



Reduce Power and Energy Consumption through ISA Extension

Steve Leibson
Technology Evangelist
Tensilica, Inc.
sleibson@tensilica.com



Agenda

- Power and energy consumption and the need for task acceleration
- Hardware acceleration versus ISA extension
- Three application examples
 - AES
 - Viterbi
 - FFT
- Conclusions

On-Chip Energy Consumption is Rising

- Dynamic and static energy consumption are rising with each new IC fabrication node (that's very bad!)

On-Chip Energy Consumption is Rising

- Dynamic and static energy consumption are rising with each new IC fabrication node (that's very bad!)
 - Trend to higher clock rates drives dynamic power up

On-Chip Energy Consumption is Rising

- Dynamic and static energy consumption are rising with each new IC fabrication node (that's very bad!)
 - Trend to higher clock rates drives dynamic power up
 - Core voltages drop to compensate for higher dynamic power levels

On-Chip Energy Consumption is Rising

- Dynamic and static energy consumption are rising with each new IC fabrication node (that's very bad!)
 - Trend to higher clock rates drives dynamic power up
 - Core voltages drop to compensate for higher dynamic power levels
 - Transistor threshold voltages drop to allow lower core voltages

On-Chip Energy Consumption is Rising

- Dynamic and static energy consumption are rising with each new IC fabrication node (that's very bad!)
 - Trend to higher clock rates drives dynamic power up
 - Core voltages drop to compensate for higher dynamic power levels
 - Transistor threshold voltages drop to allow lower core voltages
 - Leakage and static energy consumption rise due to lower transistor threshold voltages

Why Reduce On-Chip Energy Consumption?

- Higher energy consumption hurts specs, costs money



Why Reduce On-Chip Energy Consumption?

- Higher energy consumption hurts specs, costs money
 - Less battery life



Why Reduce On-Chip Energy Consumption?

- Higher energy consumption hurts specs, costs money
 - Less battery life
 - Less talk, play, record time
 - Less standby time



Why Reduce On-Chip Energy Consumption?

- Higher energy consumption hurts specs, costs money
 - Less battery life
 - Less talk, play, record time
 - Less standby time
 - More costly power supply



Why Reduce On-Chip Energy Consumption?

- Higher energy consumption hurts specs, costs money
 - Less battery life
 - Less talk, play, record time
 - Less standby time
 - More costly power supply
 - Bigger supply costs more money, takes more space in package
 - Higher power supply heat hurts reliability, raises warranty costs



Why Reduce On-Chip Energy Consumption?

- Higher energy consumption hurts specs, costs money
 - Less battery life
 - Less talk, play, record time
 - Less standby time
 - More costly power supply
 - Bigger supply costs more money, takes more space in package
 - Higher power supply heat hurts reliability, raises warranty costs
 - More costly package cost, heat sinking, and fans



Why Reduce On-Chip Energy Consumption?

- Higher energy consumption hurts specs, costs money
 - Less battery life
 - Less talk, play, record time
 - Less standby time
 - More costly power supply
 - Bigger supply costs more money, takes more space in package
 - Higher power supply heat hurts reliability, raises warranty costs
 - More costly package cost, heat sinking, and fans
 - Cheap plastic IC packages cannot dissipate a lot of power
 - Bigger heat sinks cost more, take more space in enclosure
 - Fans increase audible noise and need even more space

So How Much Could it Cost?



MacDailyNews
where mac news comes first

news opinion archiv

Friday, July 06, 2007 - 02:07 PM EDT — Apple Stock Quote: 132.1699 (-0.5601, -0.44%)

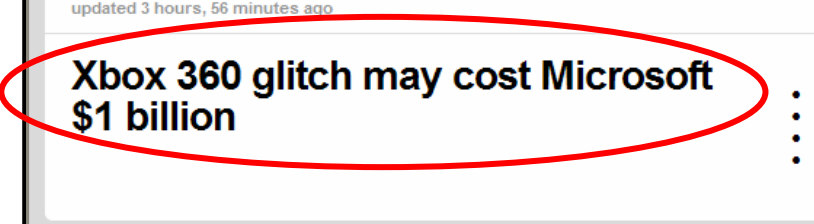
Serious flaws in Xbox 360 hardware to cost Microsoft at least \$1 billion
Friday, July 06, 2007 - 01:41 PM EDT

Go to iPodDailyNews

5 Day Most Commented

Rush Limbaugh giving away 10 Apple 8GB iPhones with

updated 3 hours, 56 minutes ago



CNN.com

HOME WORLD

Hot Topics » Weather

Xbox 360 glitch may cost Microsoft \$1 billion



Telegraph.co.uk

NO.1 WEBSITE hitwise JAN-MAR 2007

Home News Sport Business Travel Jobs Motoring Property SEARCH

Money home Business Your Money

Executive jobs 55k+ Business search Business travel Business club

Announcements Arts Blogs Comment Crossword Dating Digital Life Earth Education Expat

Microsoft takes \$1bn hit on Xbox
By Emma Thelwell, Online City Reporter
Last Updated: 9:51am BST 06/07/2007

Microsoft has admitted it will have to spend up to \$1.15bn repairing defective Xbox 360 consoles and extending their warranty policies.

The software giant said that an "unacceptable number of repairs" to its Xbox 360 has forced it to take action.

From now on, any Xbox 360 customer who experiences a "general hardware failure" - which is indicated by three flashing red lights - will be covered by a three year warranty from the date of purchase.



Robbie Bach - 'We value our community tremendously'

SEATTLE, Washington (AP) -- In another setback for Microsoft Corp.'s unprofitable entertainment devices division, the company says it is planning to spend at least \$1 billion to repair serious problems with its Xbox 360 video game console.



MICROSOFT
The Xbox 360 has seen slow sales in Japan.

Microsoft declined to detail the problem, but said it caused an onslaught of "general hardware failures" in recent months but said they will extend the warranty on the consoles to three years.

The glitches, and the bad publicity, could have sent the company down as it claws for market share in the highly competitive console market. Xbox 360 ranked No. 2 in unit sales behind Nintendo's Wii, but still beat out Sony's PS3, according to data from NPD Group.

"We don't think we've been getting the job done," said Robbie Bach, president of Microsoft's entertainment and devices division, which also makes the Zune digital music player, a distant competitor to Apple Inc.'s powerhouse iPod. "In

the past few months, we have been having to make Xbox 360 console repairs at a rate too high for our liking."

It All Starts with the Chip Design



It All Starts with the Chip Design

- Find ways to cut on-chip power and energy consumption



It All Starts with the Chip Design

- Find ways to cut on-chip power and energy consumption
 - Drive clock rates down
 - (Very heretical)



It All Starts with the Chip Design

- Find ways to cut on-chip power and energy consumption
 - Drive clock rates down
 - (Very heretical)
 - To cut dynamic power dissipation and



It All Starts with the Chip Design

- Find ways to cut on-chip power and energy consumption
 - Drive clock rates down
 - (Very heretical)
 - To cut dynamic power dissipation and
 - To reduce the need for low-threshold transistors



It All Starts with the Chip Design

- Find ways to cut on-chip power and energy consumption
 - Drive clock rates down
 - (Very heretical)
 - To cut dynamic power dissipation and
 - To reduce the need for low-threshold transistors
 - To reduce static power dissipation



It All Starts with the Chip Design

- Find ways to cut on-chip power and energy consumption
 - Drive clock rates down
 - (Very heretical)
 - To cut dynamic power dissipation and
 - To reduce the need for low-threshold transistors
 - To reduce static power dissipation
- Avoid the use of buses when possible

