Investigate the possibility of speeding-up the FPGA implementations of Skein in Xilinx Virtex 5 and Altera Stratix III by using:
   1) Fast combinational adders.
   2) Fast pipelined adders.

Your starting point is the existing implementation of Skein, developed by Cryptographic Engineering Research Group (CERG) @ GMU. This implementation is based on
   a) basic iterative architecture (one round per clock cycle),
   b) combinational Standard Carry Chain Adders (“+” in VHDL).

As a part of your investigation, please perform tasks described below:

**Part 1: Combinational Adders**
Progress report due Sunday, March 4, 2011, 11:59pm

**Task C1: Literature review**

Review
   a) lecture slides on fast adders,
   b) textbook by Parhami,
   c) research literature listed below,
   d) any other resources you can find on the web,

and determine which combinational adders (other than the Standard Carry Chain Adder), are likely to perform well in Xilinx Virtex 5 FPGAs and Altera Stratix III FPGAs.

**Literature:**


Task C2: VHDL coding

Implement by yourself at least two k-bit combinational adders identified as most promising in Task C1. Each adder should be described in VHDL in a generic form, and work correctly for at least the following values of k:

\[ k = 2^L \text{ for } L=3..12. \]

Please note that the value of k used in Skein is \( k=64=2^6 \).

All adders should be of the type used in Skein, i.e., they should have no carry in and no carry out. These adders perform addition mod \( 2^k \), which is equivalent to discarding carry out.

All adders should be thoroughly verified using functional simulation, and described in synthesizable VHDL.

Task C3: Implementation and benchmarking of adders as stand-alone units

Compare the throughputs and areas of all adders implemented in Task C2 with the throughput and area of SCCA (“+” in VHDL), for

- Xilinx Virtex 5 FPGAs,
- Altera Stratix III FPGAs.

Use \( k=64 \) and the smallest device of each FPGA family.

Throughput should be expressed in millions of additions per second (Madd/s). It should be measured separately for each adder, by surrounding a given adder with registers, and measuring the maximum clock frequency of the obtained circuit.

Area should be expressed in CLB slices for Virtex 5 and ALUTs for Stratix III. It should be measured separately for each adder, by determining the resource utilization without any surrounding registers. In case your implementation uses DSP units, please make sure to report their utilization as well.

You are encouraged to use ATHENa to

a) facilitate and speed-up collection of results,

b) optimize the choice of the FPGA tool options and the requested clock frequency.

All results should be obtained based on the post-place-and-route reports (or equivalent reports of ATHENa).

For each adder type, please report only results corresponding to the implementation that is best in terms of throughput.

Rank all investigated adders (including SCCA) in terms of the

a) throughput

b) area

c) throughput to area ratio

do the stand-alone adder implementation.
Please verify the operation of your two best adders at the maximum clock frequency using timing simulation for Xilinx Virtex 5.

**Task C4: Implementation and benchmarking of the modified Skein**

Modify the provided implementation of Skein in such a way that an adder type can be selected by a generic (of the integer type) of the top-level unit.

Make sure that your code supports all adder types implemented by you in Task C2.

Determine the throughput and area of the Skein implementations (after placing and routing) based on different combinational adder types.

**Throughput** should be expressed in $Mbits/s$. **Area** should be expressed in $CLB$ slices for Virtex 5 and $ALUT$s for Stratix III. In case your implementation uses DSP units, please make sure to report their utilization as well.

You are encouraged to use ATHENa to

a) facilitate and speed-up collection of results,

b) optimize the choice of the FPGA tool options and the requested clock frequency.

All results should be obtained based on the post-place-and-route reports (or equivalent reports of ATHENa).

For each adder type, please report only results corresponding to the implementation that is the best in terms of throughput.

Rank all investigated adders (including SCCA) in terms of the

a) throughput

b) area

c) throughput to area ratio

do the Skein implementations based on these adders.

---

**Part 2: Pipelined Adders**

**Progress report due Sunday, March 20, 2011, 11:59pm**

**Task P1: Literature Review**

Review

a) lecture slides on fast adders,

b) textbook by Parhami,

c) research literature listed below,

d) any other resources you can find on the web,

and determine which pipelined adders are likely to perform well in Xilinx Virtex 5 FPGAs and Altera Stratix III FPGAs.
Literature:


[FloPoCo] FloPoCo project web site at http://flopoco.gforge.inria.fr/

Task P2: VHDL coding

Implement by yourself at least two k-bit pipelined adders identified as most promising in Task P1.
Each adder should be described in VHDL in a generic form, and work correctly for at least the following values of k:
\[ k = 2^L \text{ for } L=3..12. \]
Please note that the value of k used in Skein is \( k=64=2^6 \).

