“Simple Single Instruction Multi Data (SIMD) Processor”

Abdullah Al Imran Ifrit
Dept. of Electrical and Electronic Engineering,
Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
Student id. 180105057

Zarin Tasnim Oishee
Dept. of Electrical and Electronic Engineering,
Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
Student id. 180105067

Kazi Aklima Sultana Urmi
Dept. of Electrical and Electronic Engineering,
Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
Student id. 180105070

Fabi Nahian Madhurja
Dept. of Electrical and Electronic Engineering,
Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
Student id. 180105090

Nasif Uddin
Dept. of Electrical and Electronic Engineering,
Ahsanullah University of Science and Technology, Dhaka, Bangladesh.
Student id. 180105097

***Abstract*—**

As the dimensions of electronic devices become smaller, they enter the realm of nano-electronics, where the device dimensions lie between the interatomic spacing (1nm) and the electron wavelength (10nm). This offers greatly-enhanced packing density and speed of operation but introduces the difficulty of maintaining signal quality over large distances. The paper introduces an architectural solution to this problem that is appropriate for the class of highly data-parallel computers [1,2,3]. In the propagated instruction architecture proposed here, instructions sweep across the array (and hence across the data set) in bands, one instruction following another closely. This paper shows a new architecture and sample algorithms of a vision chip that has the ability to reconfigure its hardware dynamically by chaining processing elements.

***Keywords*— SMD, ALU, Adder, Shifter, Register, Violations.**

1. Introduction

This project is to implement a simple SIMD processor, the core of which is a 16-bit SIMD ALU. 2's complement calculations are implemented in this ALU. The ALU operation will take two clocks. The first clock cycle will be used to load values into the registers. The second will be for performing the operations. 6-bit opcodes are used to select the functions. The instruction code, including the opcode, will be 18-bit. The ALU will be embedded into a simple processor based on a 5-stage, delay each stage will be 1 cycle, meeting the delay of the ALU, as shown in the figure below. The 5 typical stages are IF, ID, EX, MEM, and WB, without the pipeline. In stage IF, a 10-bit address will be sent to an instruction Block-RAM (BRAM) to fetch an 18-bit instruction. In the stage ID, the instruction will be decoded and some of the control registers will be set to control the following stage. In stage EX, ALU will process data in registers or implement some control commands, e.g. jump. In stage MEM, if the instruction is “store” or “load”, data would be read from written to data BRAM, based on instruction and address. Finally, in stage WB, data will be written back to the register. The pins of the clock, reset, address, data and BRAM enable will be exposed on the interface of the processor. The architecture of the SIMD processor is shown below in Figure 1.

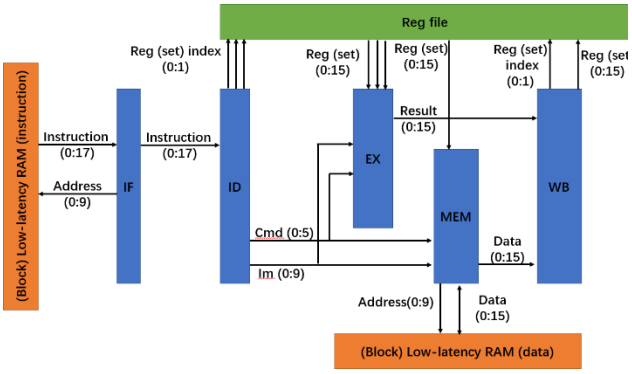
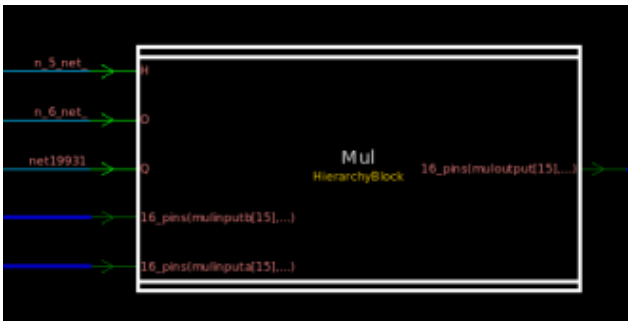
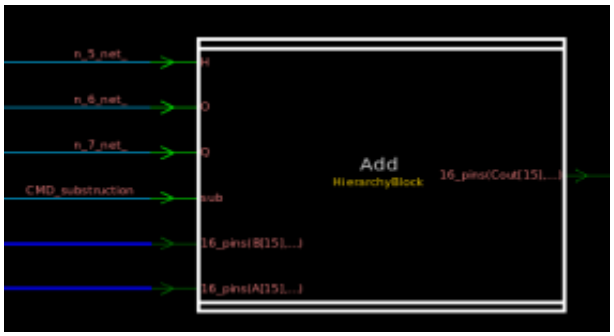


