Comparison of FPGA-Targeted Hardware Implementations of eSTREAM Stream Cipher Candidates

David Hwang, Mark Chaney, Shashi Karanam, Nick Ton, Kris Gaj

Dept. of Electrical and Computer Engineering
George Mason University, Fairfax, Virginia

GEORGE MASON UNIVERSITY

Outline

- **Purpose and Approach**
- **Cipher Specification**
- **Optimizations**
  - General Optimizations
  - Cipher-Specific Optimizations
- **Results**
- **Conclusions**

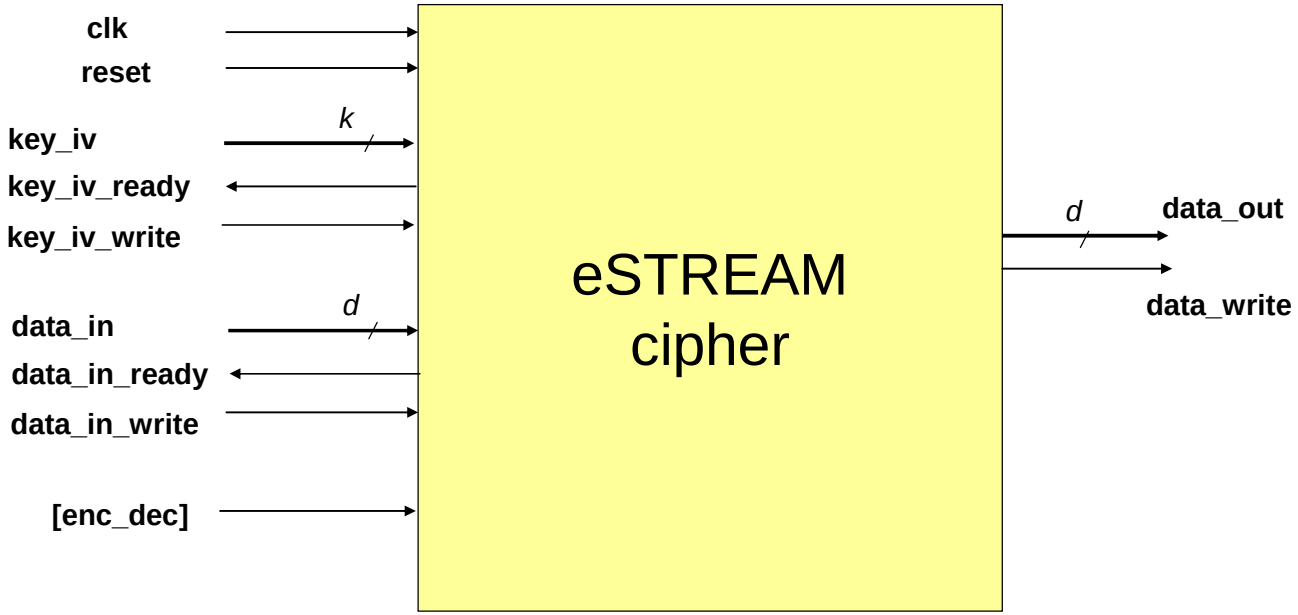
Purpose

- **Hardware comparison of all phase-3 eSTREAM profile 2 candidates**
- **Targeted for Xilinx Spartan-3 FPGAs**
  - Low-cost, high-volume applications
  - 90 nm process node
- **Implementations optimized taking into account Spartan-3 architectural features, particularly the shift register (SRL16)**

Approach

- **Part of a graduate class at George Mason University:**
  - ECE 545: Introduction to VHDL
- **Process:**
  - Each cipher variant implemented independently by at least 3 students
  - Design path: specification, algorithmic state machine (datapath vs. controller), VHDL coding, synthesis, FPGA implementation
  - Best implementation chosen and optimized (or recoded) by the authors
  - All ciphers verified post place-and-route against C test vectors on eSTREAM website
- **Tools**
  - VHDL Simulation: Aldec Active HDL 7.2, ModelSim 6.3a
  - FPGA Synthesis: Synplify Pro 8.6, Xilinx XST 9.1
  - FPGA Implementation: Xilinx ISE 9.1

Specification and Interface



- **d is the data radix**
  - d = bits/cycle of cipher input/output
  - Each cipher has a "natural value" of d
- **k is the key radix**
  - k = bits/cycle of the key_iv interface
  - Larger k → faster programming with larger area
  - Smaller k → slower programming with smaller area
  - Set k = d in all experiments

