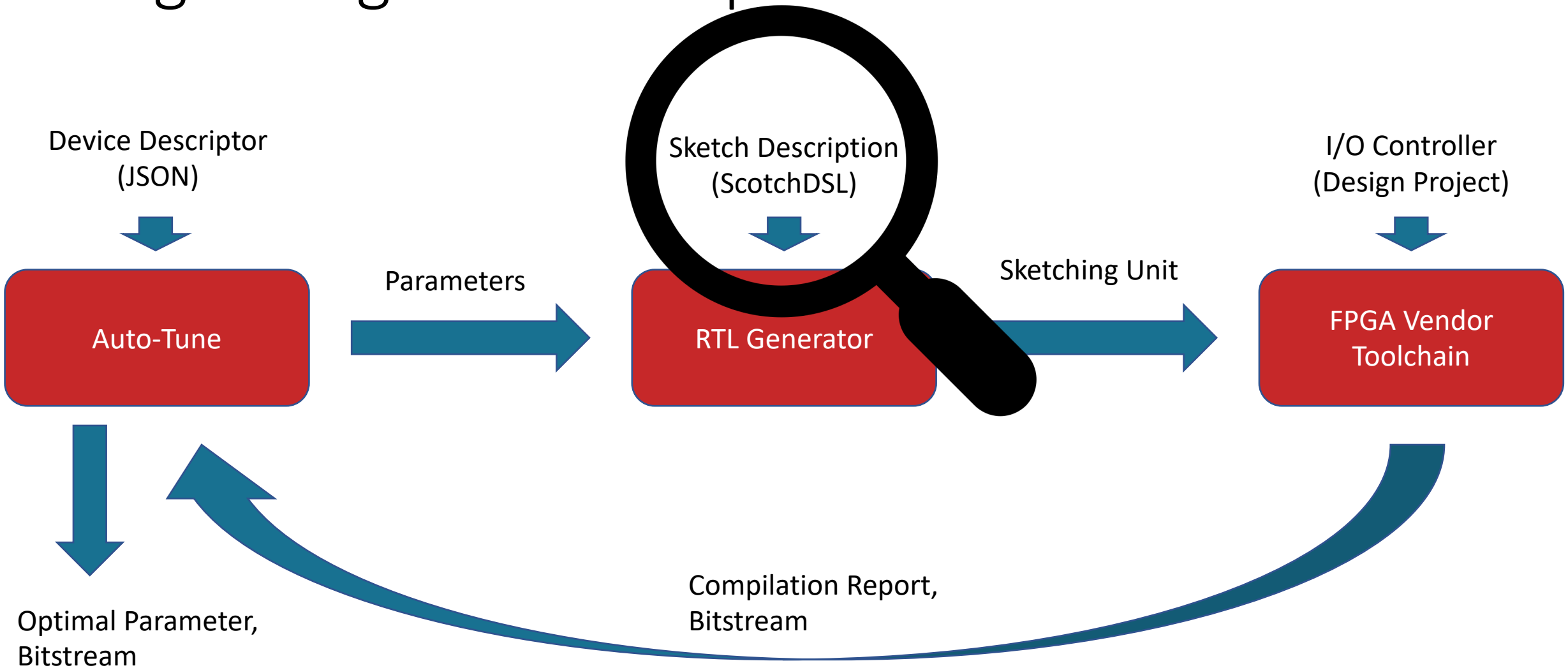
Device and I/O Agnosticism: Accelerator Architecture