Figure 1: Simple SIMD Processor Architecture

i) 1-Design of SIMD ALU

The ALU consists of three SIMD basic computation units: SIMD adder, SIMD multiplier, and SIMD shifter. These three components can be reused for the computation of data with different widths (4-bit, 8-bit, or 16-bit) and can handle all the instructions (listed in **Section 4**) in the design specification. Each of them is controlled by three input ports, H (for 16-bit), O (for 8-bit), and Q (for 4-bit), indicating the kind of input data, so that the unit can handle the data properly, as shown below.





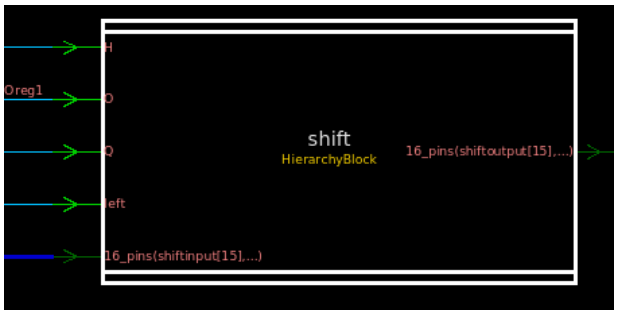


Figure 2: the interface of ALU components.

ii) SIMD adder:

The SIMD adder is implemented based on 4 4-bit adders. To reuse the adder for data with different widths, the input signals, H, O, and Q, control the forwarding of the carriers between the adders. Moreover, the adder can also support subtraction, by flipping the second operand and adding one to it when the signal sub is set. The data path of the SIMD adder is shown in **Figure 3.**

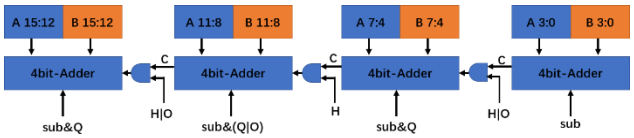


Figure 3: Example of SIMD Adder

iii) SIMD shifter:

The SIMD shifter can also support data with different widths. It is based on two 16-bit shifters and necessary overstep-correct logic. To illustrate the implementation of the shifter, the example based on logic-left-shift is shown in Figure 3. The shifter will determine whether the MSB of the 4-bit block should be set to 0 or just inherit the value from the LSB from the 4-bit block in front of it, according to the input signal, H and O.

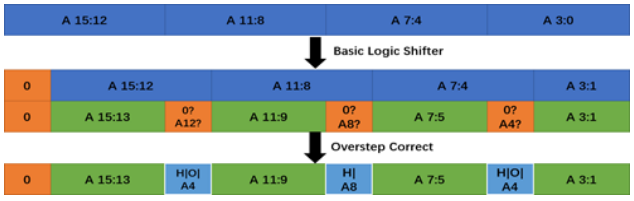


Figure 4: Example of SIMD Shifter

iv) SIMD multiplier:

The SIMD multiplier is implemented with adders and shifters. To make it support multiplication with different data widths, the input signal, H, O, and Q, are used to control the input operands of the adders and select corresponding outputs for the target data width. The 1x16x16 multiplication is implemented as a typical multiplier, as **Figure 5**, where the adder tree is utilized. The least significant 16 bits of the sum is the output of multiplication.

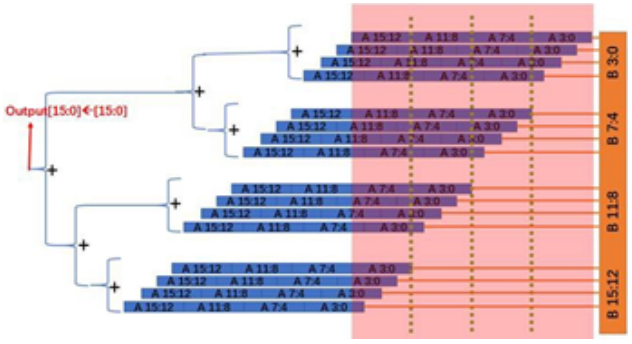


Figure 5: 1x16x16 multiplication

To reuse the structure of 1x16x16 multiplication implemented as above, for the input signal, H, O, and Q, are used to select the inputs of the adders, to control the adders to only sum up the range of bits. The 4x4x4 multiplication is implemented as shown in **Figure 5**, where for each adder, just a part of the input bit is valid while other bits will be set to 0. Similarly, the 2x8x8 multiplication is shown in **Figure 6.**