Ciphers Implemented

Candidate	Key Size (bits)	Max IV Size (bits)	Data radix d (bits/cycle)	Separate key register	Separate IV register
DECIM v2	80	64	0.25	no	yes
DECIM 128	128	128	0.25	no	no
Edon80	80	64	1	yes	yes
F-FCSR-H v2	80	80	8	no	no
F-FCSR-16	128	128	16	no	no
Grain v1	80	64	1	no	no
Grain 128	128	96	1	no	no
MICKEY 2.0	80	80	1	no	no
MICKEY 128 2.0	128	128	1	no	no
Moustique	96	104	1	yes	no
Pomaranch	80 /128	108 /162	1	yes	yes
Trivium	80	80	1	no	no

Optimizations General

- **All designs were optimized for the metrics:**
  - Throughput/area
  - Minimum area
- **General optimizations**
  - Pipelining and retiming
  - State-Machine Encoding (one-hot, gray, etc.)
  - Clock FSM outputs
  - CAD tool options and flags
- **Particular optimizations targeted to eSTREAM ciphers**
  - Not requiring separate key and iv registers (must reprogram entire key and iv to change either one)
  - Implementing LFSRs, and NFSRs in Xilinx SRL16 registers

Xilinx Shift Register

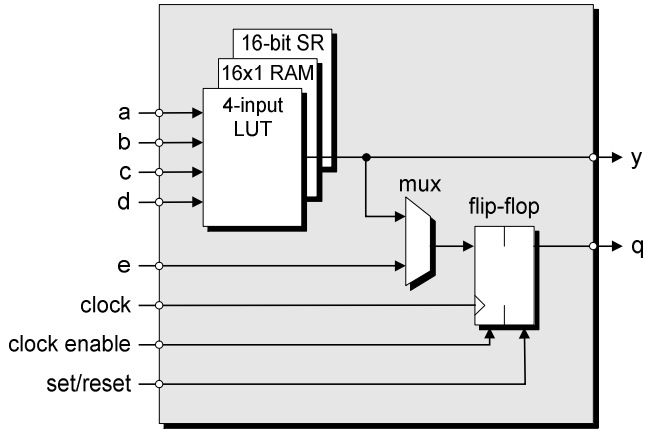


Table 7-1: **Shift Register Primitives**

Primitive	Length	Control	Address Inputs	Output
SRL16	16 bits	CLK	A3, A2, A1, A0	Q
SRL16E	16 bits	CLK, CE	A3, A2, A1, A0	Q
SRL16_1	16 bits	CLK	A3, A2, A1, A0	Q
SRL16E_1	16 bits	$\overline{\text{CLK}}$, CE	A3, A2, A1, A0	Q
SRLC16	16 bits	CLK	A3, A2, A1, A0	Q, Q15
SRLC16E	16 bits	CLK, CE	A3, A2, A1, A0	Q, Q15
SRLC16_1	16 bits	CLK	A3, A2, A1, A0	Q, Q15
SRLC16E_1	16 bits	$\overline{\text{CLK}}$, CE	A3, A2, A1, A0	Q, Q15

- **LUT can be configured as a 16-bit shift register (SRL16)**
  - 32 bit shift register normally requires 16 slices
  - Using SRL16 requires only 2 slices
- **Useful for modeling LFSRs and NFSRs,**
  - Used in DECIM, Grain, and Trivium
- **Caveats:**
  - No reset allowed
  - Must break SRL16 into smaller lengths to access internal values
    - Not useful for parallel lookahead versions of Grain and Trivium
  - Cannot use if put combinational logic between registers
    - FFCSR, Mickey fall into this category

Cipher-Specific Optimizations

- **DECIM**
  - Composed of LFSR feeding data to a Boolean F, feeds a decimation block ABSG
  - One output per four clock cycles
  - Optimizations: no key storage, small (or no) IV storage, LFSR in shift registers, buffer in shift registers
- **Edon80**
  - Binary additive cipher with pipelined stages of 80 2-bit etransformers
  - Can be implemented iterative (Kasper) or pipeline (our)
  - Optimizations: quasigroup operations as ROM tables
- **F-FCSR**
  - High-radix ciphers: d=8 bits/cycle (F-FCSR-H), d=16 bits/cycle (F-FCSR-16)
  - Core of ciphers is the Feedback with Carry Shift Register (FCSR), resembling a shift register with a full adders between flops
  - Optimizations: no key storage, no ive storage, re-use of internal registers