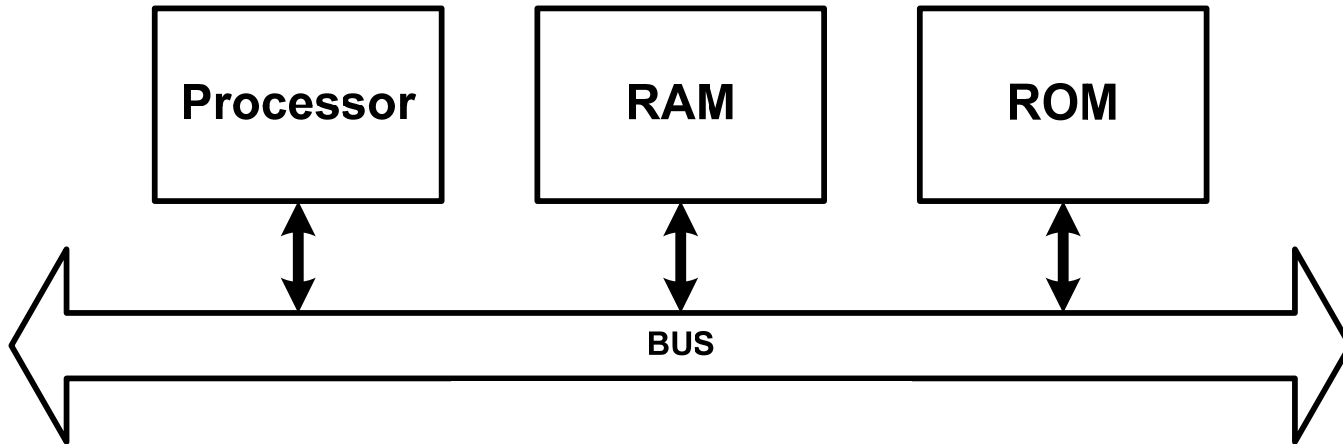


It All Starts with the Chip Design

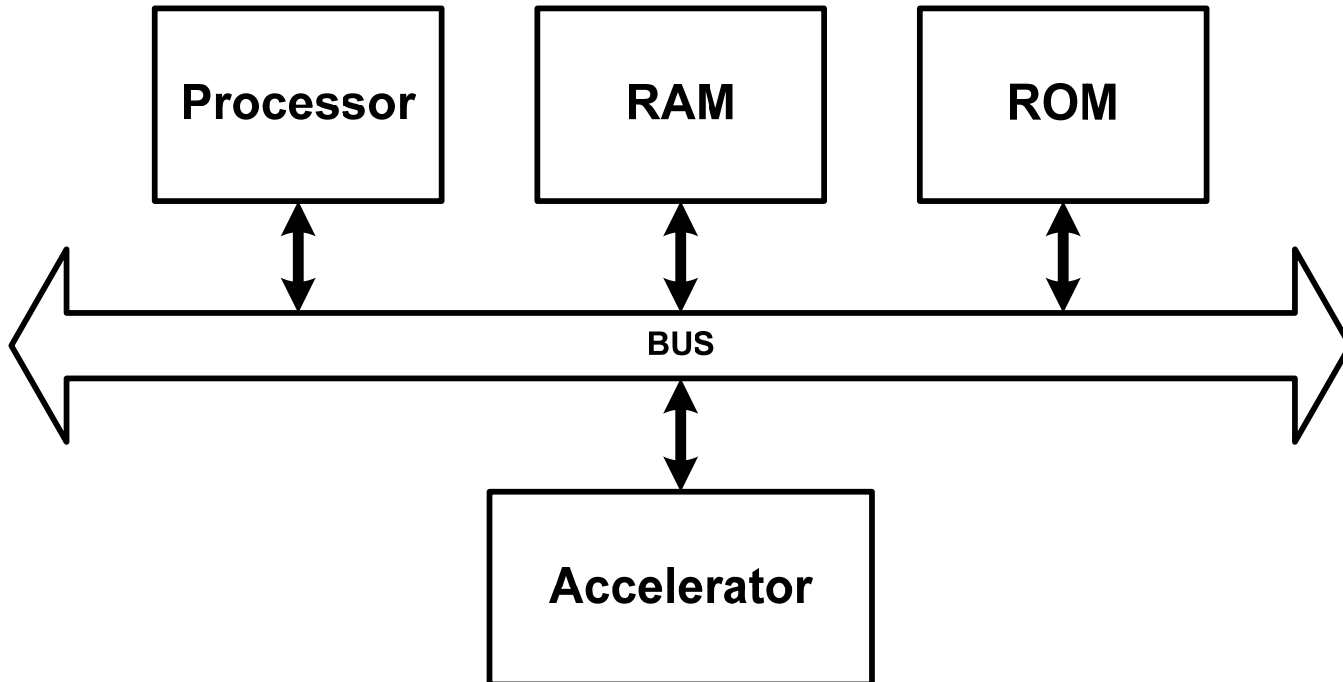
- Find ways to cut on-chip power and energy consumption
 - Drive clock rates down
 - (Very heretical)
 - To cut dynamic power dissipation and
 - To reduce the need for low-threshold transistors
 - To reduce static power dissipation
- Avoid the use of buses when possible
 - Find alternative communication methods that don't require blocks to drive wide, shared, highly capacitive buses