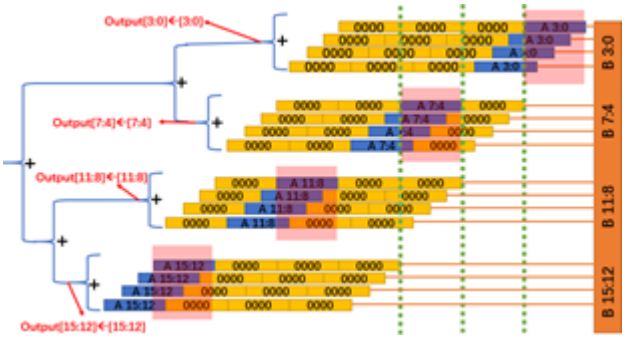


Figure 6: 4x4x4 multiplication

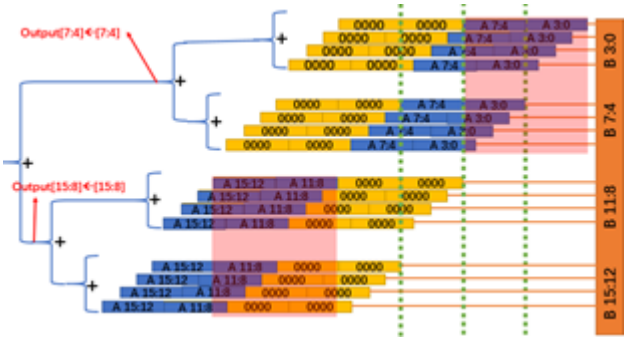


Figure 6: 2x8x8 multiplication

v) Register File

There are registers that are used to store the data for computation.

16-bit registers : H1, H2, H3, H4

8-bit registers (3 sets) : {O1, O2}, {O3, O4}, {O5, O6}

4-bit registers (3 sets) : {Q1, Q2, Q3, Q4}, {Q5, Q6, Q7, Q8}, {Q9, Q10, Q11, Q12}

Loop counter (for loop control) : LC

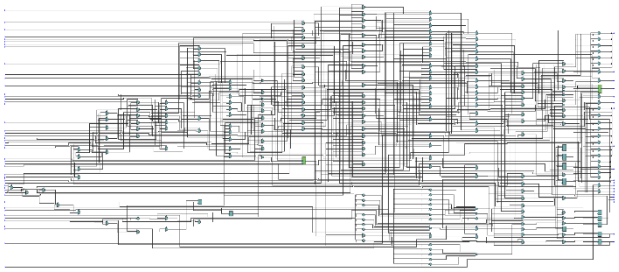
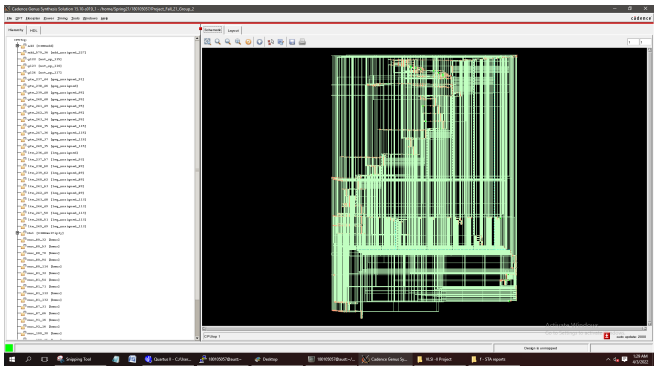


