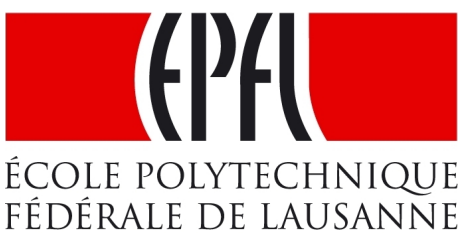
Fast Software AES Encryption

Dag Arne Osvik[1] Joppe W. Bos[1] Deian Stefan[2] David Canright[3]

[1]Laboratory for Cryptologic Algorithms, EPFL, CH-1015 Lausanne, Switzerland
[2]Dept. of Electrical Engineering, The Cooper Union, NY 10003, New York, USA
[3]Applied Math., Naval Postgraduate School, Monterey CA 93943, USA

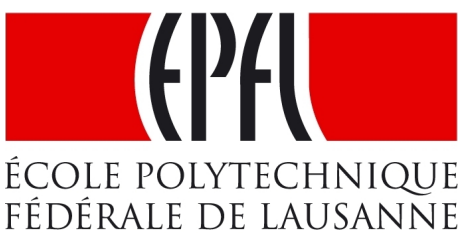


Fast Software AES Encryption

Dag Arne Osvik[1] Joppe W. Bos[1] Deian Stefan[2] David Canright[3]

Presented by: Onur Özen[1]

[1]Laboratory for Cryptologic Algorithms, EPFL, CH-1015 Lausanne, Switzerland
[2]Dept. of Electrical Engineering, The Cooper Union, NY 10003, New York, USA
[3]Applied Math., Naval Postgraduate School, Monterey CA 93943, USA



Outline

- Introduction
  - Motivation
  - Related Work
  - Contributions
- The Advanced Encryption Standard
- Target Platforms
  - The 8-bit AVR Microcontroller
  - The 32-bit Advanced RISC Machine
  - The Cell Broadband Engine Architecture
  - The NVIDIA Graphics Processing Unit
- Conclusions

Motivation

Advanced Encryption Standard

- Rijndael announced in 2001 as the AES.
- One of the most widely used cryptographic primitives.
  - IP Security, Secure Shell, Truecrypt
  - RFID and low-power authentication methods
  - Key tokens, RF-based Remote Access Control
- Many intensive efforts to speed up AES in both hard- and software.

Related work

E. Käsper and P. Schwabe. *Faster and Timing-Attack Resistant AES-GCM.* CHES 2009.

P. Bulens, et al. *Implementation of the AES-128 on Virtex-5 FPGAs* AFRICACRYPT 2008.

O. Harrison and J. Waldron. *Practical Symmetric Key Cryptography on Modern Graphics Hardware.* USENIX Sec. Symp. 2008.

S. Rinne, et al. *Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers.* SPEED 2007.

K. Shimizu, et al. *Cell Broadband Engine Support for Privacy, Security, and Digital Rights Management Applications.* 2005.

Contributions

New software speed records for various architectures

- Target both ends of the performance spectrum
  - low-end microcontrollers
  - high-end architectures processing many streams simultaneously
- Specific optimizations for each platform

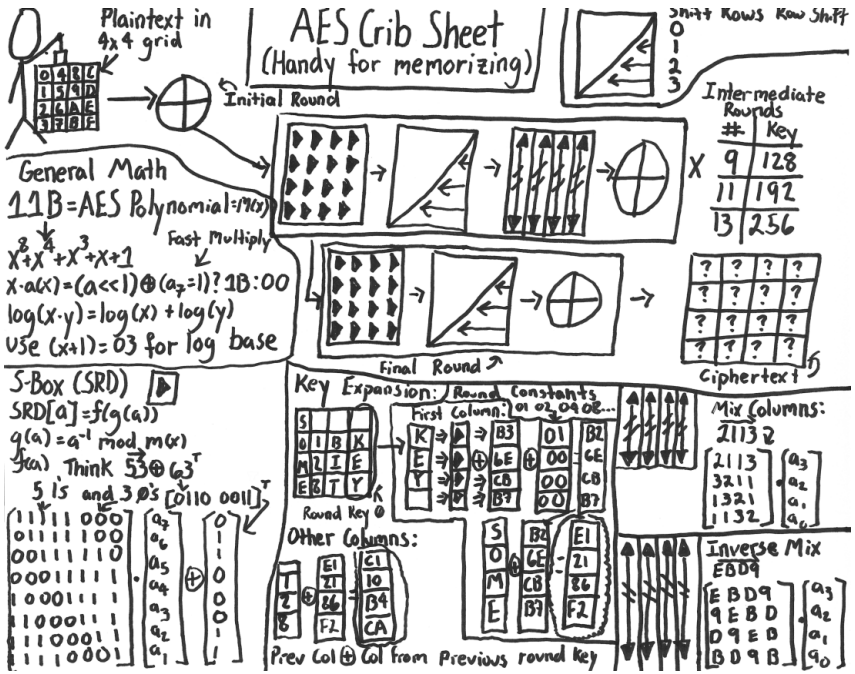
Contributions

New software speed records for various architectures

- 8-bit AVR microcontrollers
  - compact, efficient single stream AES version
- 32-bit Advanced RISC Machine
  - one of the most widely used processors for mobile applications
  - efficient single stream AES version (T-table based)
- Synergistic processing elements of the Cell broadband engine
  - widely available in the PS3 video game console
  - single instruction multiple data (SIMD) architecture
  - process 16 streams in parallel (bytesliced)
- NVIDIA graphics processing unit
  - first AES decryption implementation
  - single instruction multiple threads (SIMT) architecture
  - process thousand of streams in parallel (T-table based)