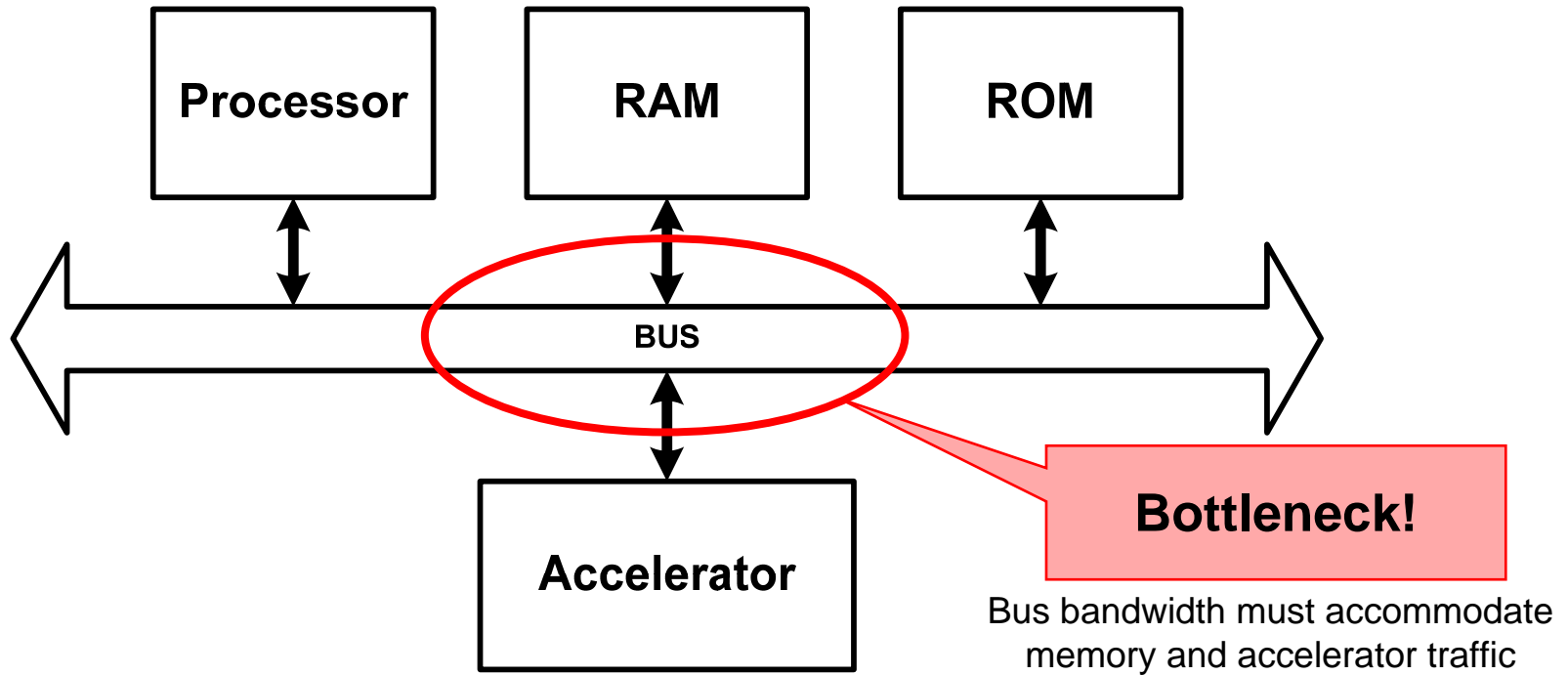
Conventional Hardware Acceleration



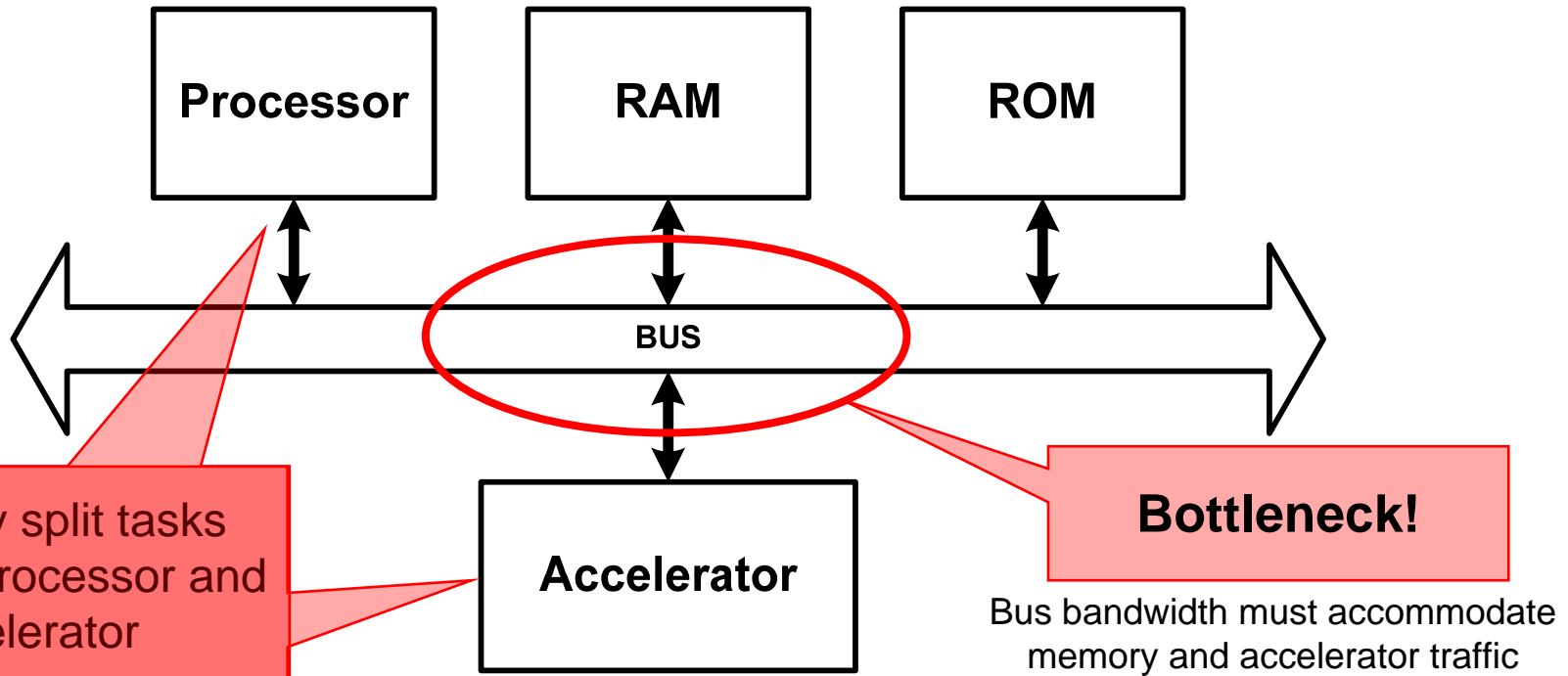
Conventional Hardware Acceleration



Conventional Hardware Acceleration



Conventional Hardware Acceleration



Manually split tasks between processor and accelerator

Bottleneck!

Bus bandwidth must accommodate memory and accelerator traffic

Hardware-accelerated tasks live outside of the software environment

Reduce Energy Use With ISA Extension



ISA = instruction-set architecture (registers + instructions)



Reduce Energy Use With ISA Extension

- Use processor ISA extension to improve task-execution speed, energy consumption, or both



ISA = instruction-set architecture (registers + instructions)



Reduce Energy Use With ISA Extension

- Use processor ISA extension to improve task-execution speed, energy consumption, or both
- Algorithm-specific registers and operations reduce the number of cycles needed to execute the algorithm



ISA = instruction-set architecture (registers + instructions)



Reduce Energy Use With ISA Extension

- Use processor ISA extension to improve task-execution speed, energy consumption, or both
- Algorithm-specific registers and operations reduce the number of cycles needed to execute the algorithm
- Reduces energy consumption by executing same number of task iterations in many fewer clock cycles



ISA = instruction-set architecture (registers + instructions)



Reduce Energy Use With ISA Extension

- Use processor ISA extension to improve task-execution speed, energy consumption, or both
- Algorithm-specific registers and operations reduce the number of cycles needed to execute the algorithm
- Reduces energy consumption by executing same number of task iterations in many fewer clock cycles
 - ✓ Fewer clock cycles allow the processor to sleep more at the same clock rate (low standby power)
or...
 - ✓ Allows the processor to run at a lower clock rate, reducing power dissipation and energy consumption in a superlinear ($1/2 CV^2F$) fashion due to lower operating frequency combined with lower core operating voltage



ISA = instruction-set architecture (registers + instructions)



Translating from Processor Speak

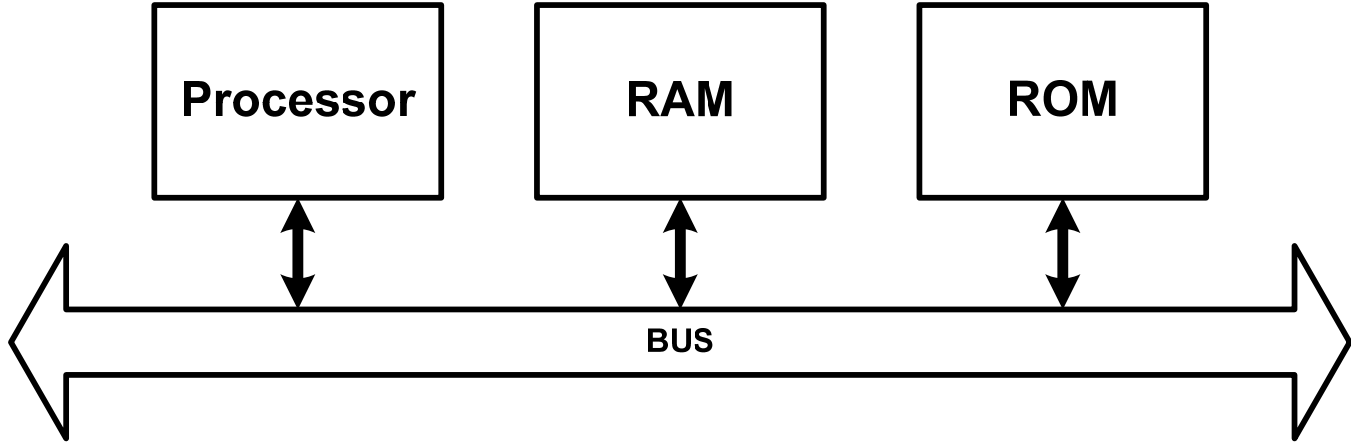
When processor vendors talk about
“adding instructions” or “extending the ISA”

they mean:

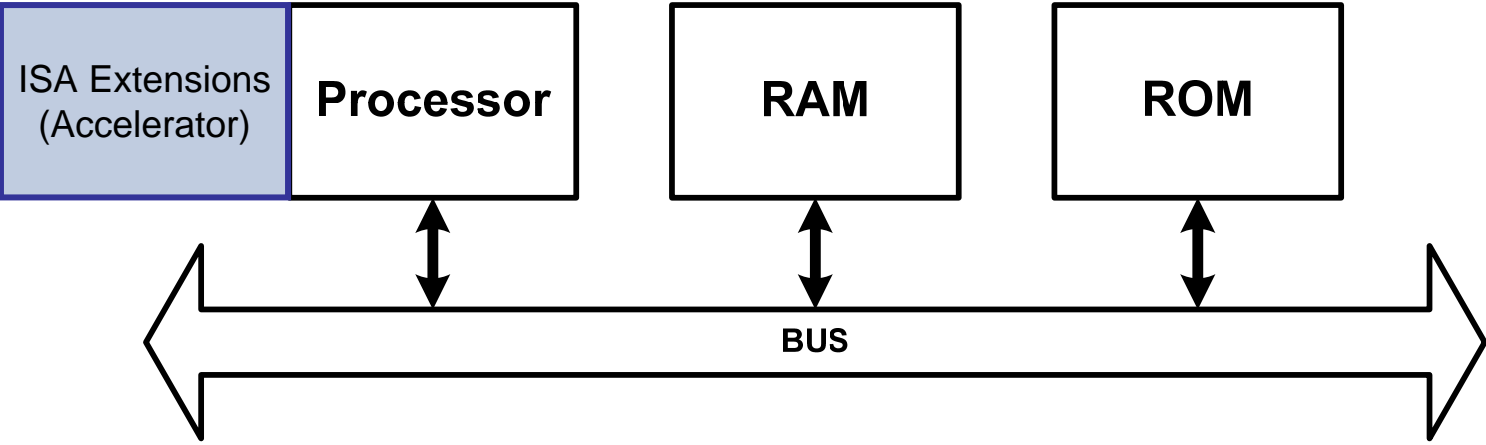
“adding hardware to the processor”