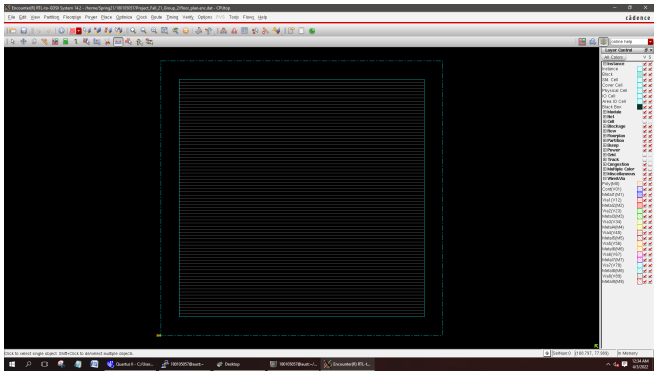
Figure 7: RTL View

2. SYNTHESIS RESULT

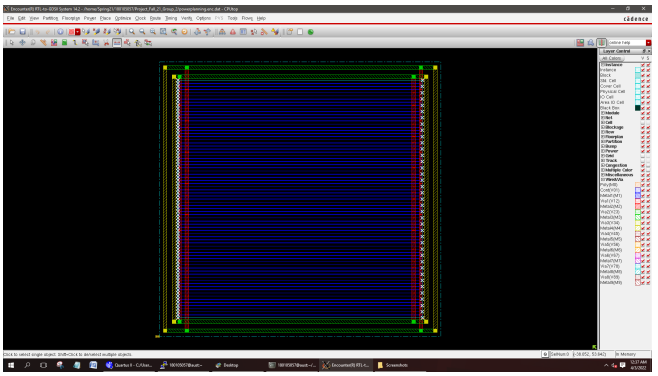


3. PNR OUTPUT

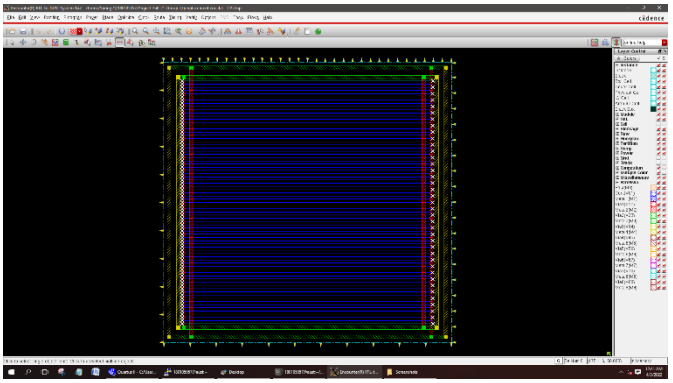
a. Floorplan



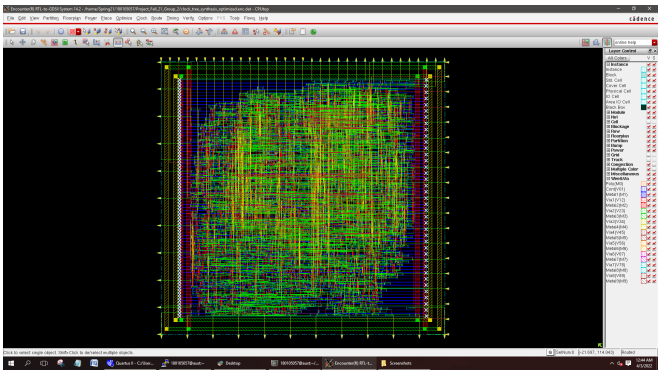
b. Power Plan



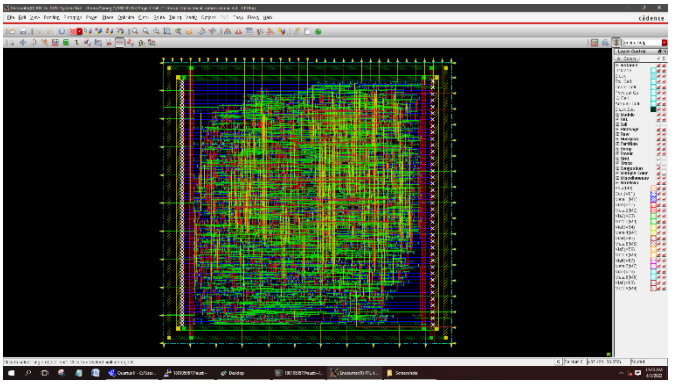
c. Pin Placement



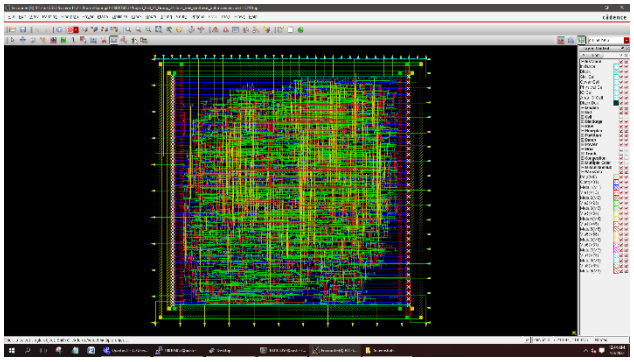
d. Placement



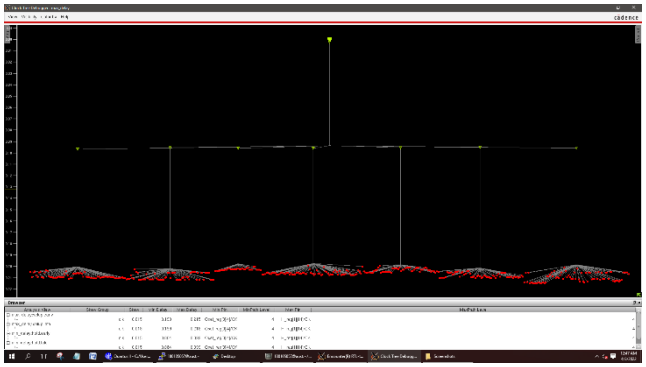
e. Placement Optimized



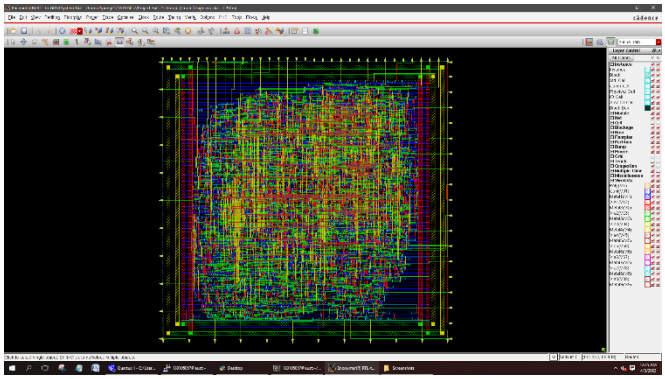
f. **After clock tree synthesis**



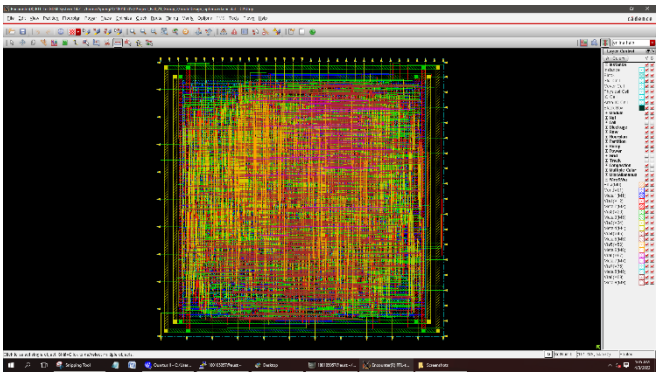
g. **Clock tree diagram**



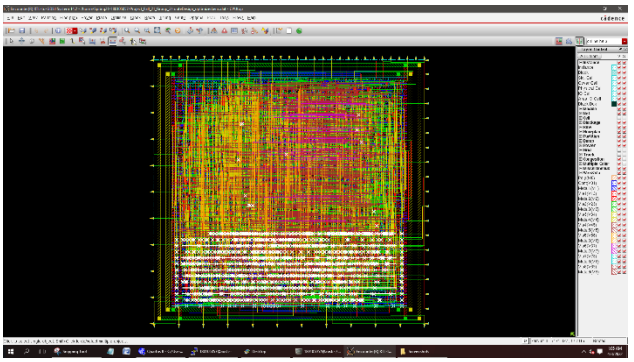
h. **Route design**



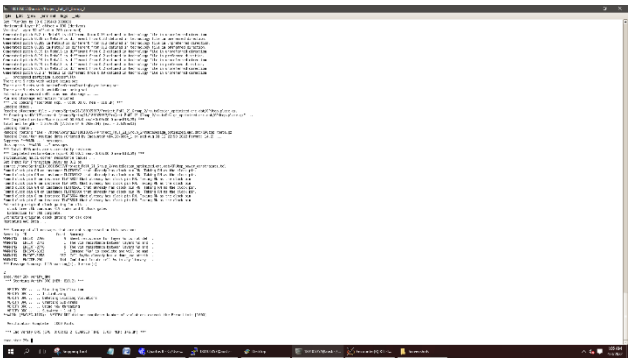
i. **Route design optimized**



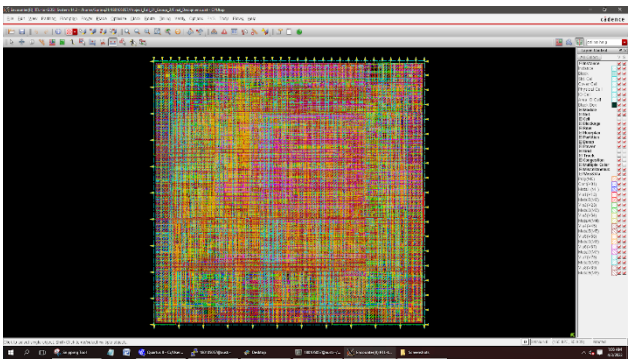
j. **DRC Error**



k. **DRC Error Violations**



l. **After Solving Violations and Final Design**



m. **Violation Check on Final Design**

```
encounter 32> verify_drc
 *** Starting Verify DRC (MEM: 831.7) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
  VERIFY DRC ..... Using new threading
  VERIFY DRC ..... Sub-Area : 1 of 1
  VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:00:01.0  ELAPSED TIME: 1.00  MEM: 146.1M) ***
```

```
encounter 37> verify_drc
 *** Starting Verify DRC (MEM: 922.4) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
  VERIFY DRC ..... Using new threading
  VERIFY DRC ..... Sub-Area : 1 of 1
**WARN: (ENCVFG-1103):   VERIFY DRC did not complete: Number of violations exceeds the Error Limit [1000]

  Verification Complete : 1000 Viols.

 *** End Verify DRC (CPU: 0:00:01.2  ELAPSED TIME: 2.00  MEM: 112.4M) ***
```

4. STA REPORTS BEFORE AND AFTER VIOLATION AND CLEANING.

A. STA REPORT BEFORE

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+--------+--------+--------+
|     Setup mode     |   all  | reg2reg | default |
+--------------------+--------+--------+--------+
|           WNS (ns):|  5.542 |  5.542 |  9.757 |
|           TNS (ns):|  0.000 |  0.000 |  0.000 |
|    Violating Paths:|    0   |    0   |    0   |
