Function Replication in FPGA for Compute Intensive Applications

MD5 Bruteforcer

MD5 is a hashing algorithm

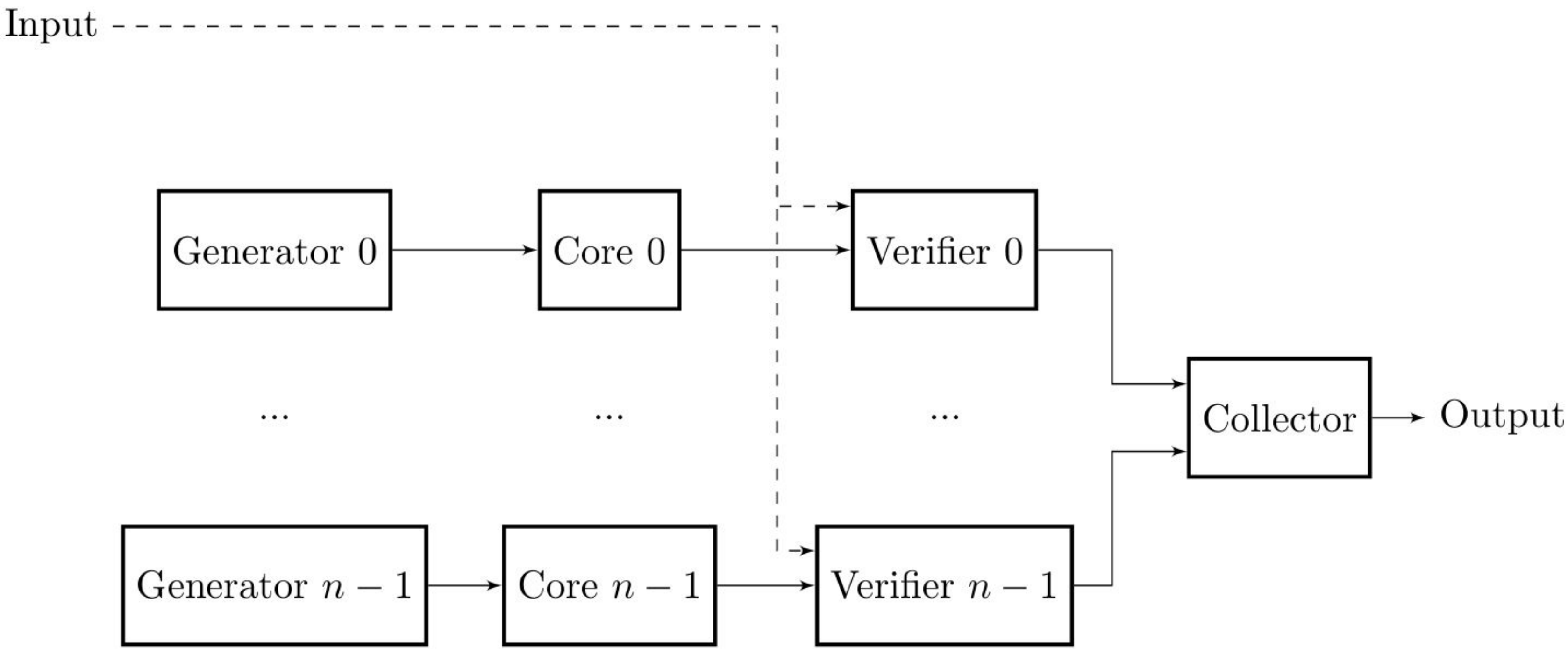
Getting the original value from a hash cannot be computed, but can be guessed.

Each guess can be performed in parallel

Structure of a solver

- Input generator
- MD5 computation
    - Pack data
    - Perform computation loop (64 iterations)
    - Add initial values
- Check whether computed hash matches target hash
- To parallelize each generator needs to know its own range, and all of the verifiers needs to be “connected” when a result has been found

Structure of a brute forcer



Different implementations

- C
  - For CPU
  - For FPGA (HLS)
- OpenCL
  - For GPGPU
  - For FPGA (HLS)
- SME
  - For FPGA

C implementation

- Each part of a solver becomes a function
- The generator calls the core, the core calls the verifier
- A shared variable for indicating that a result has been found
- The solvers are run in a parallel OpenMP loop

- For HLS, additional tuning was required
  - Loop unrolling
  - Pipelining
  - Array “unrolling”

OpenCL implementation

Exactly the same structure as the C implementation, except for the requirement for a host program.

