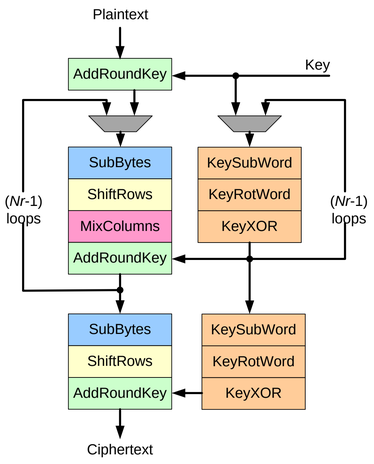
A High-Performance Area-Efficient AES Cipher on a Many-Core Platform

Bin Liu and Bevan M. Baas

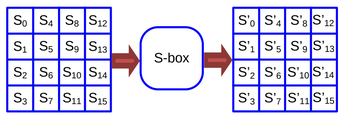
Department of Electrical and Computer Engineering, University of California, Davis, CA 95616

Advanced Encryption Standard

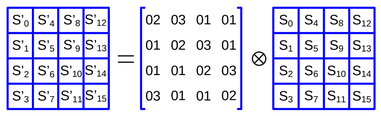


Length of *round key* (bits)	Number of Rounds (N_r)
128	10
192	12
256	14

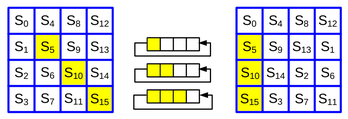
SubBytes: byte substitution from a look up table



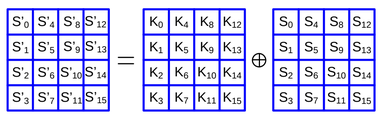
MixColumns: each column multiplies a fixed polynominal over GF(2^8)



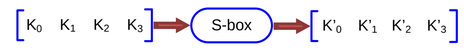
ShiftRows: cyclically shift by one, two and three bytes in the 2nd, 3rd and 4th row



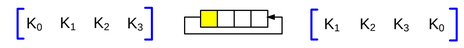
AddRoundKey: round key is added to byte blocks using a bitwise XOR operation



- **KeySubWord**



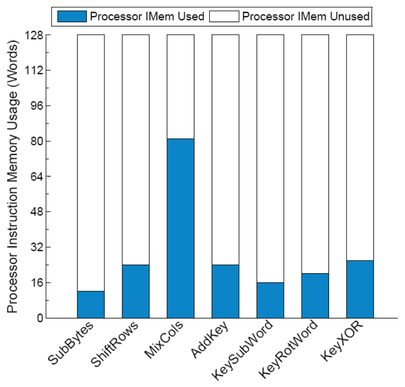
- **KeyRotWord**

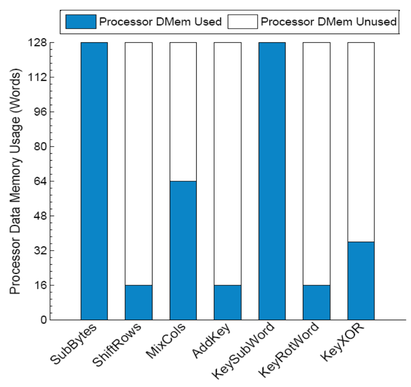


- **KeyXOR**: Every word *w*[*i*] is equal to the bitwise XOR of the previous word, *w*[*i-1*], and the word *Nk* position earlier, *w*[*i-Nk*].

  Nk equals 4, 6 or 8 for the key length of 128, 192 and 256 bits

Optimization II: Reduced Number of Cores





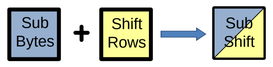
- Before optimization

  22% average IMem usage

  43% average DMem usage

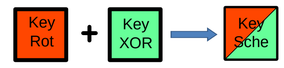
- Core merging should not introduce new bottlenecks or exceed memory limitations

- Step I: Combine the neighboring *SubBytes* and *ShiftRows* into one *SubShift* core