- **generally including (but not limited to):**
 - **new registers**
 - **new register files**
 - **execution-unit additions.**

Acceleration Through ISA Extension

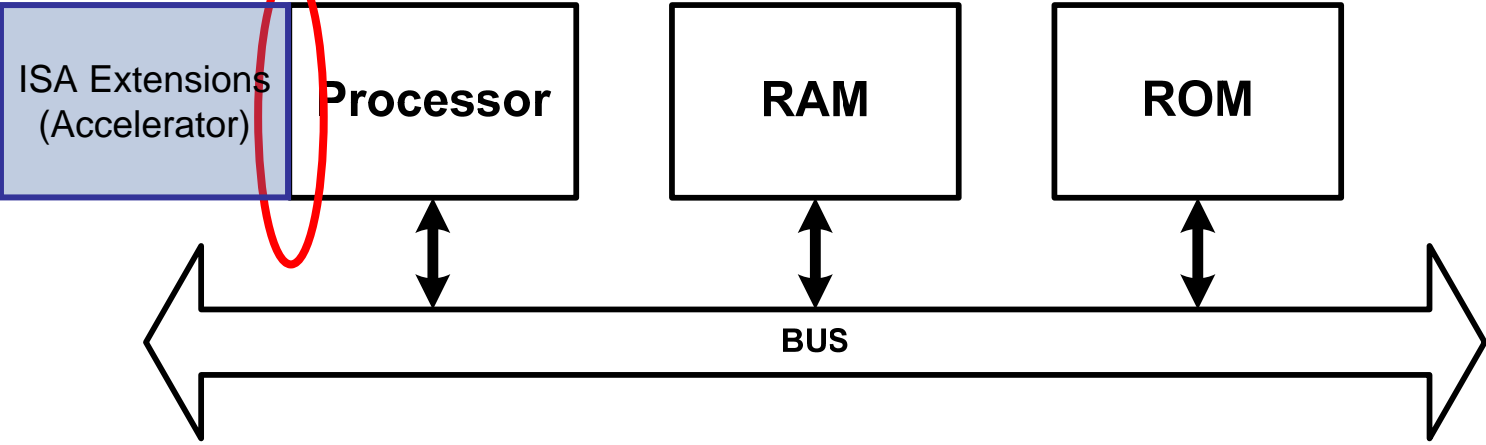


Acceleration Through ISA Extension

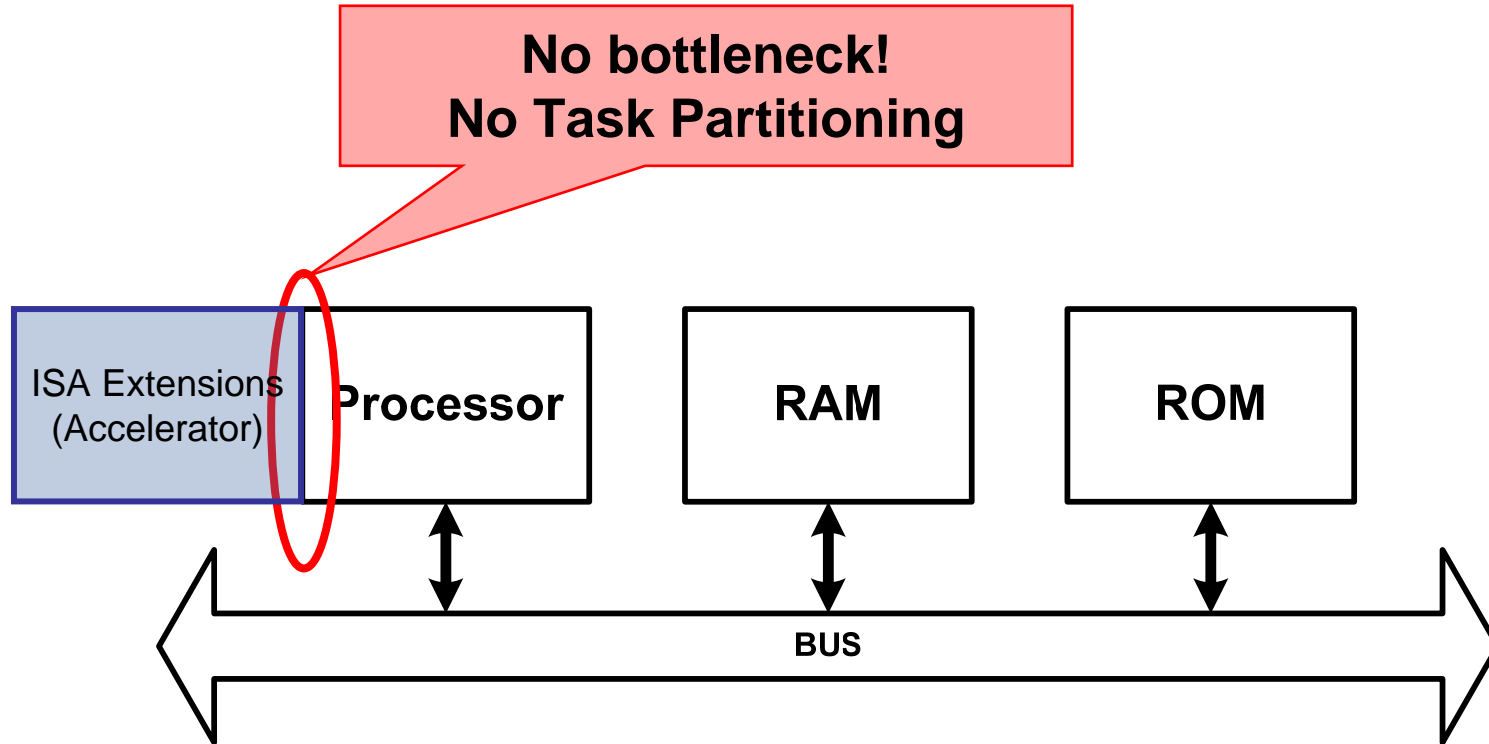


Acceleration Through ISA Extension

No bottleneck!
No Task Partitioning



Acceleration Through ISA Extension



Task-specific data types can be directly mapped to extended registers of any width and remain part of the software environment unlike bus-attached hardware accelerators

How to Add Hardware to a Processor

1. Analyze the algorithm or task
2. Identify key operations for acceleration
3. Identify appropriate data types (as opposed to “wedge” and “cut up” data types to fit existing resources)
4. Define and design resources to accommodate special data types and accelerate operations