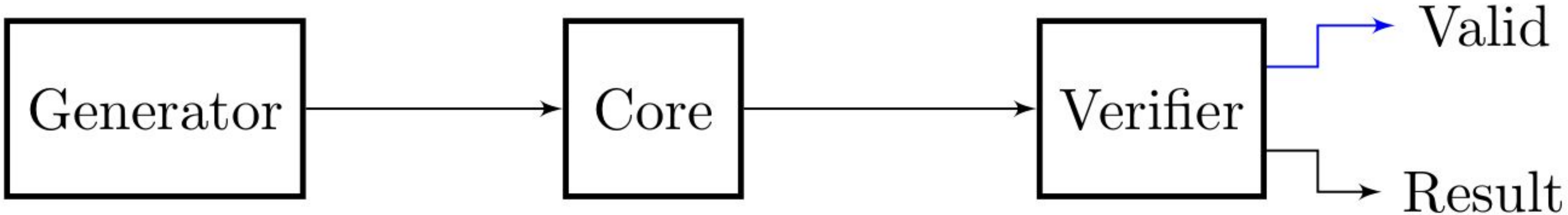
Instead of having a shared variable, they run for a set number of iterations. This is due to the lack of shared resources in OpenCL.

For each set of iterations, the host checks whether any result has been found. If not - they run for another set of iterations.

The same optimisations for FPGA has been made in HLS as with the C implementation.

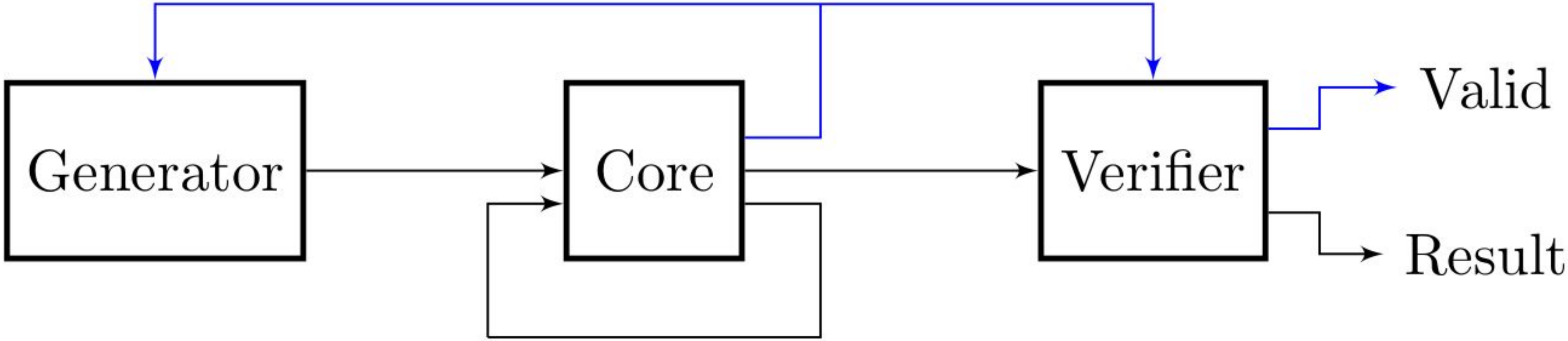
SME simple implementation

- In the simple implementation, the core performs all 64 iterations of the core loop in 1 clock cycle.



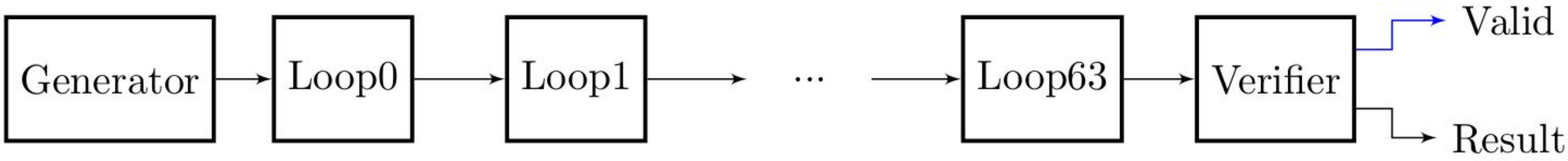
SME compact implementation

- In the compact implementation, the core loop only executes one iteration of the loop per clock cycle. This results in increased latency (64 clock cycles).