Cipher-Specific Optimizations

- **Grain**
  - Simple structure with LFSR and NFSR
  - Can also implement lookahead version: d=16 bits/cycle for Grain v1 and d=32 bits/cycle for Grain 128
  - Optimizations: LFSR/NFSR inferred in SRL16, no key or iv storage
- **MICKEY**
  - Usually a "linear" R register and "non-linear" S register
  - R and S registers have logic between flops, no shift register inferred
  - Optimizations: no key and iv storage, reducing logic on compare signals
- **Moustique**
  - Self-synchronizing cipher with encrypt/decrypt modes
  - Conditional complementing shift register
  - Optimizations: no iv storage

Cipher-Specific Optimizations

- **Pomaranch**
  - Consists of jump registers whose outputs cascade from one register to another, controller configuration of flip-flops
  - S-Box can be implemented with lookup tables or using composite fields and Boolean operations
  - Optimizations: implementation of both S-Box configurations
- **Trivium**
  - Simple structure with shift register and basic gates (xor, and)
  - Can implement higher radix (d = 64 bits/cycle) version
  - Optimizations: no key and iv storage, infer SRL16 blocks

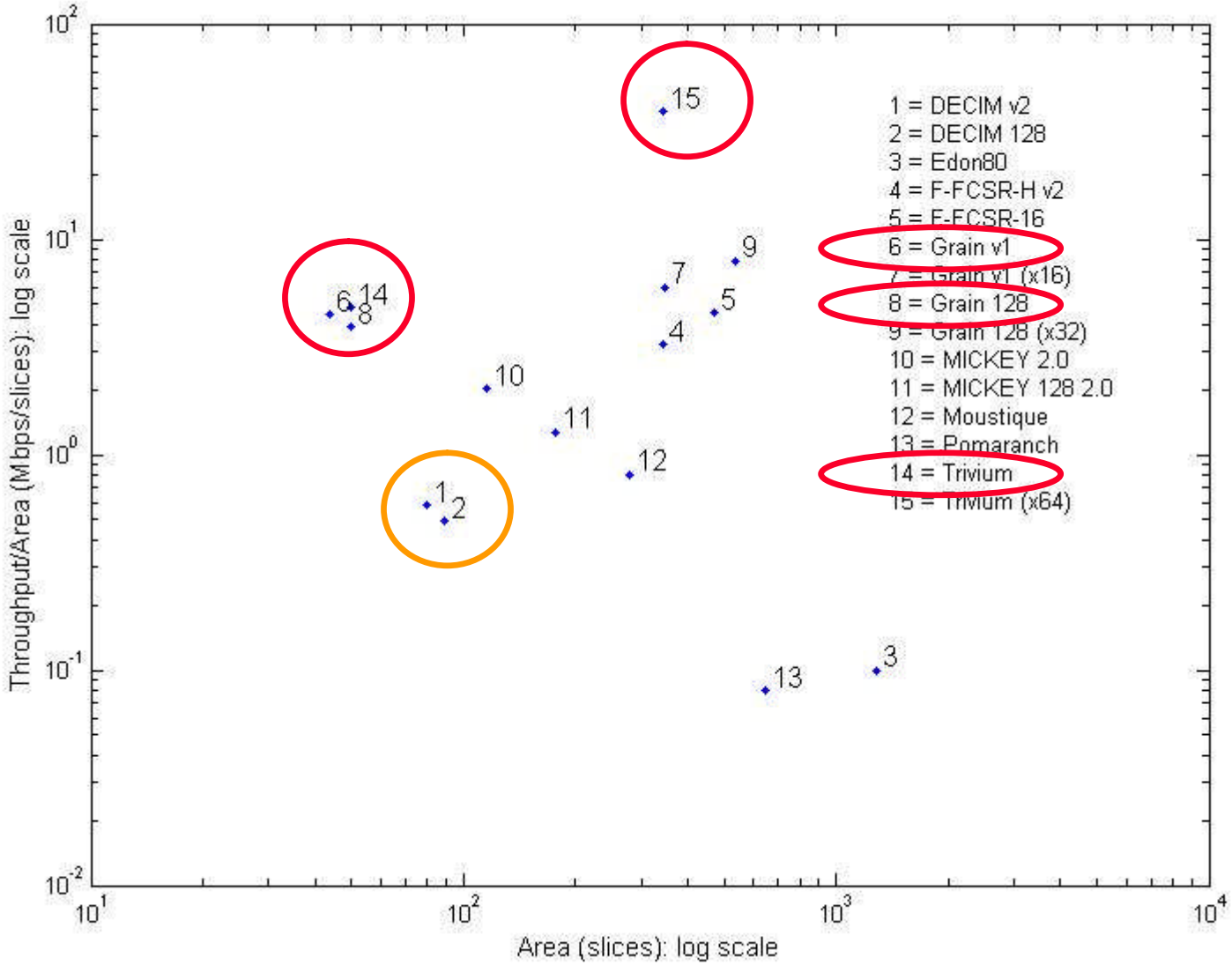
Results

Candidate	Data rate (bits/ cycle)	Maximum Clock Frequency (MHz)	Maximum Throughput (Mbps)	Area (slices)	Throughput/ Area (Mbps / slices)	Device
DECIM v2	0.25	185	46.25	80	0.58	xc3s50-5pq208
DECIM 128	0.25	174	43.5	89	0.49	xc3s50-5pq208
Edon80	1	130	130	1284	0.10	xc3s200-5pq208
F-FCSR-H v2	8	138	1104	342	3.23	xc3s50-5pq208
F-FCSR-16	16	134	2144	473	4.53	xc3s50-5pq208
Grain v1	1	196	196	44	4.45	xc3s50-5pq208
Grain v1 (x16)	16	130	2080	348	5.98	xc3s50-5pq208
Grain 128	1	196	196	50	3.92	xc3s50-5pq208
Grain 128 (x32)	32	133	4256	534	7.97	xc3s50-5pq208
MICKEY 2.0	1	233	233	115	2.03	xc3s50-5pq208
MICKEY 128 2.0	1	223	223	176	1.27	xc3s50-5pq208
Moustique	1	225	225	278	0.81	xc3s50-5pq208
Pomaranch	1	49	49	648	0.08	xc3s50-5pq208