The Advanced Encryption Standard

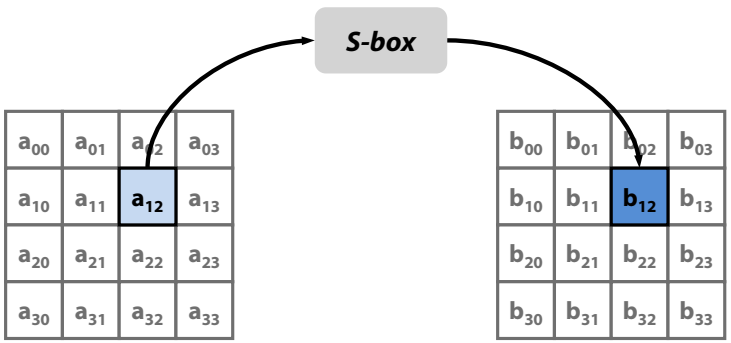
- Fixed block length version of the Rijndael block cipher
- Key-iterated block cipher with 128-bit state and block length
- Support for 128-, 192-, and 256-bit keys
- Strong security properties
  - only related key attacks
  - on AES-192 and AES-256
  - no attacks on full AES-128
- Very efficient in hardware and software



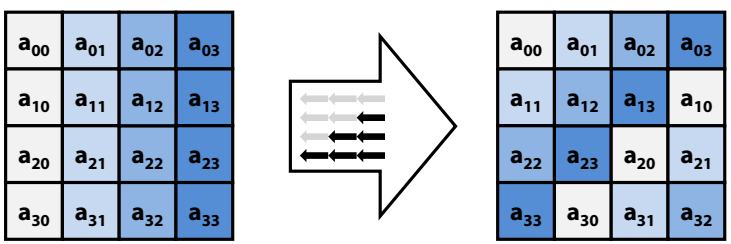
(Created by Jeff Moser)

Round Function Steps

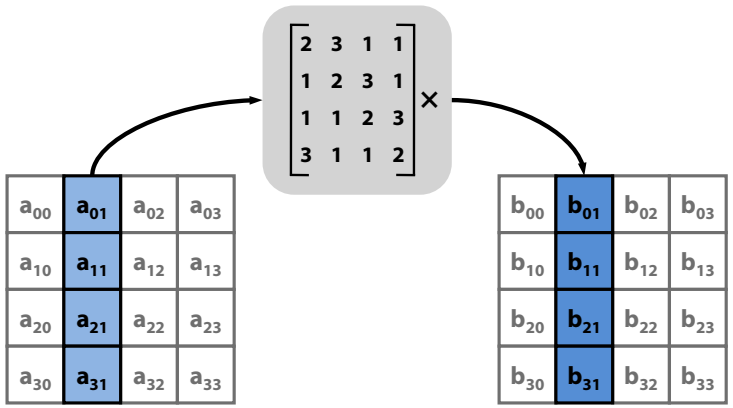
1 SubBytes:



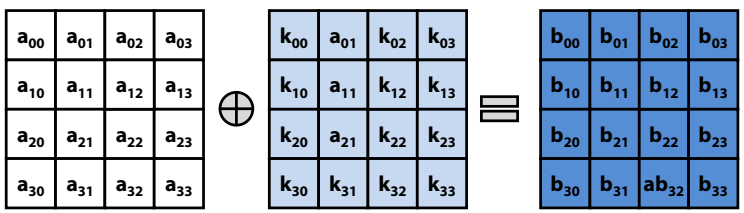
2 ShiftRows:



3 MixColumns:



4 AddRoundKey:

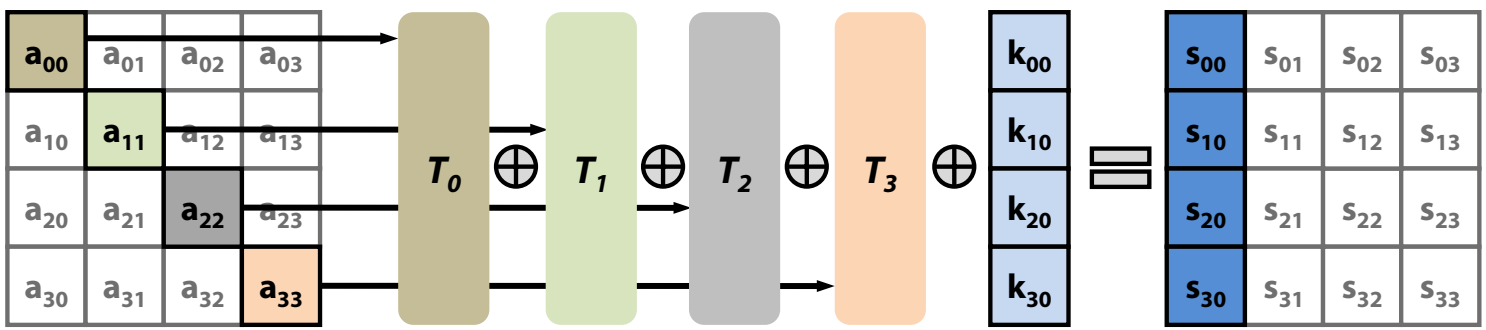


Efficient T-table implementation

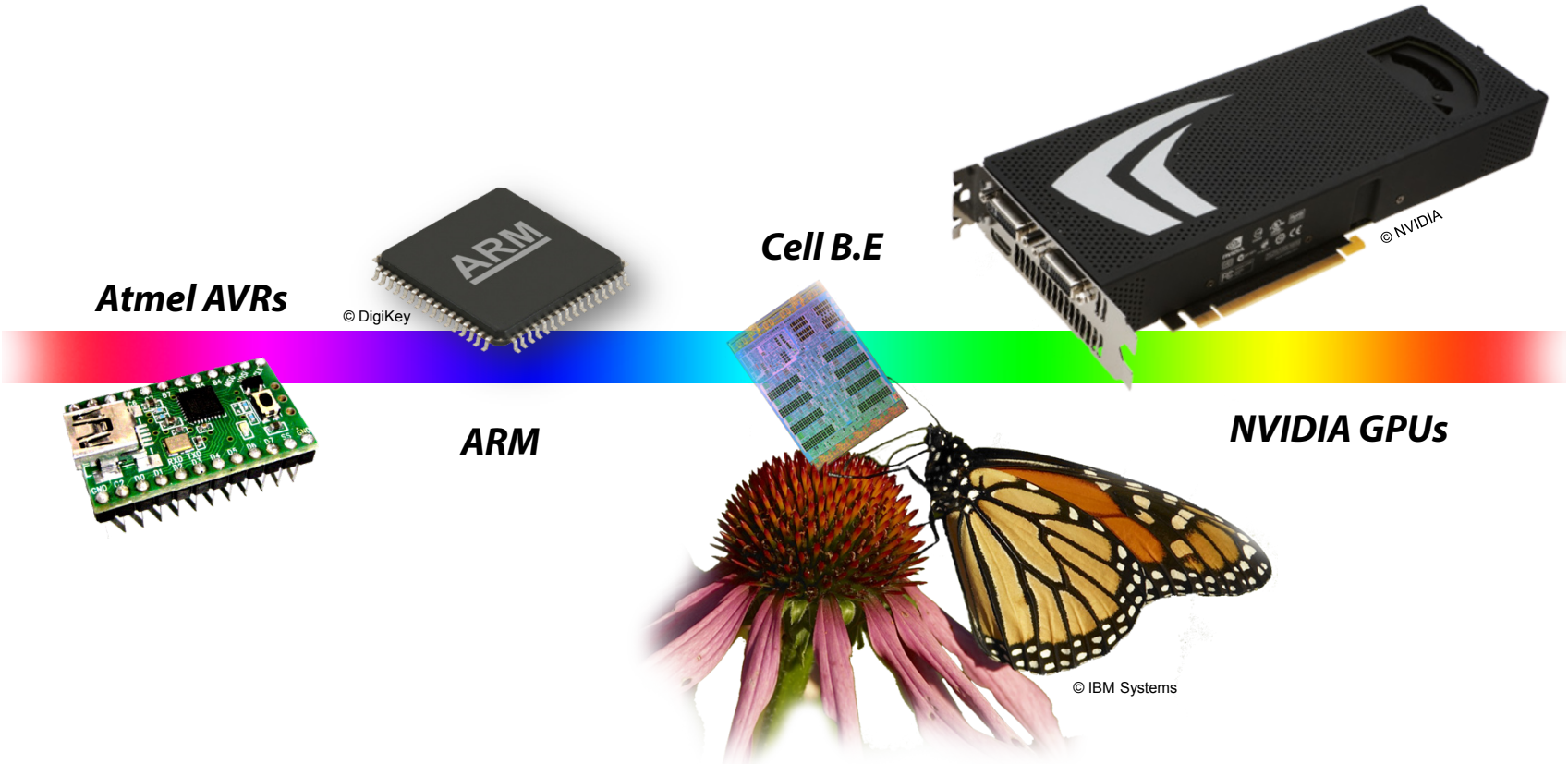
- Combine `SubBytes`, `ShiftRows`, `MixColumns` using the standard "T-table" approach. Update each column ($0 \leq j \leq 3$):

$$[s_{j0}, s_{j1}, s_{j2}, s_{j3}]^{\mathrm{T}} = T_0[a_{c_0 0}] \oplus T_1[a_{c_1 1}] \oplus T_2[a_{c_2 2}] \oplus T_3[a_{c_3 3}] \oplus k_j,$$

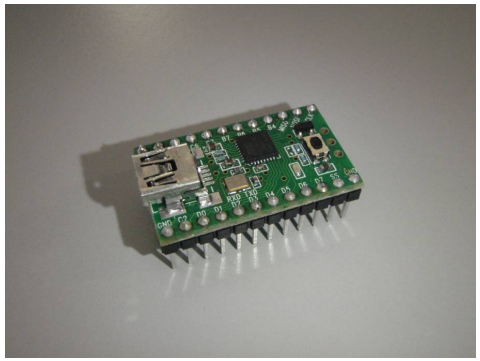
  where each T_i is $1KB$ and k_j is the jth column of the round key.
- T_i's are rotations of one table.
- Example ($j = 0$):



Target Platforms

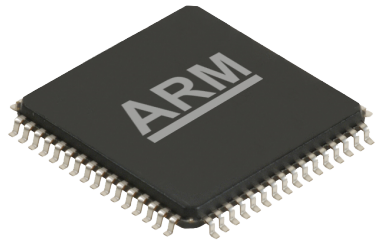


Microcontrollers



AVR

- Modified Harvard architecture
- 32 · 8-bit registers
- 3 · 16-bit pointer registers
- Registers are addressable
- Mostly single-cycle execution
- $\frac{1}{2}$ to 384KB flash memory
- 0 to 32KB SRAM
- 0 to 4KB EEPROM