Example: AES Cryptography

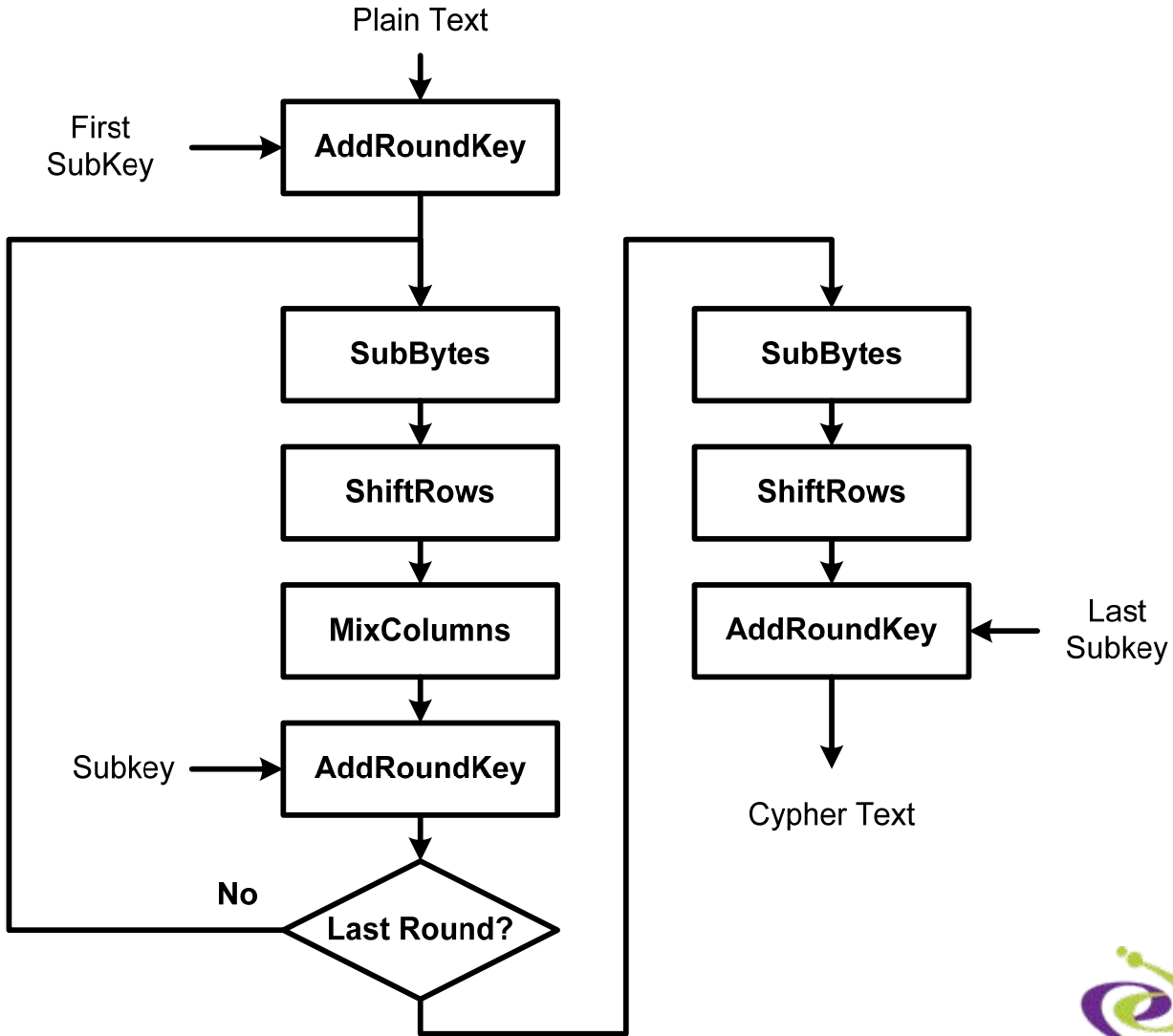
- Block-oriented crypto based on Rijndael cipher
 - FIPS 197, adopted by the US government in 2002
 - Obsoletes DES but not Triple DES
 - Current technology needs 147 trillion years to crack an AES cipher using exhaustive search at 2^{55} keys/sec
- Some AES Applications
 - SATA disk interface (1.2-4.8 Gbits/sec)
 - Wireless LAN (to 10 Mbits/sec to 1 Gbit/sec)



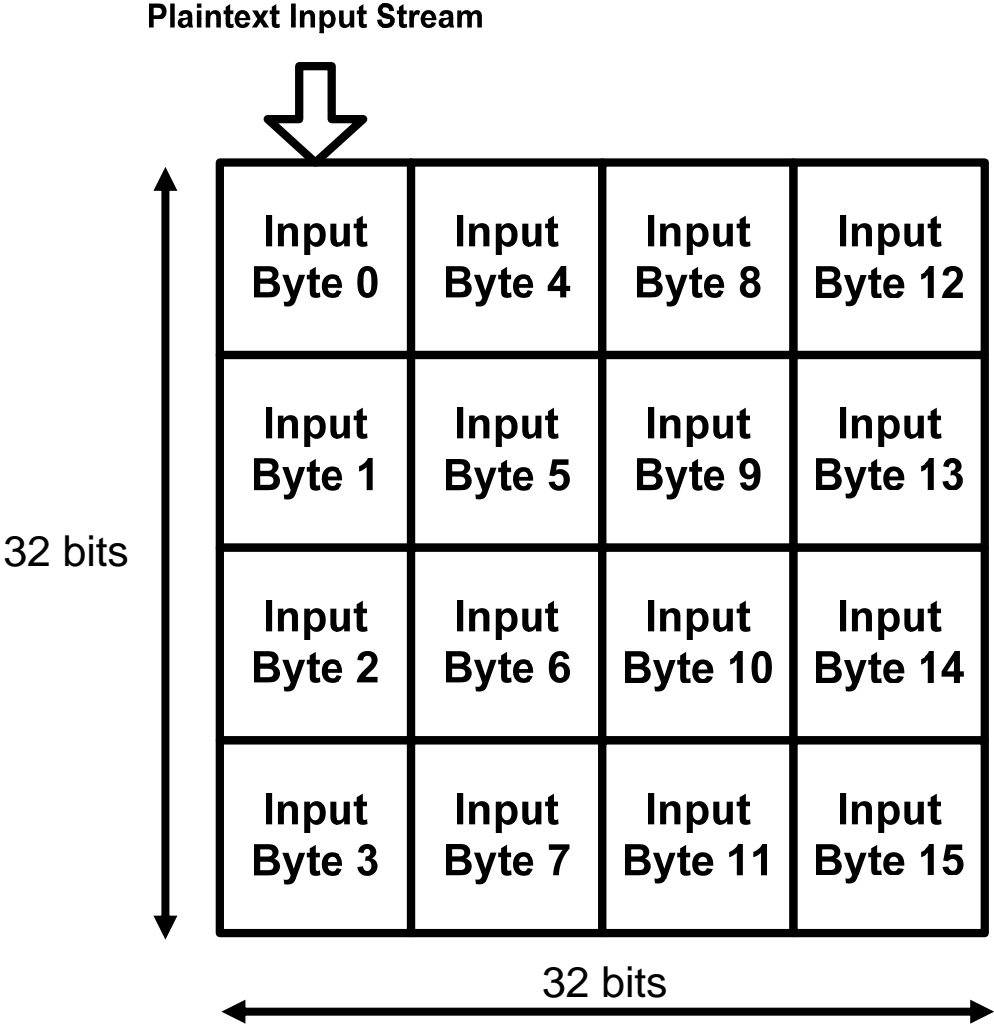
AES Cryptography Operations

- Four basic transformations in the AES algorithm
 - ✓ **SubBytes** – Table-based, byte-wise substitution in state array
 - ✓ **ShiftRow** – Rearrange bytes within rows of the state array
 - ✓ **MixColumn** – Galois matrix multiplication on state array columns
 - ✓ **AddRoundKey** – Byte-wise Galois field addition (128-bit XOR) with bytes from appropriate subkey in key schedule