Trivium	1	240	240	50	4.80	xc3s50-5pq208
Trivium (x64)	64	211	13504	344	39.26	xc3s400-5fg320

Results Sorted by Minimum Area and Maximum Throughput/Area

Candidate	Area (slices)	Candidate	Throughput/Area (Mbps/slices)
Grain v1	44	Trivium (x64)	39.26
Grain 128	50	Grain 128 (x32)	7.97
Trivium	50	Grain v1 (x16)	5.98
DECIM v2	80	Trivium	4.80
DECIM 128	89	F-FCSR-16	4.53
MICKEY 2.0	115	Grain v1	4.45
MICKEY 128 2.0	176	Grain 128	3.92
Moustique	278	F-FCSR-H v2	3.23
F-FCSR-H v2	342	MICKEY 2.0	2.03
Trivium (x64)	344	MICKEY 128 2.0	1.27
Grain v1 (x16)	348	Moustique	0.81
F-FCSR-16	473	DECIM v2	0.58
Grain 128 (x32)	534	DECIM 128	0.49
Pomaranch	648	Edon80	0.10
Edon80	1284	Pomaranch	0.08

Throughput / Area vs. Area



Comparison to Prior Art

Candidate	Maximum Clock Frequency (MHz)	Maximum Throughput (Mbps)	Area (slices)	Throughput/ Area (Mbps / slices)	Device
Edon80 [3]	149	1.87	50	0.04	Spartan-3
Grain v1 [12]	-	105	48	2.19	Spartan-II
Grain v1 [8]	193	193	122	1.58	Spartan-3
Grain v1 (x16) [8]	155	2480	356	6.97	Spartan-3
Grain 128 [10]	181	181	48	3.77	Virtex-II
MICKEY 128 2.0 [15]	170	170	167	1.02	Virtex
MICKEY 128 2.0 [10]	200	200	190	1.05	Virtex-II
MICKEY 128 2.0 [8]	156	156	261	0.60	Spartan-3
Moustique [16]	-	369	252	1.46	Virtex-II
Mosquito [12]	-	137	298	0.46	Spartan-II
Trivium [10]	207	207	41	5.05	Virtex-II
Trivium [12]	-	102	40	2.55	Spartan-II
Trivium [8]	201	201	188	1.07	Spartan-3
Trivium (x64) [8]	190	12160	388	31.34	Spartan-3

Conclusion

- **Assuming mathematically equivalent security, Grain and Trivium most efficient implementations on Xilinx FPGAs**
  - Minimum area
  - Maximum throughput/area
  - Use of SLR16 helped to decrease area
- **MICKEY showed good balance between low area and good throughput/area**
- **Implementation remarks:**
  - Ciphers that have long shift registers with no external signaling or combinational logic are compact in Xilinx FPGAs
  - This compactness may not port directly to ASIC technologies