|          All Paths:|   707  |   707  |   563  |
+--------------------+--------+--------+--------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |     24 (175)     |   -0.359   |     24 (175)     |
|   max_fanout   |      0 (0)       |     0      |      1 (1)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 53.291%
Routing Overflow: 0.00% H and 0.00% V
```

```
------------------------------------------------------------
    optDesign Final Summary
------------------------------------------------------------

+--------------------+--------+--------+--------+
|     Setup mode     |   all  | reg2reg | default |
+--------------------+--------+--------+--------+
|           WNS (ns):|  5.472 |  5.472 | 10.088 |
|           TNS (ns):|  0.000 |  0.000 |  0.000 |
|    Violating Paths:|    0   |    0   |    0   |
|          All Paths:|   707  |   707  |   563  |
+--------------------+--------+--------+--------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |     77 (77)      |     -8     |     78 (78)      |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 53.637%
Routing Overflow: 0.00% H and 0.00% V
```

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+--------+--------+--------+
|     Setup mode     |   all  | reg2reg | default |
+--------------------+--------+--------+--------+
|           WNS (ns):|  5.654 |  5.654 | 10.029 |
|           TNS (ns):|  0.000 |  0.000 |  0.000 |
|    Violating Paths:|    0   |    0   |    0   |
|          All Paths:|   707  |   707  |   563  |
+--------------------+--------+--------+--------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |     24 (175)     |   -0.359   |     24 (175)     |
|   max_fanout   |     77 (77)      |     -8     |     84 (84)      |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 53.985%
Total number of glitch violations: 0
```

B. STA REPORT AFTER

```
------------------------------------------------------------
    optDesign Final Summary
------------------------------------------------------------

+--------------------+--------+--------+--------+