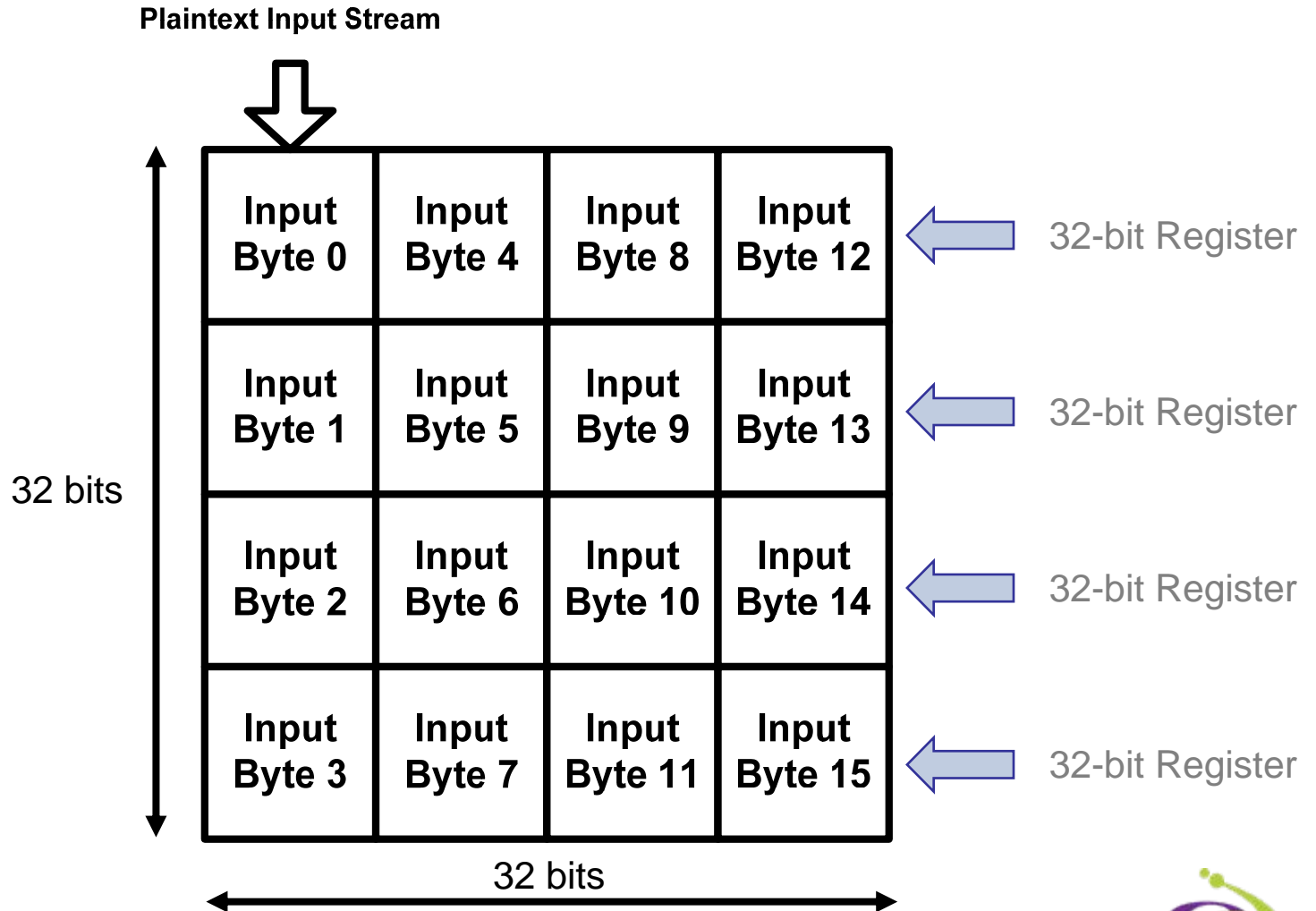
AES Encryption Flowchart



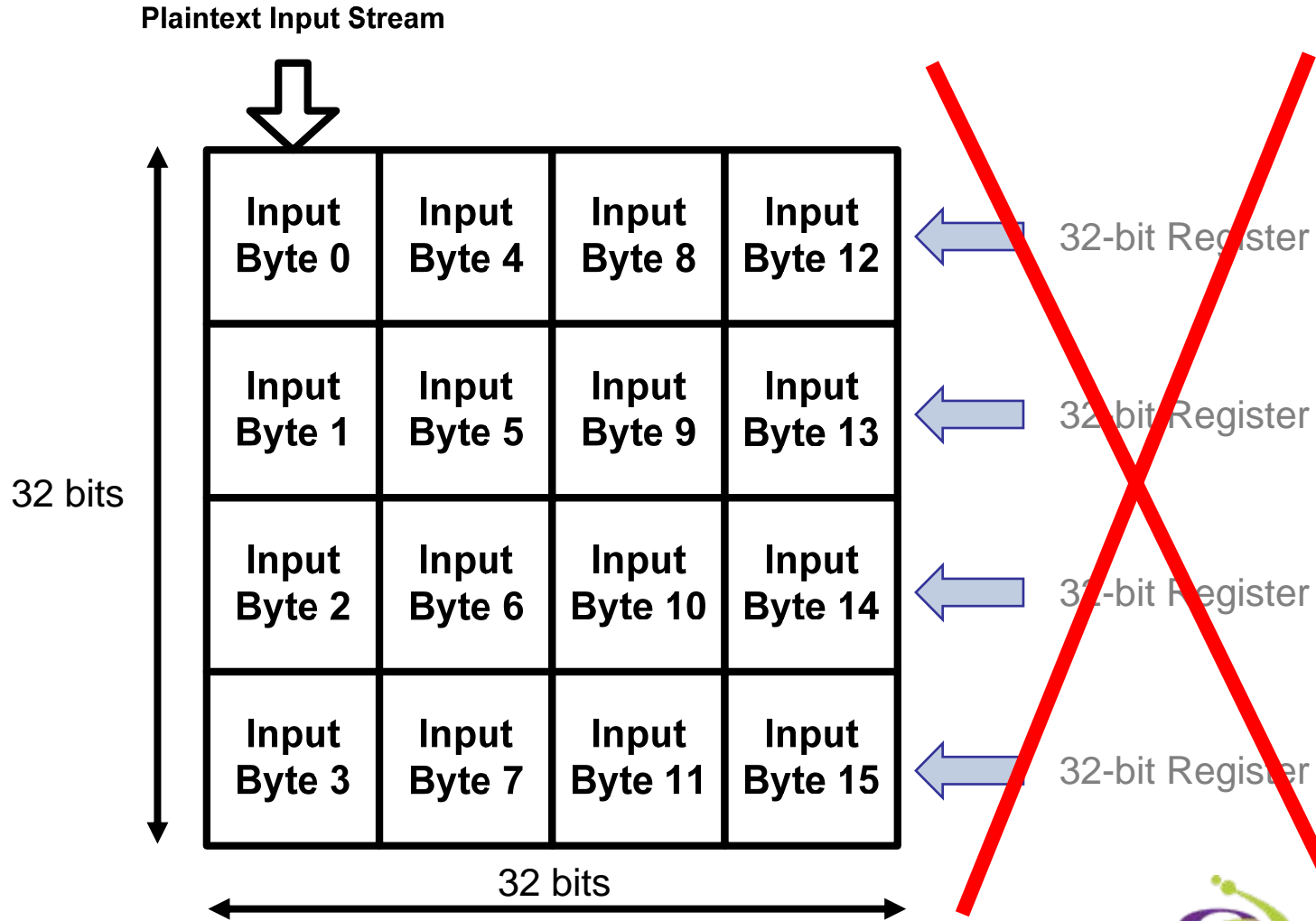
128-Bit AES State Array (Data Type)



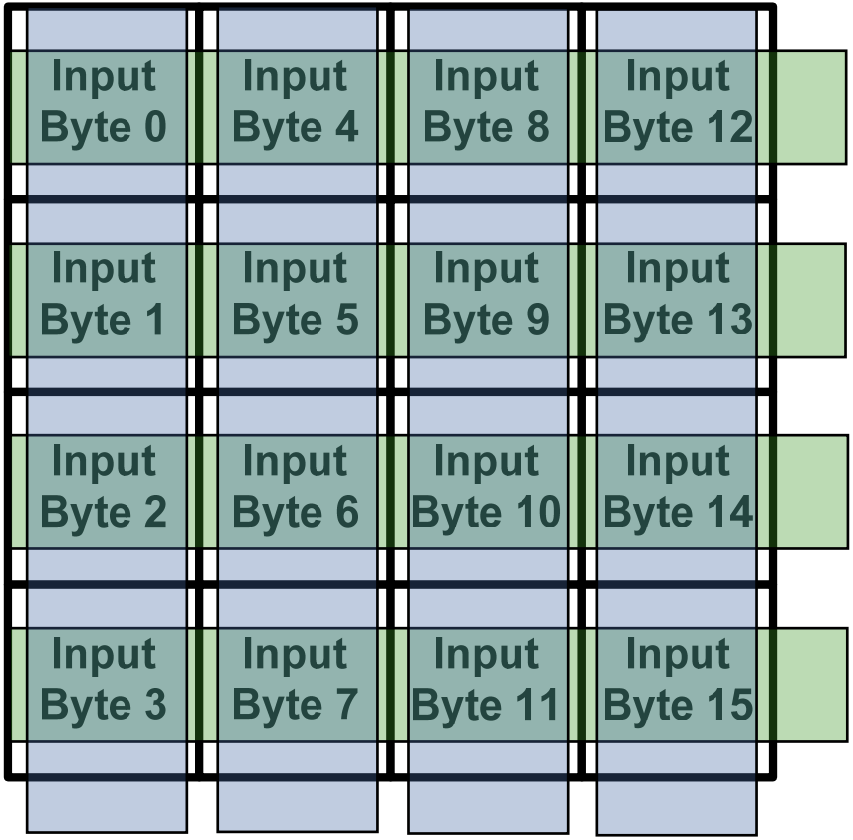
Processor-Based Thinking Immediately Kicks In