ARM

- Harvard/von Neumann architecture
- 16 · 32-bit registers available
- Conditional execution
- Optional modification of flags
- Mostly single-cycle execution
- Load/store multiple registers
- Inline barrel shifter

Microcontrollers – Results

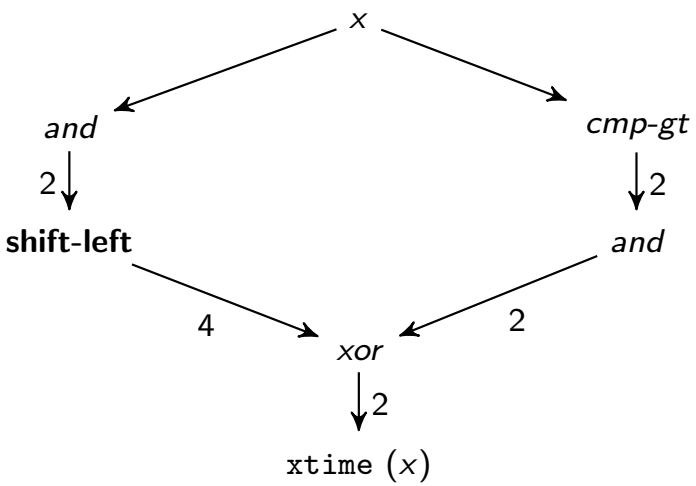
Reference	Key scheduling	Encryption (cycles)	Decryption (cycles)	Code size (bytes)
8-bit AVR microcontroller				
Rinne et al.	precompute	3,766	4,558	3,410
Poettering (Fast)	precompute	2,474	3,411	3,098
Poettering (Furious)	precompute	2,739	3,579	1,570
Poettering (Fantastic)	precompute	4,059	4,675	1,482
Otte	precompute	2,555	6,764	2,070
Otte	precompute	2,555	3,193	2,580
New - Low RAM	precompute	2,153	2,901	1,912
New - Fast	precompute	1,993	2,901	1,912
32-bit ARM microprocessor				
Atasu et. al	precompute	639	638	5,966
New	precompute	544	-	3,292

Cell Broadband Engine Architecture

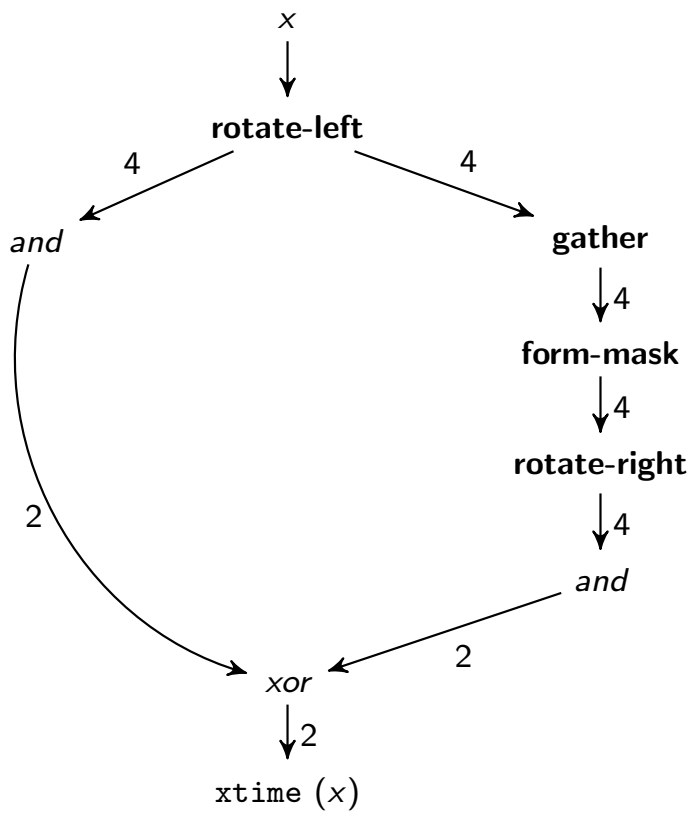
- Use the Synergistic Processing Elements
  - runs at 3.2 GHz
  - 128-bit wide SIMD-architecture
  - two instructions per clock cycle (dual pipeline)
  - in-order processor
  - rich instruction set: i.e.
    all distinct binary operations
    $f : \{0,1\}^2 \to \{0,1\}$ are present.
- "Expensive" QS22 Blade Servers (2×8 SPEs)
- PCI express cards:
  cryptographic accelerator ($\{$ 4, 8 $\}$ SPEs)
- "Cheap" PS3 video game console (6 SPEs)

Dual pipeline – Example

Balancing pipeline loads may lead to more instructions but fewer cycles.
16-way SIMD `xtime`: $a \cdot x \quad (a \in \mathbb{F}_{2^8} \cong \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1))$
Similar techniques for 16-way SIMD S-box lookups.