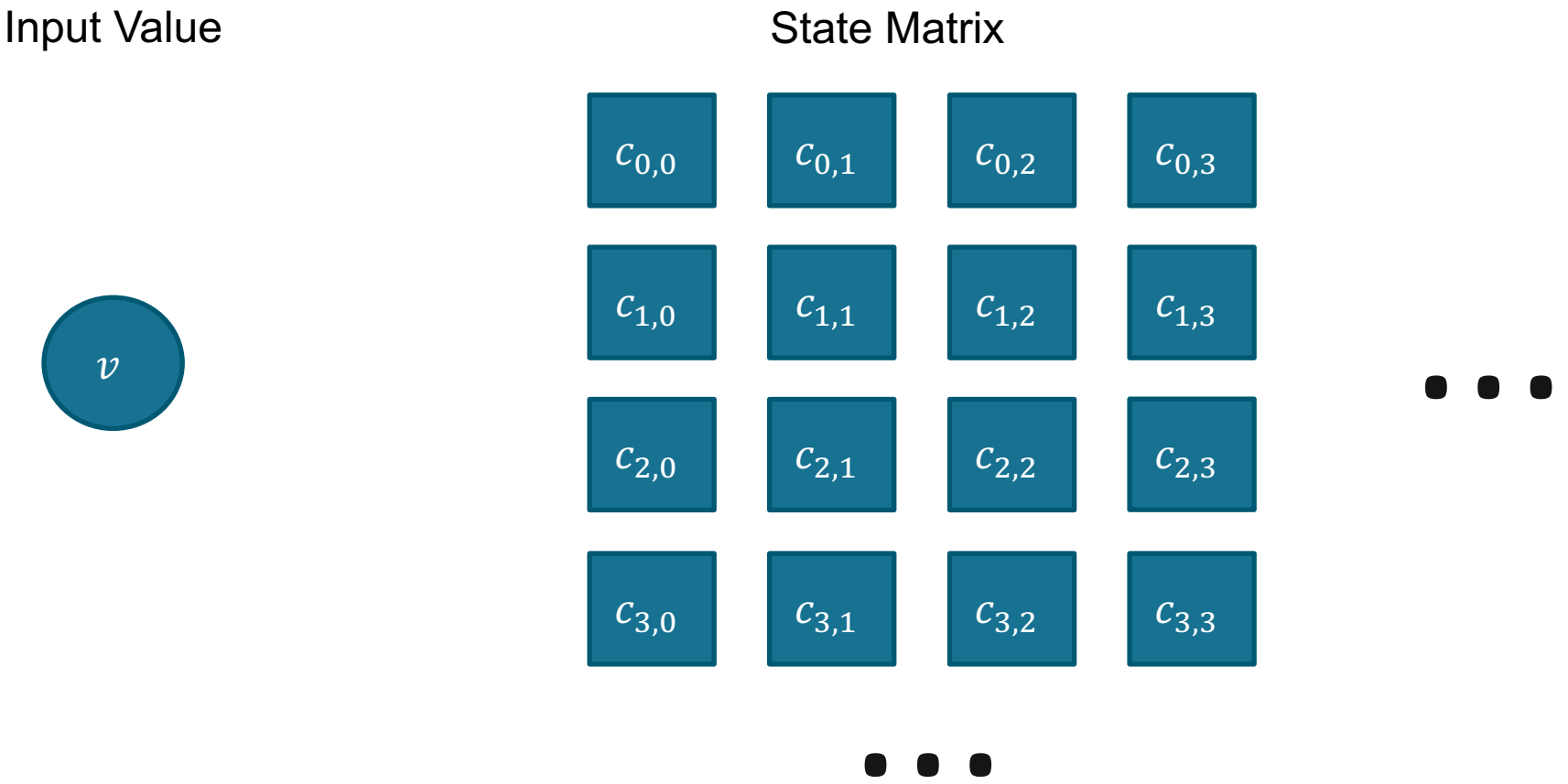
Scotch System Architecture



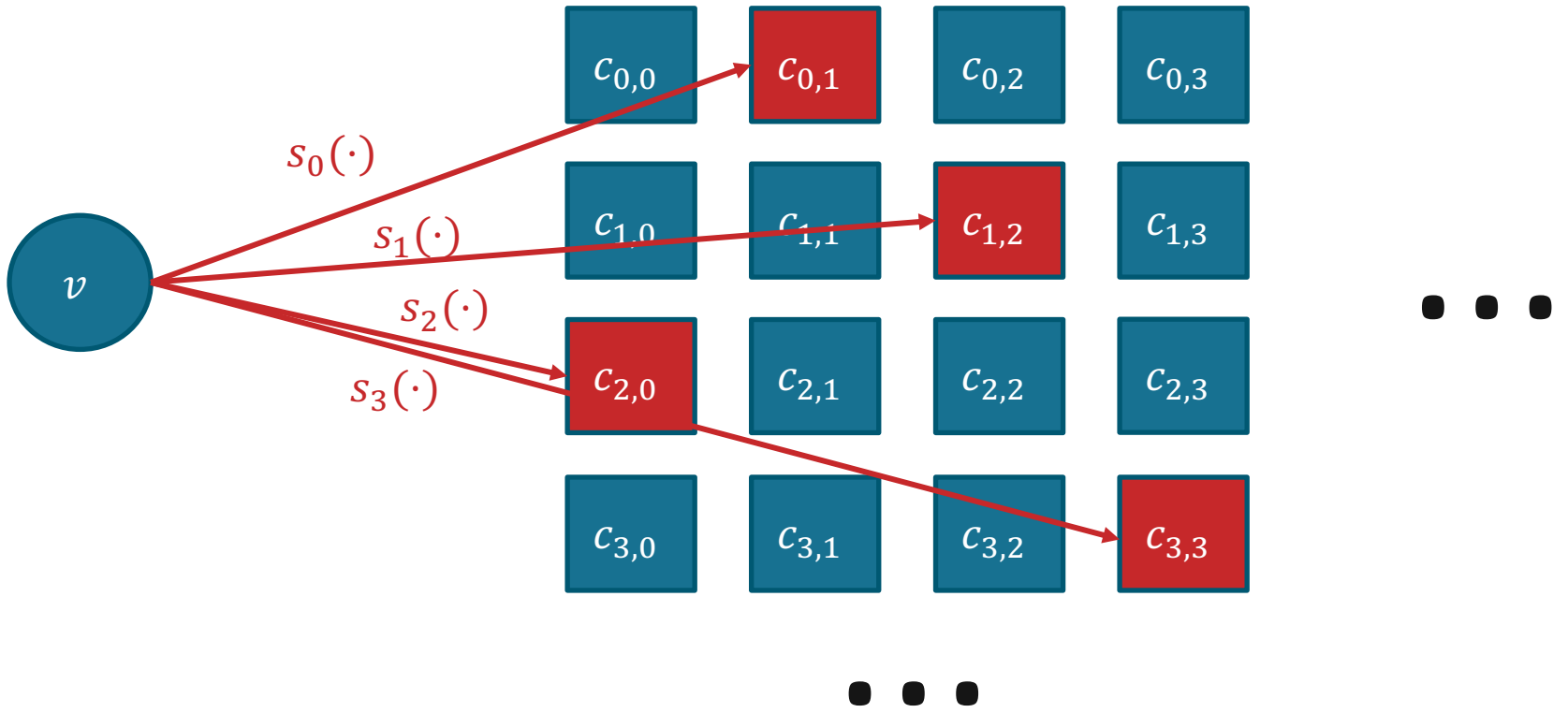