A second generic, \( N \), should be used to identify the number of pipeline stages. The allowed values of this generic should be clearly specified in the entity declaration. For example, the allowed number of pipeline stages might be \( N=2, 3, 6 \) for \( k=64 \); any divisor of \( L=\log_2(k) \) for arbitrary \( k \); or any other combination of integers depending on the adder type.

All adders should be of the type used in Skein, i.e., they should have no carry in and no carry out. These adders perform addition mod \( 2^k \), which is equivalent to discarding carry out.

All adders should be thoroughly verified using functional simulation, and described in synthesizable VHDL.

Task P3: Implementation and benchmarking of adders as stand-alone units

Compare the throughputs and areas of all adders implemented in Task P2 with the throughput and area of SCCA (“+” in VHDL), for
- Xilinx Virtex 5 FPGAs,
- Altera Stratix III FPGAs.
Use \( k=64 \) and the smallest device of each FPGA family.

Throughput should be expressed in millions of additions per second (Madd/s). It should be measured separately for each adder, by surrounding a given adder with registers, and measuring the maximum clock frequency of the obtained circuit.
Area should be expressed in CLB slices for Virtex 5 and ALUTs for Stratix III. It should be measured separately for each adder, by determining the resource utilization without any surrounding registers. In case your implementation uses DSP units, please make sure to report their utilization as well.

For each adder type, investigate the influence of the number of pipeline stages, N, on the obtained results. Do it by repeating your implementation for all values of N supported by a given type of adder for k=64.

You are encouraged to use ATHENa to
   a) facilitate and speed-up collection of results
   b) optimize the choice of the FPGA tool options and the requested clock frequency.

All results should be obtained based on the post-place-and-route reports (or equivalent reports of ATHENa).

For each adder type, please report only results corresponding to the implementation that is the best in terms of throughput.

Rank all investigated adders (including SCCA) in terms of the
   a) throughput
   b) area
   c) throughput to area ratio

of the stand-alone adder implementations.

Please verify the operation of your two best adders at the maximum clock frequency using timing simulation for Xilinx Virtex 5.

Task P4 (bonus)

Repeat Task P3 for the VHDL codes of the following pipelined adders:
   a) Classical
   b) Alternative
   c) Short-Latency,

obtained using the FloPoCo generator of arithmetic cores downloaded from the FloPoCo project web site at http://flopoco.gforge.inria.fr/

Task P5 (bonus)

Revise the provided implementation of Skein, in such a way that this implementation can process two (or more) independent messages at the same time, and benefit from using pipeline adders with two (or more) pipeline stages.

In order to do that, you may need to modify the circuit:
   a) interface
   b) block diagram
   c) ASM chart
   d) VHDL code
You will receive partial credit for completing correctly any of the above steps.

Determine the throughput and area of the revised implementation and compare it with the throughput and area of the basic implementation provided to you.

**Deliverables**

1. Comprehensive report summarizing all your
   - Assumptions (beyond those given in the specification)
   - Choices
   - Methodology used
   - Status of your codes
   - Results
   - Conclusions
   - Encountered problems.
2. Synthesizable source codes for all tasks except C1 and P1.
3. Testbenches used in all tasks except C1 and P1.
4. ATHENA reports or a list of non-default options of tools (including requested clock frequencies) for tasks C3-C4, P3-P5.

**Contest**

The results obtained for functionally correct codes will be eligible for participation in the contest for the:

a. fastest 64-bit combinational adder (Task C3)
   b. fastest implementation of Skein based on combinational adders (Task C4)
   c. fastest 64-bit pipelined adder (Tasks P3 & P4)
   d. fastest implementation of Skein based on pipelined adders (Task P5).

The winners in each category, separately for Xilinx Virtex 5 and Altera Stratix III, will receive bonus points.

**Important Note:** The GMU honor code will be strictly enforced. Any attempt to present somebody’s else work as your own (including project solutions from previous years, codes downloaded from the web, codes developed by a fellow student, etc.) may result in the zero grade for the entire project, and a substantial reduction in the final course grade.