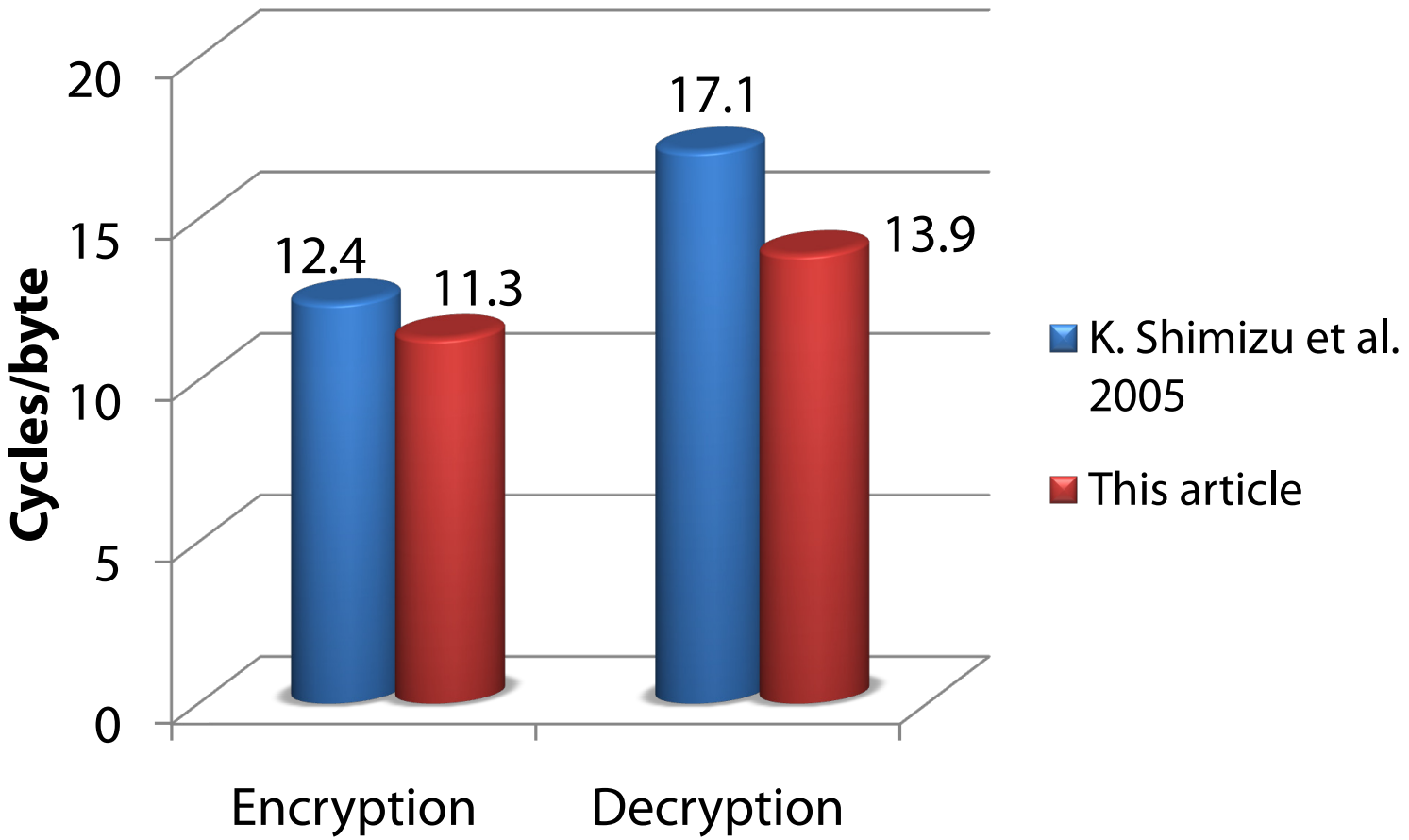


4 even, 1 odd
latency: 8 cycles



3 even, 4 odd
latency: 20 cycles

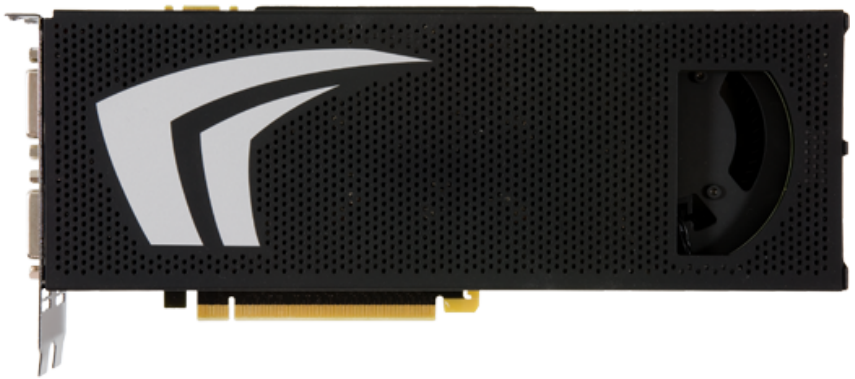
SPU Results Comparison

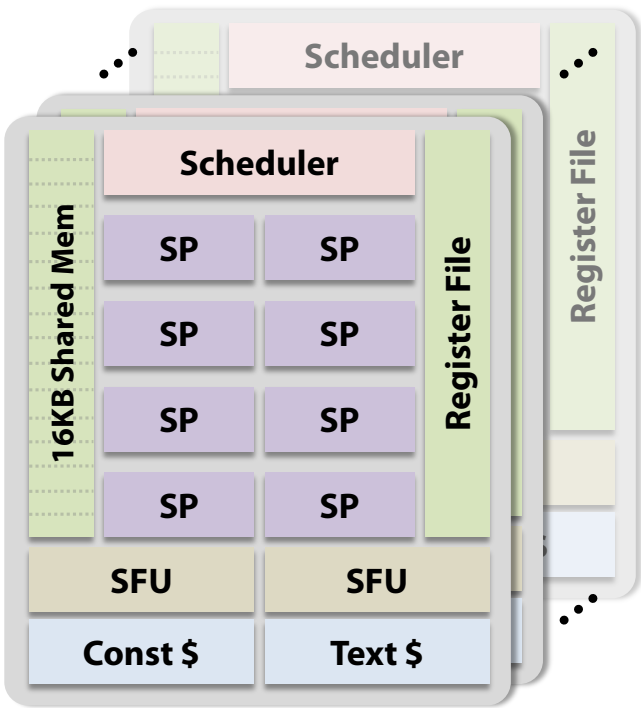


Throughput per PS3: **13.8** (encryption) and **10.8 Gbps** (decryption)