Lightweight Sketch-Specification



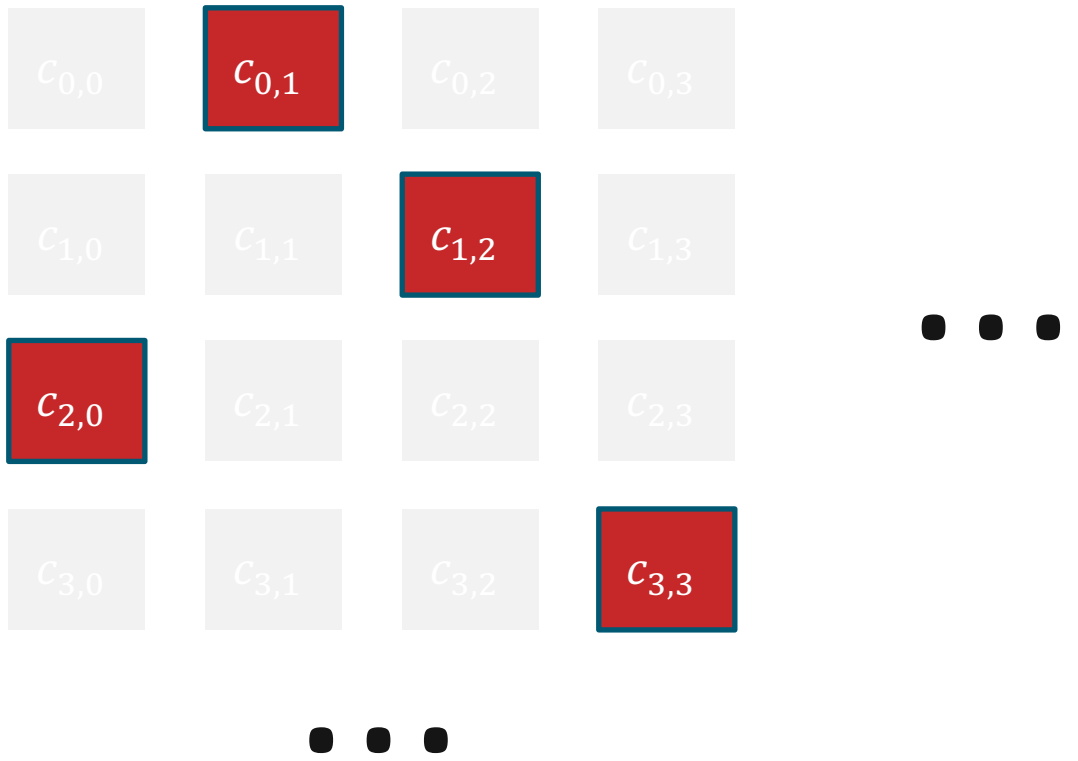
Lightweight Sketch Specification: Select-Update