Please Resist the Temptation



AES State Array Row and Column Operations

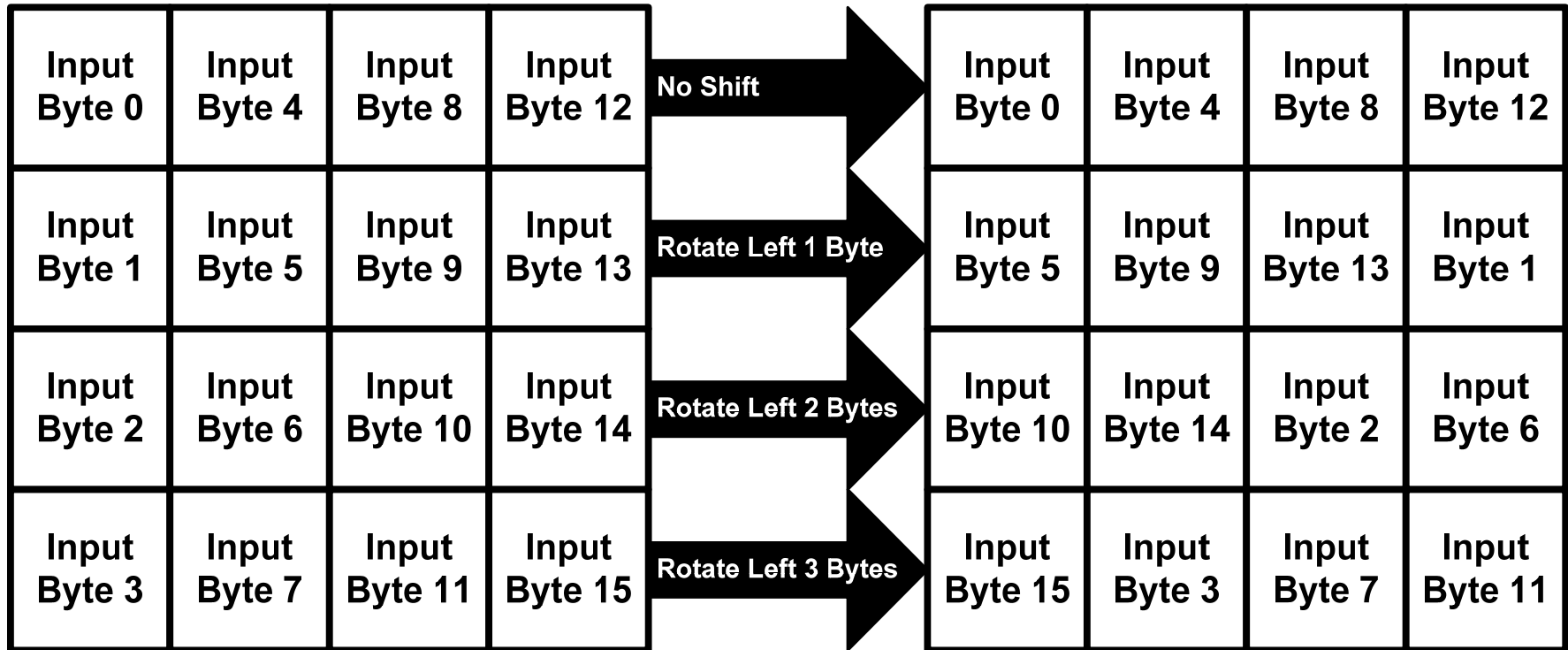


ShiftRow transformation operates on rows

MixColumns transformation operates on columns

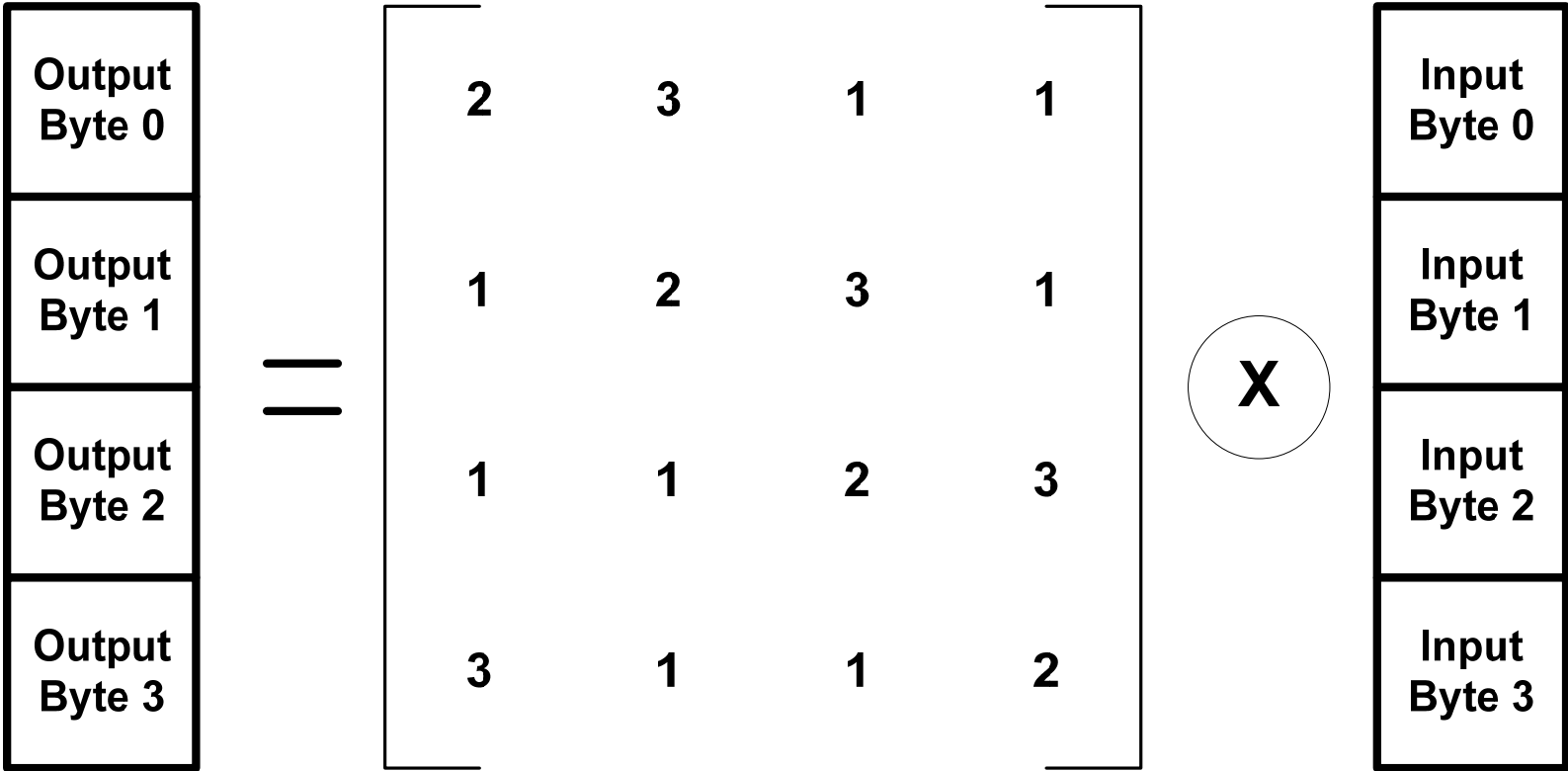


AES ShiftRow Transformation



AES MixColumn Transformation

AES Galois Matrix Multiplication on a State-Array Column



Notes:
Galois multiplication is much simpler than binary multiplication.
The multiplicand is always 1, 2, or 3.