NVIDIA Graphic Processing Units

- Contain 12-30 *simultaneous multiprocessors* (SMs):
  - 8 streaming processors (SPs)
  - 16KB 16-way banked *fast* shared memory
  - 8192/16384 32-bit registers
  - 8KB constant memory cache
  - 6KB-8KB texture cache
  - 2 special function units
  - instruction fetch and scheduling unit
- GeForce 8800GTX:
  16 SMs @ 1.35GHz
- GTX 295:
  2×30 SMs @ 1.24GHz



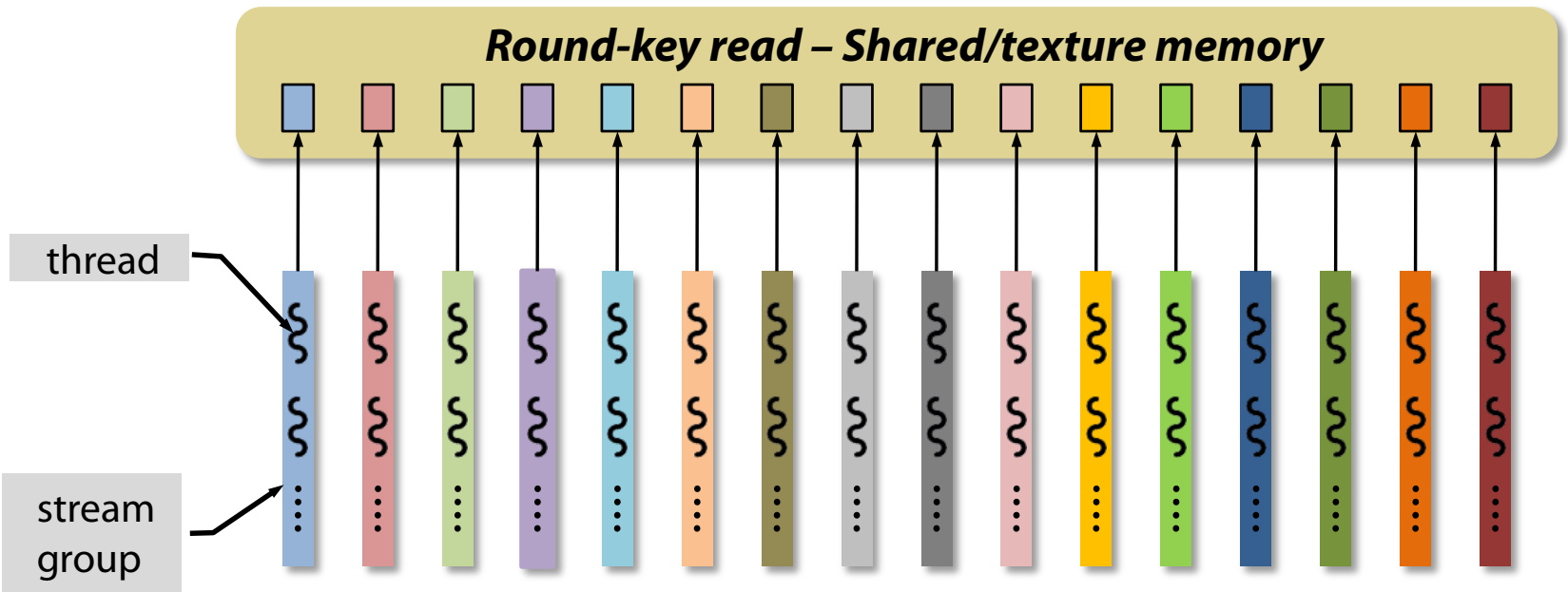


AES GPU Implementation

- Each thread processes multiple blocks
  $\longrightarrow$ hide memory latency using double buffering!
- Group multiple (e.g., 16) threads into *stream groups*
  $\longrightarrow$ share a common key, expanded in texture/shared memory
- Execute 16 (scheduling-unit) stream groups per *thread block*
  $\longrightarrow$ allows any caching to shared memory with no collisions!
- Launch multiple thread blocks per grid
  $\longrightarrow$ increase device utilization!
- All global memory access is coalesced
  $\longrightarrow$ maximize memory access throughput!