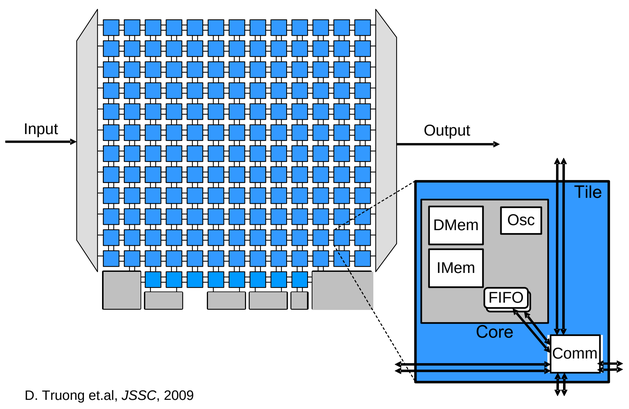
  - $T_{EXE\_SUBSHIFT}$ = 148 cycles per data block
  - 80% Imem and 100% DMem usage

- Step II: Combine the neighboring *KeyRot* and *KeyXOR* into one *KeySche* core



  - $T_{EXE\_KEYSCHE}$ = 60 cycles per data block
  - 24% Imem and 28% DMem usage

Targeted Fine-Grained Many-Core Platform

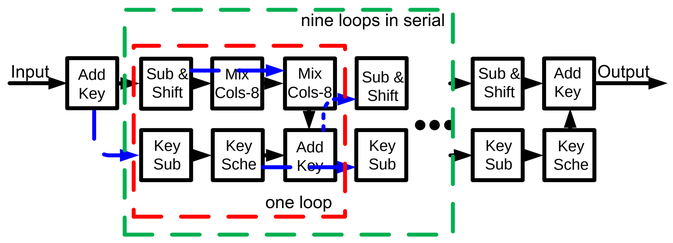


D. Truong et.al, *JSSC*, 2009

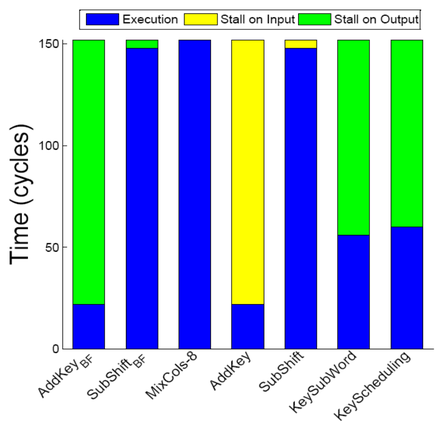
AsAP2 Single Tile (164 total)	
Area	0.17 mm^2
Transistors	325,000
CMOS Tech.	65 nm low-leakage
Max. frequency	1.2GHz @ 1.3 V
Instru. Memory	128 x 32-bit
Data Memory	128 x 16-bit

- In-order single-issue 6-stage pipeline
- No-specialized instructions
- Reconfigurable 2D-mesh network

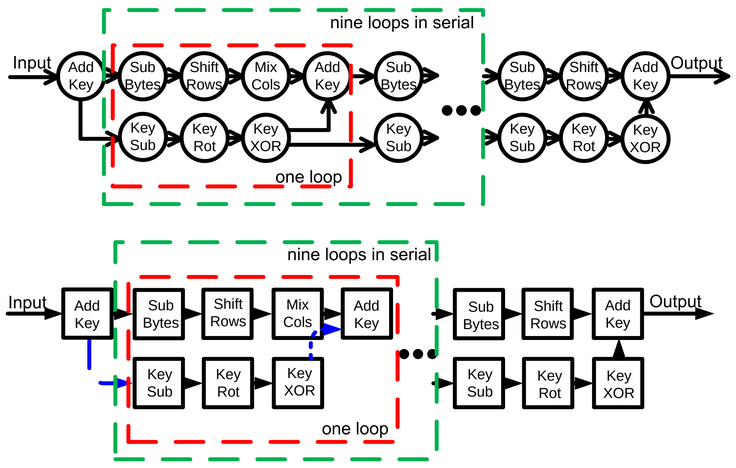
Optimized Design of AES Cipher



- Throughput is 43% higher (9.5 cycles per block)
- 16% fewer cores required (59 cores)