|     Setup mode     |   all  | reg2reg | default |
+--------------------+--------+--------+--------+
|           WNS (ns):|  5.472 |  5.472 | 10.088 |
|           TNS (ns):|  0.000 |  0.000 |  0.000 |
|    Violating Paths:|    0   |    0   |    0   |
|          All Paths:|   707  |   707  |   563  |
+--------------------+--------+--------+--------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |     77 (77)      |     -8     |     78 (78)      |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 53.637%
Routing Overflow: 0.00% H and 0.00% V
```

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+--------+--------+--------+
|     Setup mode     |   all  | reg2reg | default |
+--------------------+--------+--------+--------+
|           WNS (ns):|  5.654 |  5.654 | 10.029 |
|           TNS (ns):|  0.000 |  0.000 |  0.000 |
|    Violating Paths:|    0   |    0   |    0   |
|          All Paths:|   707  |   707  |   563  |
+--------------------+--------+--------+--------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |     77 (77)      |     -8     |     84 (84)      |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 53.985%
Total number of glitch violations: 0
```

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+--------+--------+--------+
|     Setup mode     |   all  | reg2reg | default |
+--------------------+--------+--------+--------+
|           WNS (ns):|  5.542 |  5.542 |  9.757 |
|           TNS (ns):|  0.000 |  0.000 |  0.000 |
|    Violating Paths:|    0   |    0   |    0   |
|          All Paths:|   707  |   707  |   563  |
+--------------------+--------+--------+--------+

+----------------+-------------------------------+------------------+
|                |              Real             |       Total      |
|    DRVs        +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |     24 (175)     |   -0.359   |     24 (175)     |
|   max_fanout   |      0 (0)       |     0      |      1 (1)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 53.291%
Routing Overflow: 0.00% H and 0.00% V
```