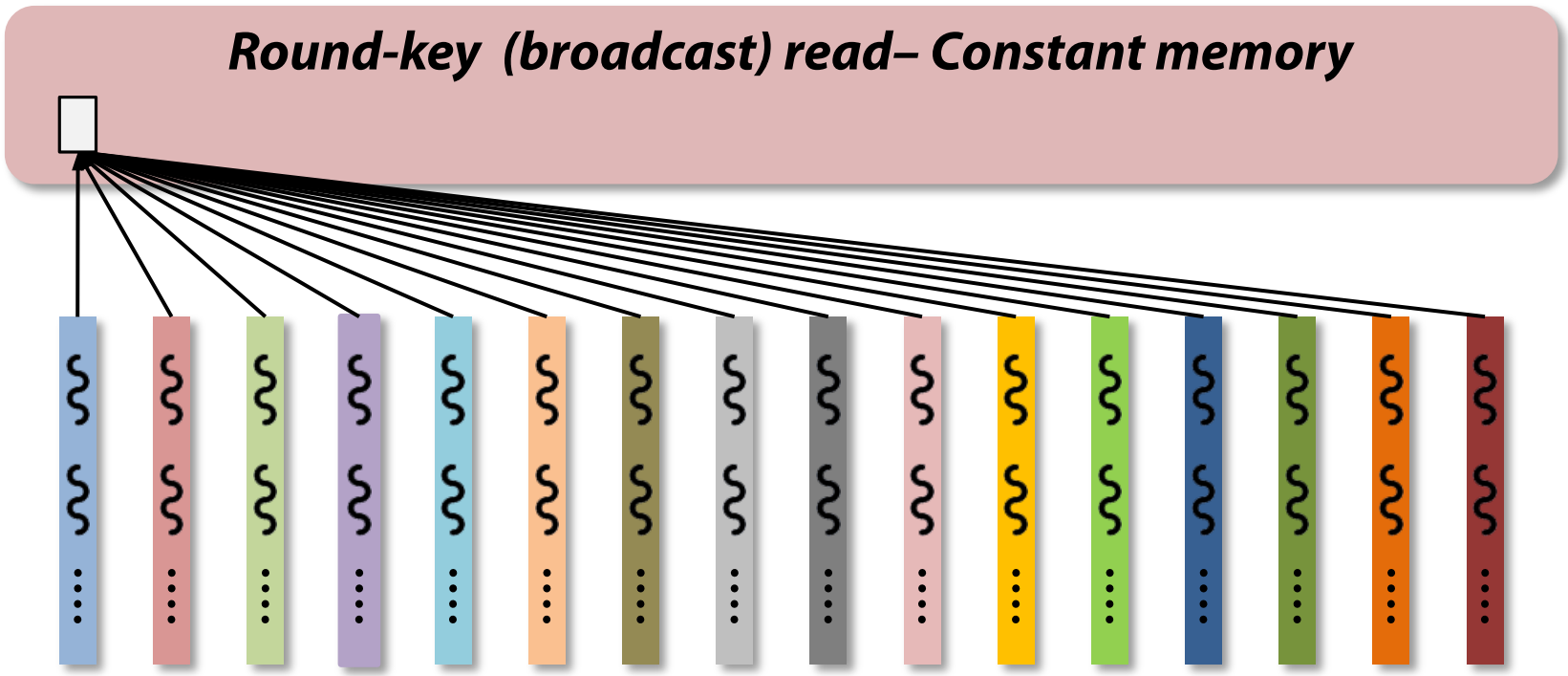
AES GPU Implementation (cont.)

- On-the-fly version
  $\longrightarrow$ all threads cache 1^{st} round key to shared memory
- Texture-memory version
  $\longrightarrow$ threads of same stream group share a common key



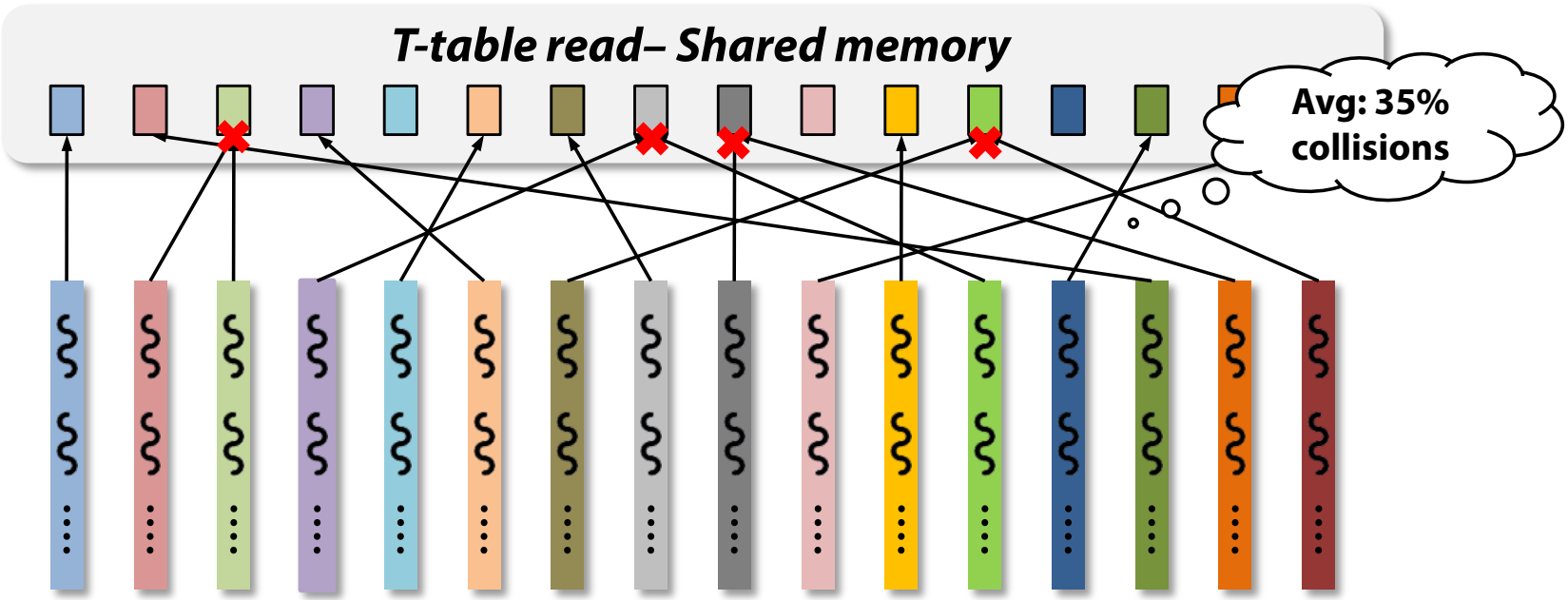
AES GPU Implementation (cont.)