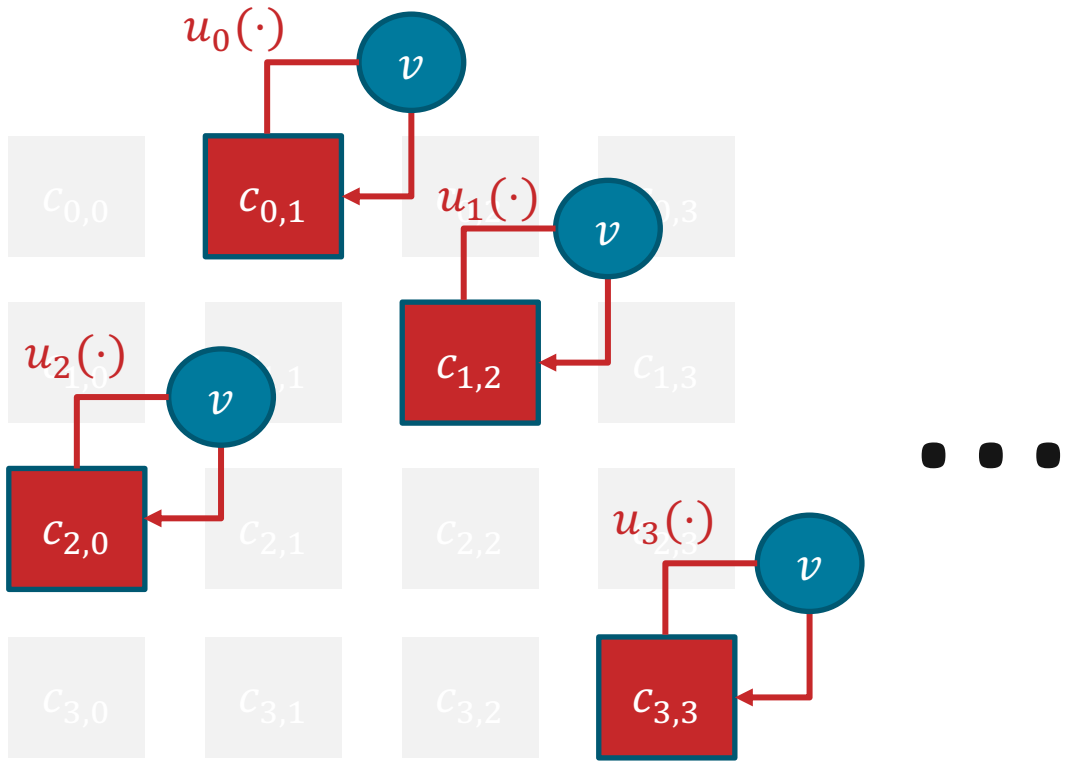
Lightweight Sketch Specification: Select-Update