5. Violation (Before and After cleaning)

a. Before Cleaning

```
encounter 43> verify_connectivity
VERIFY_CONNECTIVITY use new engine.

******** Start: VERIFY CONNECTIVITY ********
Start Time: Thu Sep  1 13:56:46 2022

Design Name: CPUtop
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (154.7200, 150.0300)
Error Limit = 1000; Warning Limit = 50
Check all nets
**** 13:56:46 **** Processed 5000 nets.
Net VDD: dangling Wire.
Net VSS: dangling Wire.

Begin Summary
    77 Problem(s) (ENCVFC-94): The net has dangling wire(s).
    77 total info(s) created.
End Summary

End Time: Thu Sep  1 13:56:46 2022
Time Elapsed: 0:00:00.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 77 Viols.  0 Wrngs.
  (CPU Time: 0:00:00.2  MEM: 0.000M)
```

```
encounter 37> verify_drc
 *** Starting Verify DRC (MEM: 922.4) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
  VERIFY DRC ..... Using new threading
  VERIFY DRC ..... Sub-Area : 1 of 1
**WARN: (ENCVFG-1103):  VERIFY DRC did not complete: Number of violations exceeds the Error Limit [1000]

  Verification Complete : 1000 Viols.

 *** End Verify DRC (CPU: 0:00:01.2  ELAPSED TIME: 2.00  MEM: 112.4M) ***

verifyPowerDomain  verifyPowerSwitch  verifyPowerVia  verifyProcessAntenna
encounter 39> verifyProcessAntenna

******* START VERIFY ANTENNA ********
Report File: CPUtop.antenna.rpt
LEF Macro File: CPUtop.antenna.lef
5000 nets processed: 0 violations
Verification Complete: 0 Violations
******* DONE VERIFY ANTENNA ********
(CPU Time: 0:00:00.4  MEM: 0.000M)
```

```
encounter 38> verifyGeometry
 *** Starting Verify Geometry (MEM: 1026.0) ***

  VERIFY GEOMETRY ..... Starting Verification
  VERIFY GEOMETRY ..... Initializing
  VERIFY GEOMETRY ..... Deleting Existing Violations
  VERIFY GEOMETRY ..... Creating Sub-Areas
                  ..... bin size: 1920
  VERIFY GEOMETRY ..... SubArea : 1 of 1
**WARN: (ENCVFG-47):  This warning message means the PG pin of macro/macros is not connected to relevant PG net in the design. If we query the particular PG pin 'net:NULL' will be displayed in the Encounter GUI.

  VERIFY GEOMETRY ..... Cells          : 0 Viols.
  VERIFY GEOMETRY ..... SameNet        : 0 Viols.
  VERIFY GEOMETRY ..... Wiring         : 0 Viols.
  VERIFY GEOMETRY ..... Antenna        : 0 Viols.
  VERIFY GEOMETRY ..... Sub-Area : 1 complete 0 Viols. 0 Wrngs.
VG: elapsed time: 1.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 0
  Overlap     : 0
End Summary

  Verification Complete : 0 Viols.  0 Wrngs.
```