- PTX pre-expanded key (benchmark-only) version
  $\longrightarrow$ threads in a thread block share 1 key.



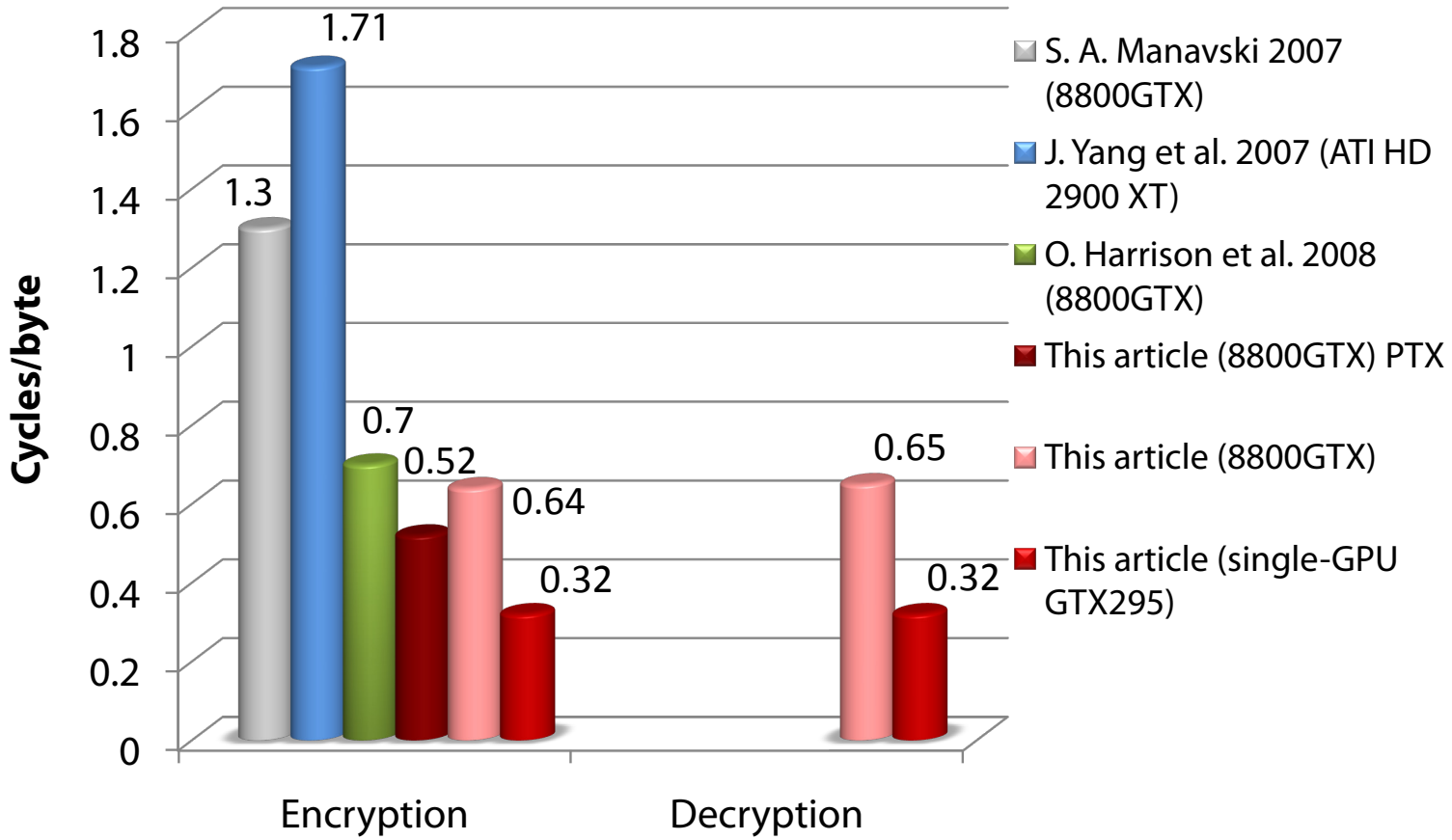
AES GPU Implementation (cont.)

- *T*-tables placed in shared memory:



- Even with collisions → greatest speedup!

GPU Results Comparison



Encryption: **30.9** and **16.8 Gbps** on single GTX 295 GPU and 8800GTX.
Decryption: **30.8** and **16.6 Gbps** on single GTX 295 GPU and 8800GTX.

Conclusions

Our new AES-128 software speed records for encryption and decryption

x times faster compared to the previous records

- 8-bit AVR
  - 1.24× encryption
  - 1.10× decryption
  - smaller code size
- 32-bit ARM
  - 1.17× encryption
- Cell Broadband Engine (SPE)
  - 1.10× encryption
  - 1.23× decryption
- NVIDIA GPU
  - 1.2× encryption
  - 1.34× encryption (kernel-only)
  - First decryption implementation
- All numbers subject to further improvements

Conclusions

Our new AES-128 software speed records for encryption and decryption

x times faster compared to the previous records

- 8-bit AVR
  - 1.24× encryption
  - 1.10× decryption
  - smaller code size
- 32-bit ARM
  - 1.17× encryption
- Cell Broadband Engine (SPE)
  - 1.10× encryption
  - 1.23× decryption
- NVIDIA GPU
  - 1.2× encryption
  - 1.34× encryption (kernel-only)
  - First decryption implementation
- All numbers subject to further improvements

To be continued...