Lightweight Sketch Specification: Select-Update



Lightweight Sketch Specification: Select-Update



Lightweight Sketch Specification: ScotchDSL

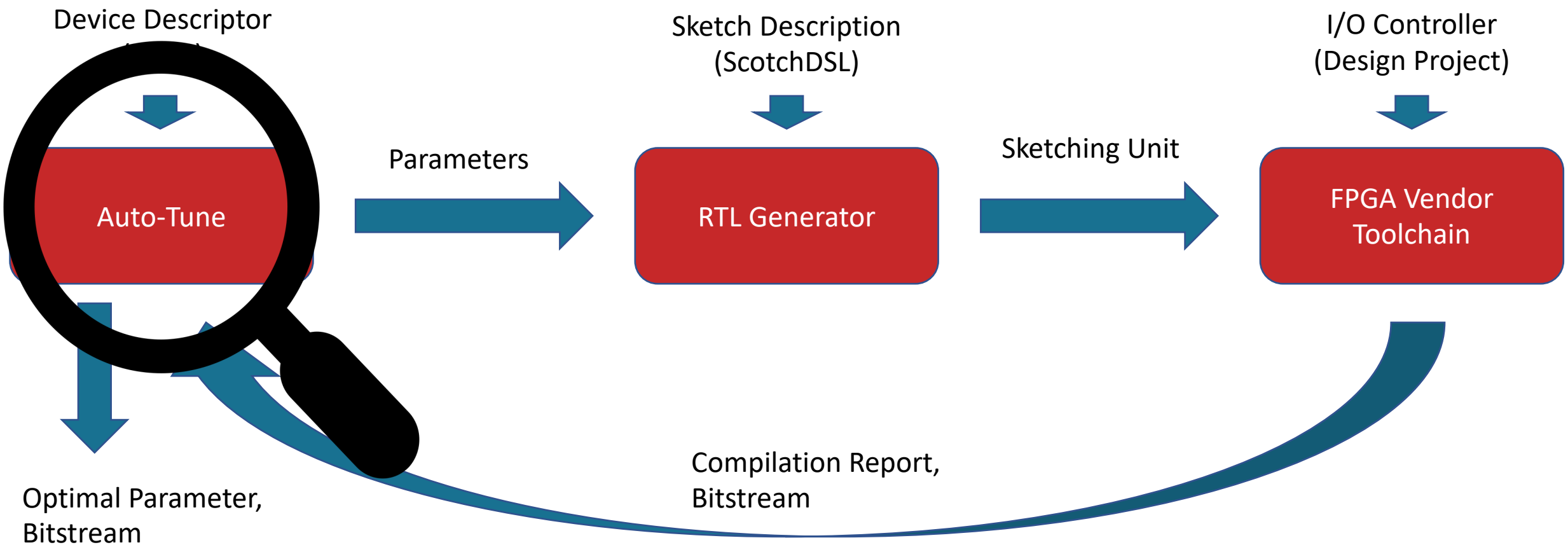
- DSL for select and update functions
- Bitvectors as first-class citizens
- Logic and arithmetic operations
- Restricted control flow
- Clocks and control signals are not exposed

EH3 Update Function (+1/-1 Updates):

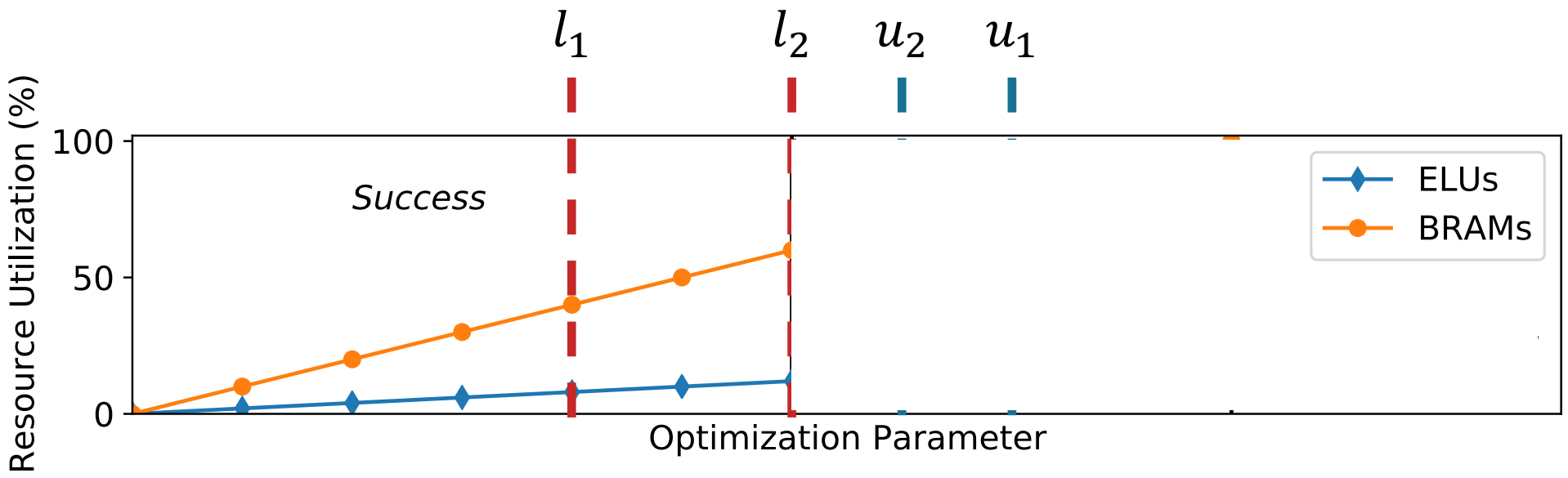
```
update(seed, v, state, outstate){
        mask <= '10101010101010101010101010101';
        h    <= parity(v(30 downto 0) | v(31 downto 1) & mask);
        eh3 <= seed(0) ^ parity(seed(32 downto 1) & v) ^ h;

        outstate <= eh3 = '0' ? signed(state) + 1 : signed(state) - 1;
}
```