b. After Cleaning

```
encounter 32> verify_drc
 *** Starting Verify DRC (MEM: 831.7) ***

  VERIFY DRC ..... Starting Verification
  VERIFY DRC ..... Initializing
  VERIFY DRC ..... Deleting Existing Violations
  VERIFY DRC ..... Creating Sub-Areas
  VERIFY DRC ..... Using new threading
  VERIFY DRC ..... Sub-Area : 1 of 1
  VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.

  Verification Complete : 0 Viols.

 *** End Verify DRC (CPU: 0:00:01.0  ELAPSED TIME: 1.00  MEM: 146.1M) ***
```

```
encounter 33> verify_connectivity
VERIFY_CONNECTIVITY use new engine.

******** Start: VERIFY CONNECTIVITY ********
Start Time: Thu Sep  1 13:42:22 2022

Design Name: CPUtop
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (154.7200, 1
Error Limit = 1000; Warning Limit = 50
Check all nets
**** 13:42:23 **** Processed 5000 nets.

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Thu Sep  1 13:42:23 2022
Time Elapsed: 0:00:01.0
```

```
encounter 34> verifyGeometry
 *** Starting Verify Geometry (MEM: 979.8) ***

  VERIFY GEOMETRY ..... Starting Verification
  VERIFY GEOMETRY ..... Initializing
  VERIFY GEOMETRY ..... Deleting Existing Violations
  VERIFY GEOMETRY ..... Creating Sub-Areas
                  ..... bin size: 1920
  VERIFY GEOMETRY ..... SubArea : 1 of 1
  VERIFY GEOMETRY ..... Cells          :  0 Viols.
  VERIFY GEOMETRY ..... SameNet        :  0 Viols.
  VERIFY GEOMETRY ..... Wiring         :  0 Viols.
  VERIFY GEOMETRY ..... Antenna        :  0 Viols.
  VERIFY GEOMETRY ..... Sub-Area : 1 complete 0 Viols. 0 Wrngs.
VG: elapsed time: 3.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 0
  Overlap     : 0
End Summary

  Verification Complete : 0 Viols.  0 Wrngs.

encounter 35> verifyProcessAntenna

******* START VERIFY ANTENNA ********
Report File: CPUtop.antenna.rpt
LEF Macro File: CPUtop.antenna.lef
5000 nets processed: 0 violations
Verification Complete: 0 Violations
******* DONE VERIFY ANTENNA ********
(CPU Time: 0:00:00.4  MEM: 0.000M)
```

6. Final Design

```
encounter 41> checkPlace
**Info: (ENCSP-307): Design contains fractional 34 cells.
Begin checking placement ... (start mem=965.6M, init mem=965.6M)
*info: Placed = 7895           (Fixed = 8)
*info: Unplaced = 0
Placement Density:105.74%(17369/16427)
Finished checkPlace (cpu: total=0:00:00.1, vio checks=0:00:00.0; mem=965.6M)
0
```

7. Results

Name	Before Optimization	After Optimization
Final density	83.98%	105.74%
Placed cells	7895	7895
DRC Error	1000	0
Geometry Violation	0	0
Antena Violation	0	0
Connectivity violation	77	0
Gitch violation	33	0

8. Future Modifications

From the optimized time summary report of the final design, we can see the fanout violation remains unsolved even after the optimizations. So, this problem can be solved manually in the future. We might need to modify our design to optimize the glitch violations even more. Parallel memory structure allows great flexibility for programmers to perform data access of different block sizes and different word lengths. The configurable SIMD structure allows almost "random" register file access and slightly different operations in ALUs inside SIMD.

9. Conclusions

An SIMD processor was applied to the RC4 key search problem and was able to achieve a high level of parallelism as well as utilize the higher memory bandwidth available on the device. Although the design was slower than a hardwired design, the development time for an application using this machine is significantly lower since designers can adopt a purely programming-based model and need not be concerned with lower-level details such as datapath design, control design, place and route, design optimization, etc. The SIMD approach also has benefits in that the design can be amortized over many different applications potentially resulting in a large overall saving in the development effort. Finally, using this approach, there is potential to customize the instruction set of the processor as well as to add co-processing elements to further accelerate applications.

10. References

1) FPGA-based SIMD Processor Stanley Y.C. Li, Gap C.K. Cheuk, K.H. Lee and Philip H.W. Leong

2) SIMD Processor-Based Turbo Decoder Supporting Multiple Third-Generation Wireless Standards Myoung-Cheol Shin, Member, IEEE, and In-Cheol Park, Senior Member, IEEE

3) Xetal-Pro: An Ultra-Low Energy and High Throughput SIMD Processor Yifan He, Yu Pu Eindhoven University of Technology, the Netherlands {y.he, y.pu}@tue.nl Zhenyu Ye Eindhoven University of Technology, the Netherlands z.ye@tue.nl Sebastian M. Londono Eindhoven University of Technology, the Netherlands s.moreno@tue.nl Richard Kleihorst VITO, Belgium richard.kleihorst@vito.be Anteneh A. Abbo Philips Research, the Netherlands anteneh.a.abbo@philips.com Henk Corporaal Eindhoven University of Technology, the Netherlands h.corporaal@tue.nl

4) Simple SIMD Processor ——Report of Final Project Tingyuan LIANG (ID:20436607) Instructor: Professor Chi-Ying TSUI TA: Jingyang ZHU