Preliminary Design of AES Cipher



Processor Name	Execution Time for Processing One 128-bit Data Block (Clock Cycles)
SubBytes	132
ShiftRows	38
MixColumns	266
AddRoundKey	22
KeySubWord	56
KeyRotWord	26
KeyXOR	56

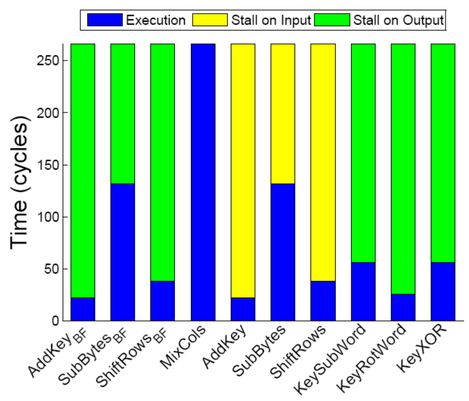
- (*Nr*-1) times loop unrolling
- Throughput is 266 cycles per datablock
- 70 cores are used

Comparison with Related Work

Platform	Method	Tech. (nm)	Area (mm^2)	Max Freq. (MHz)	Throughput (cycles/byte)	Scaled Throughput (Mbps)	Scaled Area (mm^2)	Scaled Throughput/Area (Mbps/mm^2)
Pentium 4 561	Bitslice	90	112	3600	16	2492	58.42	42.66
Athlon 64 3500	Bitslice	90	193	2200	10.6	2299	101	22.76
Core 2 Duo E6400	Bitslice	65	111	2130	9.19	1854	111	16.70
Core 2 Quad Q6600 (one core)	Bitslice + SSSE3	65	286/2 = 143	2400	9.32	2060	143	14.41
Core 2 Quad Q9550 (one core)	Bitslice + SSSE3	45	214/4 = 53.5	2830	7.59	2065	112	18.44
Core i7 920 (one core)	Bitslice + SSSE3	45	263/4 = 65.75	2668	6.92	2135	133	16.05
TI C6201		180	NA	200	14.25	311	NA	NA
GeForce 8800 GTX	T-Box	90	484	575	NA	11500	252	45.63
This Work AsAP		65	6.63	1210	9.5	1019	6.63	153.70

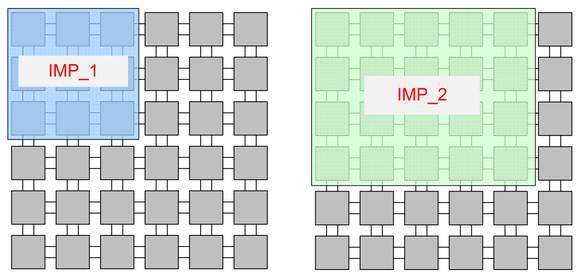
- Compared to CPUs, our design achieves 3.6-10.7x higher throughput per chip area
- Compared to DSP, our design achieves 1.5x higher throughput
- Compared to GPU, our design achieves 3.4x higher throughput per chip area

Optimization I: Increased Throughput



- Cores running *MixColumns* workloads are 2x slower than others, so we parallelize each into two *MixCol-8* programs
- Throughput is increased by 43% (152 cycles per block)
- Increased parallelization requires 10 more cores
- MixCol-8 cores are new bottlenecks

Core-Scaling on Many-Core Platforms



- Fine-grain many-core processor arrays with large numbers of cores enable application libraries with varying numbers of cores
- Optimum run-time programs chosen with joint “core scaling” and supply voltage and clock frequency scaling
- New possibilities to tradeoff number of processors, performance and energy efficiency

Acknowledgments: The authors gratefully acknowledge support from NSF Grant 0430090, 0903549 and CAREER Award 0546907, SRC GRC Grant 1598 and 1791, CSR Grant 1659, UC Micro, ST Microelectronics, Intel, and Intellasys. The authors also acknowledge the support of the C2S2 Focus Center, one of the six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation entity.