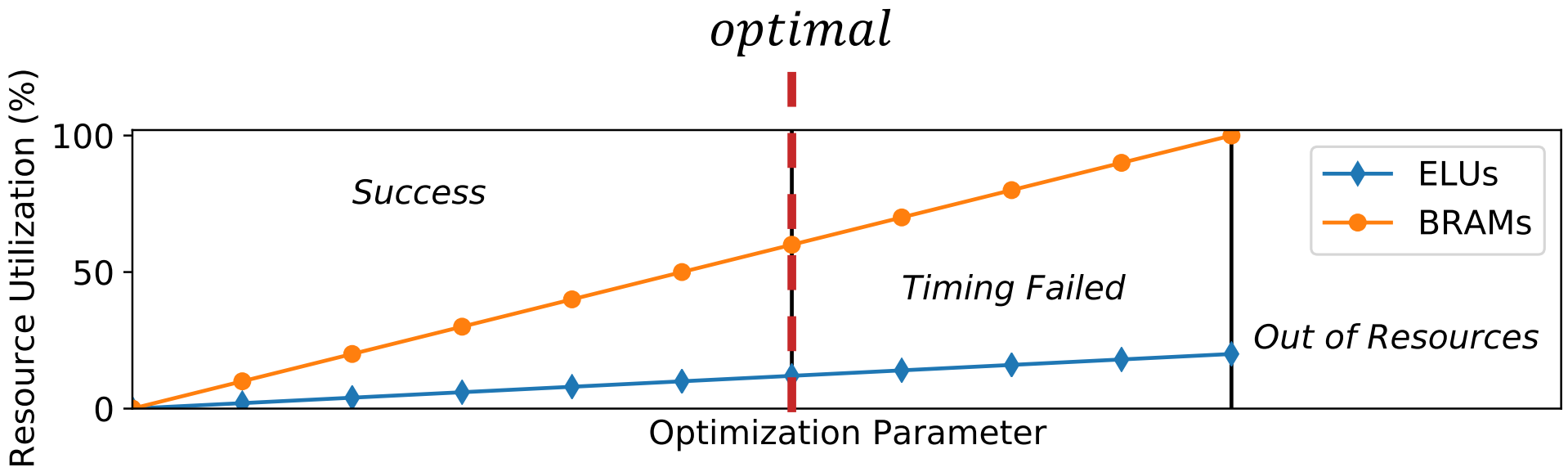
Automated Tuning



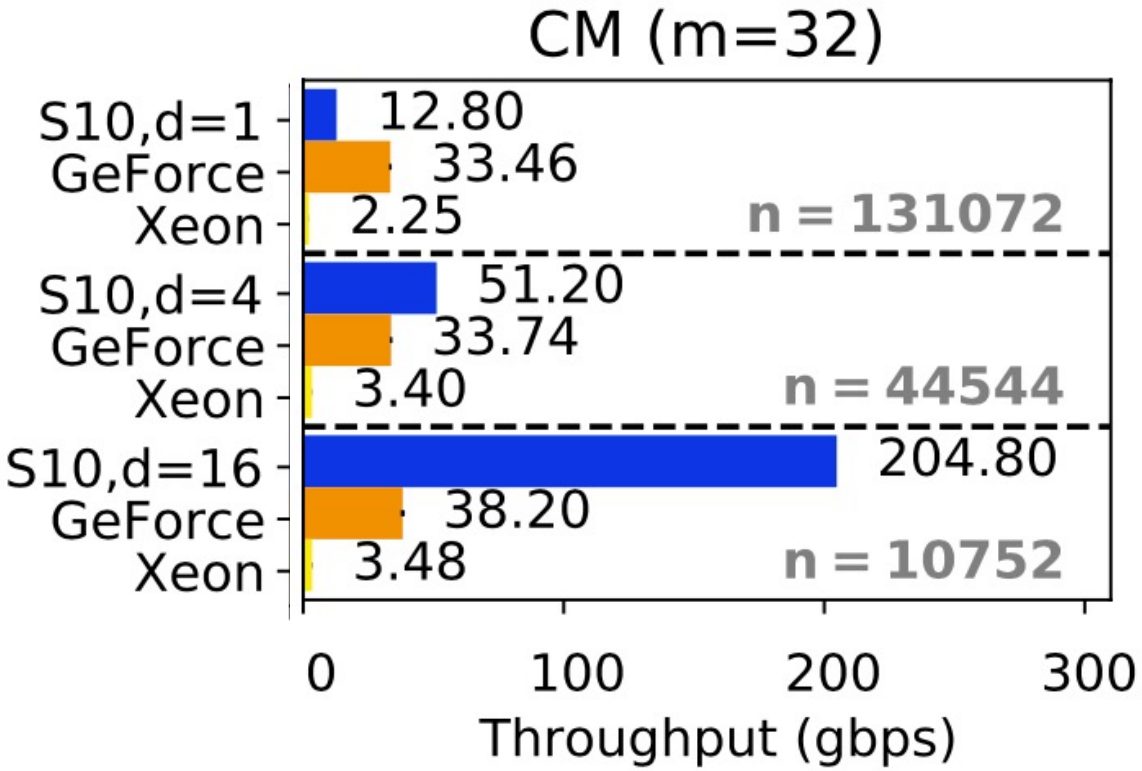
Automated Tuning:
Binary Search in the Parameter Space



Automated Tuning:
Binary Search in the Parameter Space



Evaluation (Count-Min, 400 Mhz)

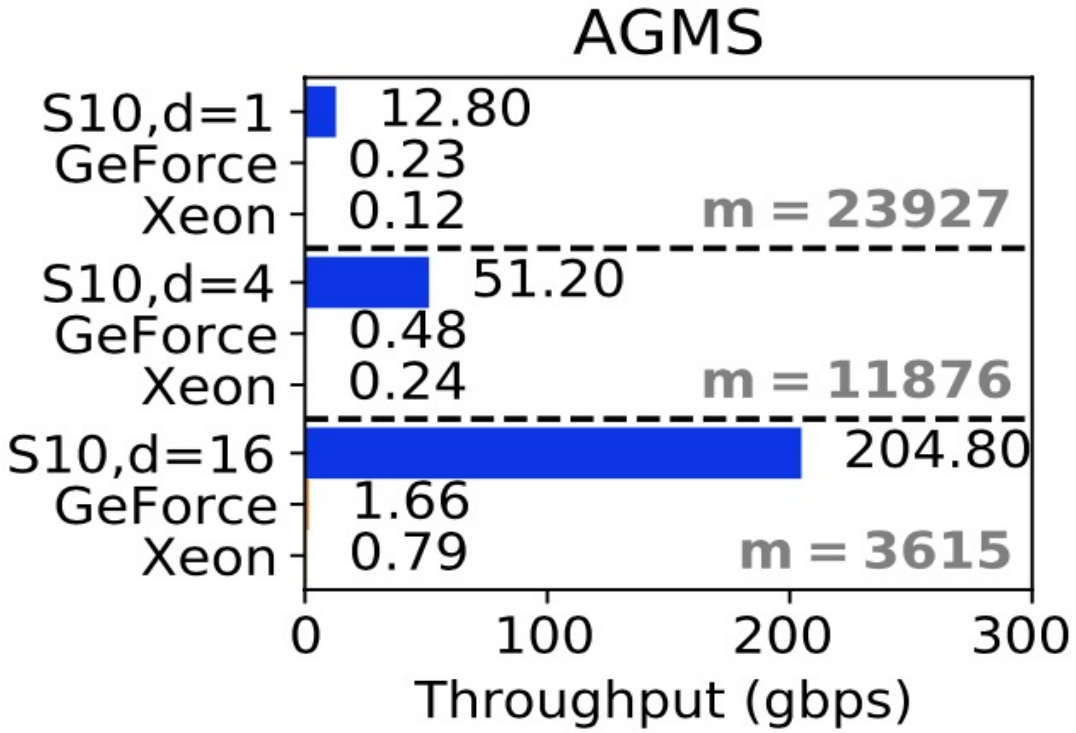


- Data parallelism is crucial for high throughput
- Tradeoff between throughput and summary size