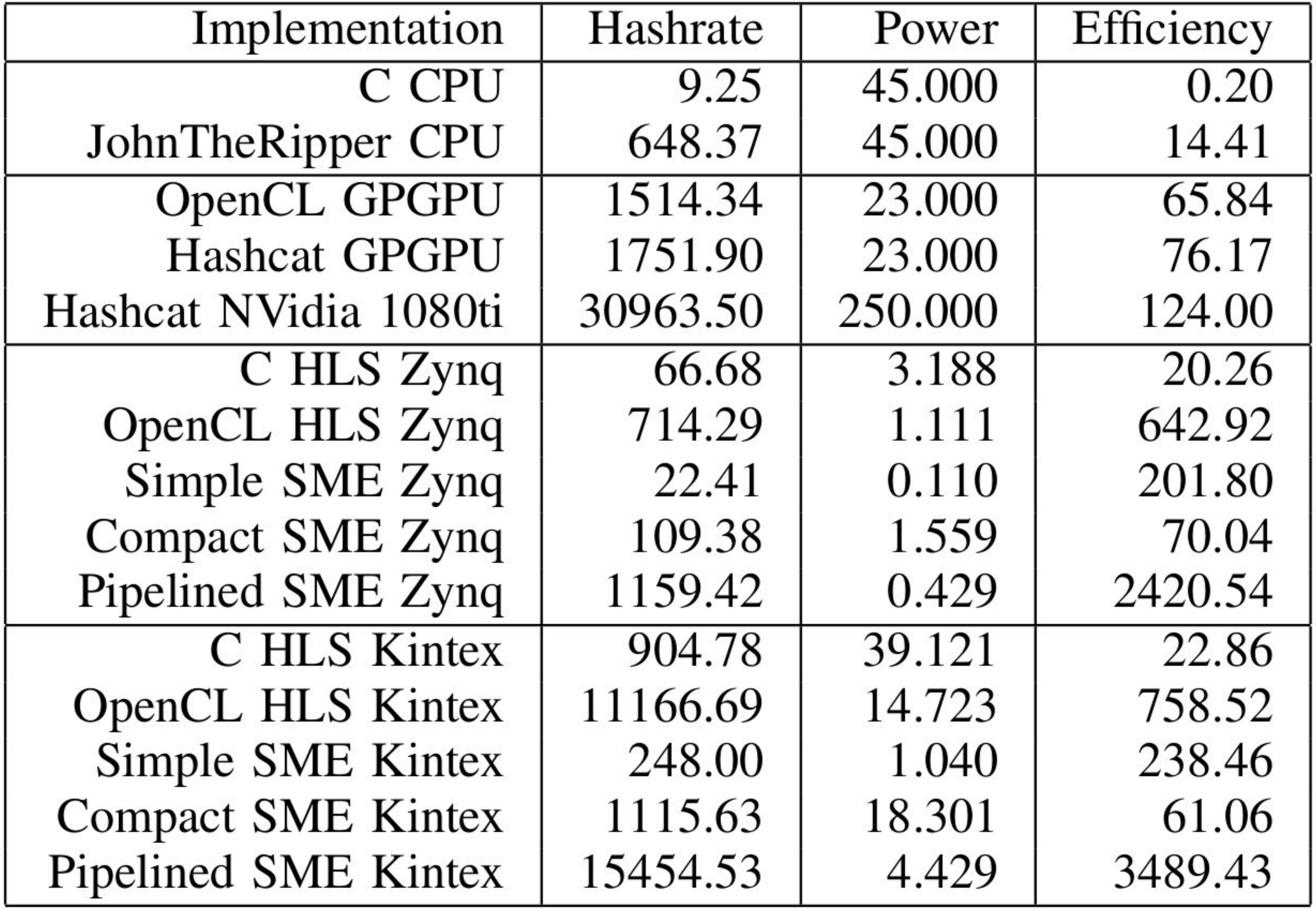


SME Pipelined implementation

- In the pipelined implementation, each iteration of the loop is put into its own process. Once the pipeline has been filled, it computes one hash per clock cycle.



Results



Implementation	Hashrate	Power	Efficiency
C CPU	9.25	45.000	0.20
JohnTheRipper CPU	648.37	45.000	14.41
OpenCL GPGPU	1514.34	23.000	65.84
Hashcat GPGPU	1751.90	23.000	76.17
Hashcat NVidia 1080ti	30963.50	250.000	124.00
C HLS Zynq	66.68	3.188	20.26
OpenCL HLS Zynq	714.29	1.111	642.92
Simple SME Zynq	22.41	0.110	201.80
Compact SME Zynq	109.38	1.559	70.04
Pipelined SME Zynq	1159.42	0.429	2420.54
C HLS Kintex	904.78	39.121	22.86
OpenCL HLS Kintex	11166.69	14.723	758.52
Simple SME Kintex	248.00	1.040	238.46
Compact SME Kintex	1115.63	18.301	61.06
Pipelined SME Kintex	15454.53	4.429	3489.43