Stratix **10** GX 2800 (400 MHz) | **GeForce** RTX 2080 Ti | Intel **Xeon** Silver 4214 | 32 Bit State & Input | Uniform Data

Evaluation (AGMS, n=1)

- *Each row is updated for every input value*

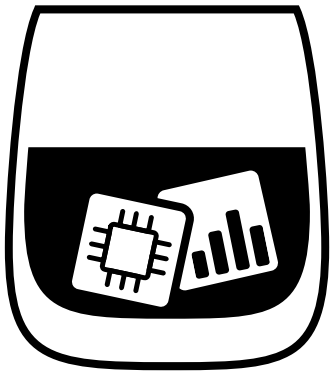


- FPGAs fit compute-intensive sketches excellently

Stratix **10** GX 2800 (400 MHz) | **GeForce** RTX 2080 Ti | Intel **Xeon** Silver 4214 | 32 Bit State & Input | Uniform Data

Summary

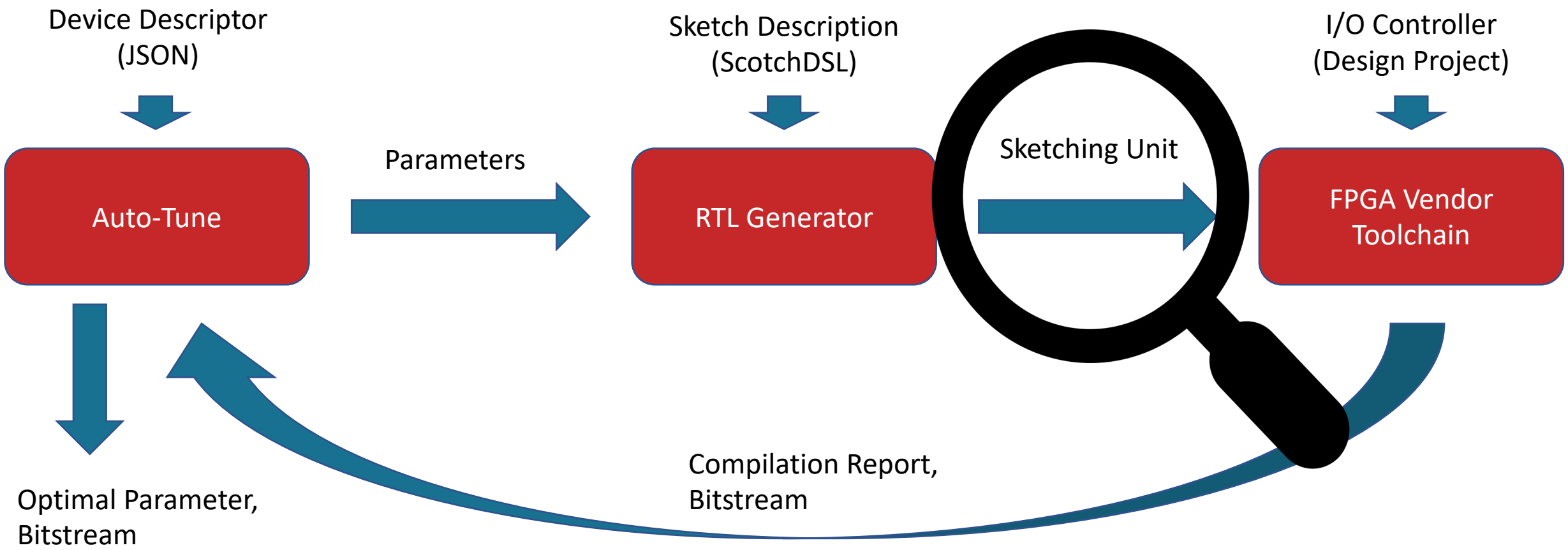
- Generating FPGA-based sketching accelerators
  - ScotchDSL
  - RTL generator
  - Auto-tune
- High guaranteed throughput
- Low power consumption
- No expert in the loop
- But wait, there's more!
  - Sketching unit
  - Data parallelism strategies
  - Extensive evaluation



GitHub

martinkiefer/scotch

Constant Processing Rate: Sketching Unit Architecture



Constant Processing Rate: Sketching Unit Architecture