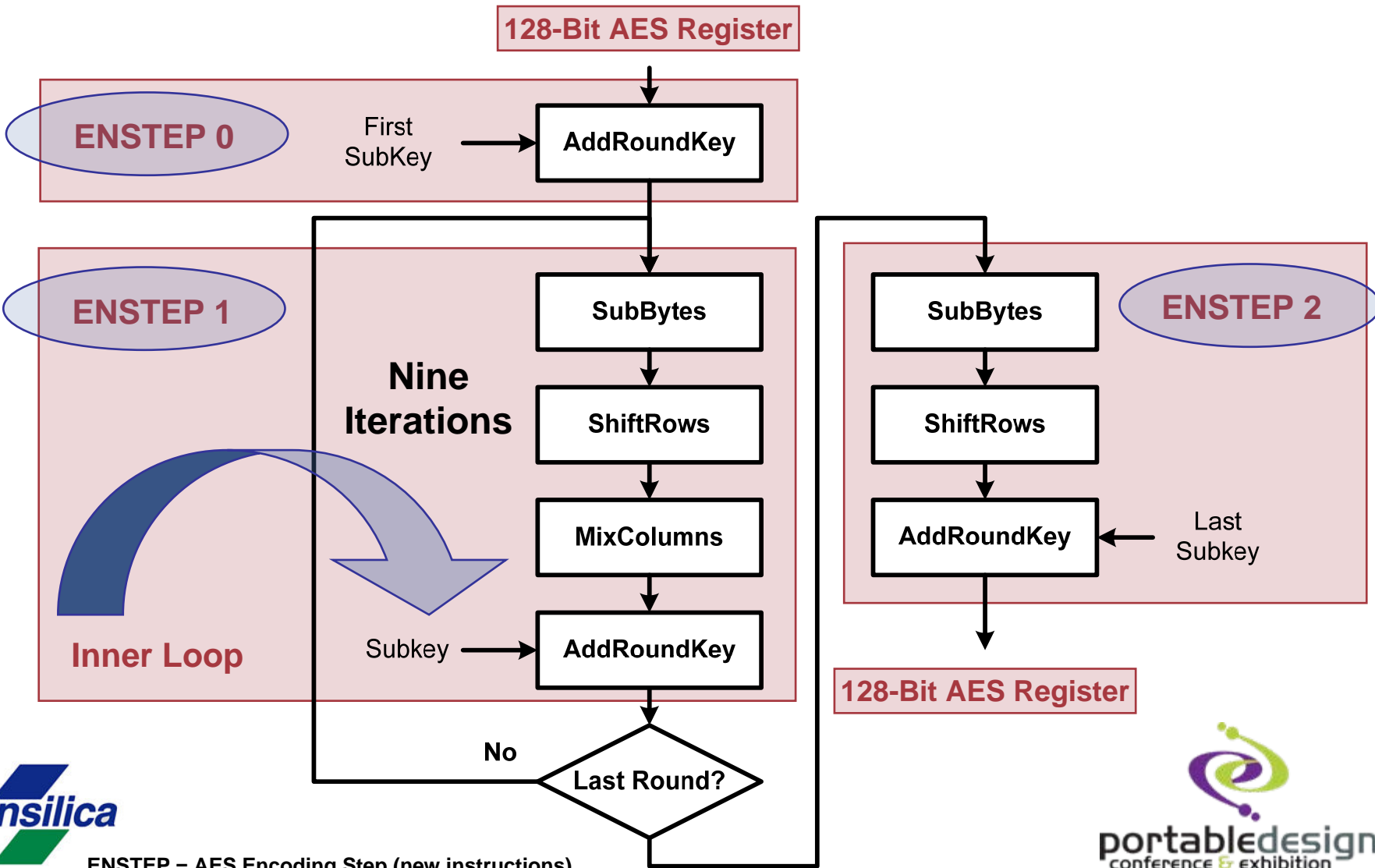
Baseline AES Results Coded in C for RISC processor

	Cycle Count (per 10 blocks)	Estimated Energy (uJ)	Estimated Instantaneous Power (mW @ 100 MHz)
AES Encryption	353,493	57.03	16.13
AES Encryption and Decryption	679,517	108.18	15.92[*]

Estimates generated by Xenergy energy estimator and the Xtensa ISS, assuming TSMC 130 LV process @ 100 MHz.

* Lower power for encryption and decryption is due to more stalled (lower-power) processor cycles while waiting for memory

Add Three 128-Bit AES Encoding Instructions and a 128-Bit AES Register



ENSTEP = AES Encoding Step (new instructions)



Code Encryption Rounds in TIE

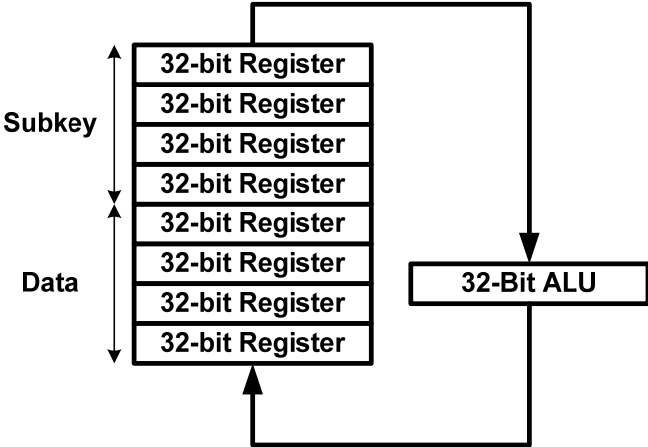
The three new ENSTEP instructions use combinations of one or more AES encoding steps:

- 1. Load 128-bit subkey from the key schedule with automatic subkey increment (forces a 2-cycle instruction to wait for the load)**
- 2. AddRoundKey transformation (ENSTEP 0,1,2)**
- 3. SubBytes transformation (ENSTEP 1 & 2)**
- 4. ShiftRow transformation (ENSTEP 1 & 2)**
- 5. MixColumn transformation (ENSTEP 1 only)**

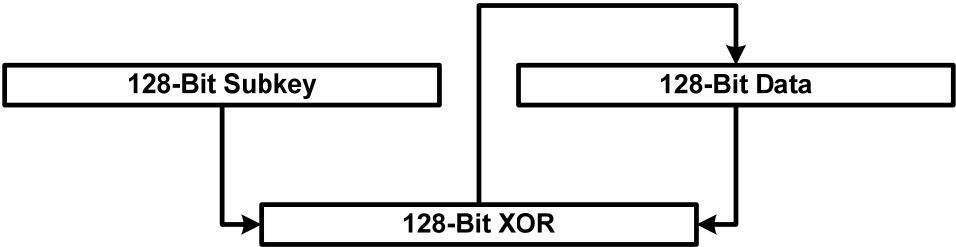
128-Bit AddRoundKey Transformation

$$\text{AES Register} = \text{Data} \text{ XOR } \text{Subkey}$$

Conventional RISC Processor

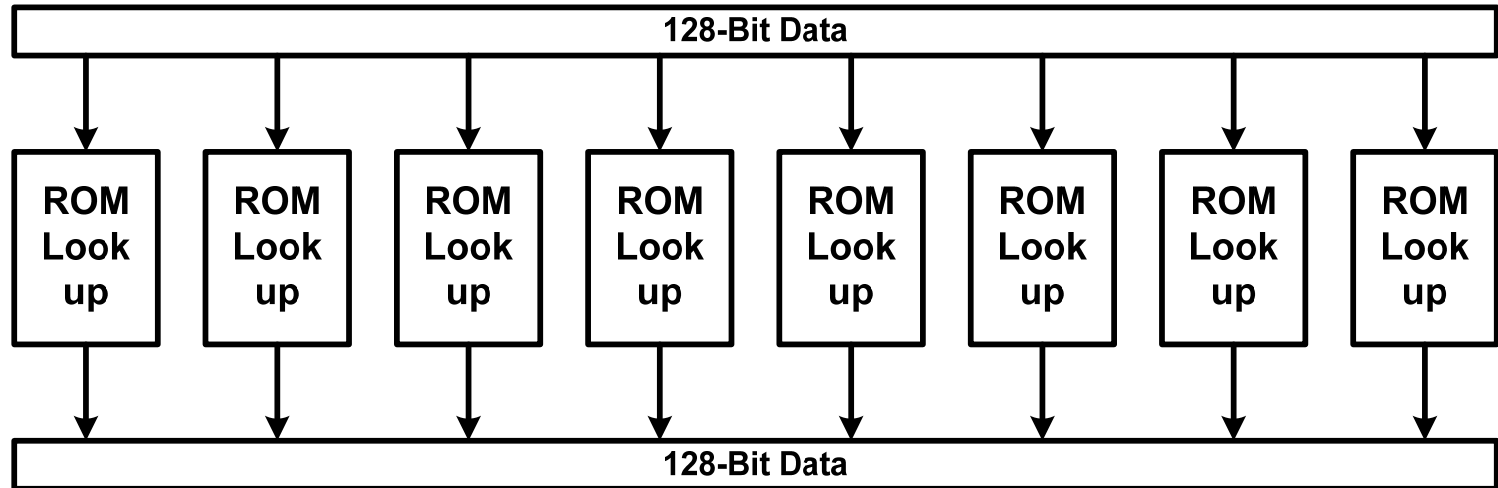


ISA Extension



128-Bit SubBytes Transformation

Use lookup tables to perform 16 simultaneous byte substitutions for the 16 data bytes



128-Bit SubBytes Transformation

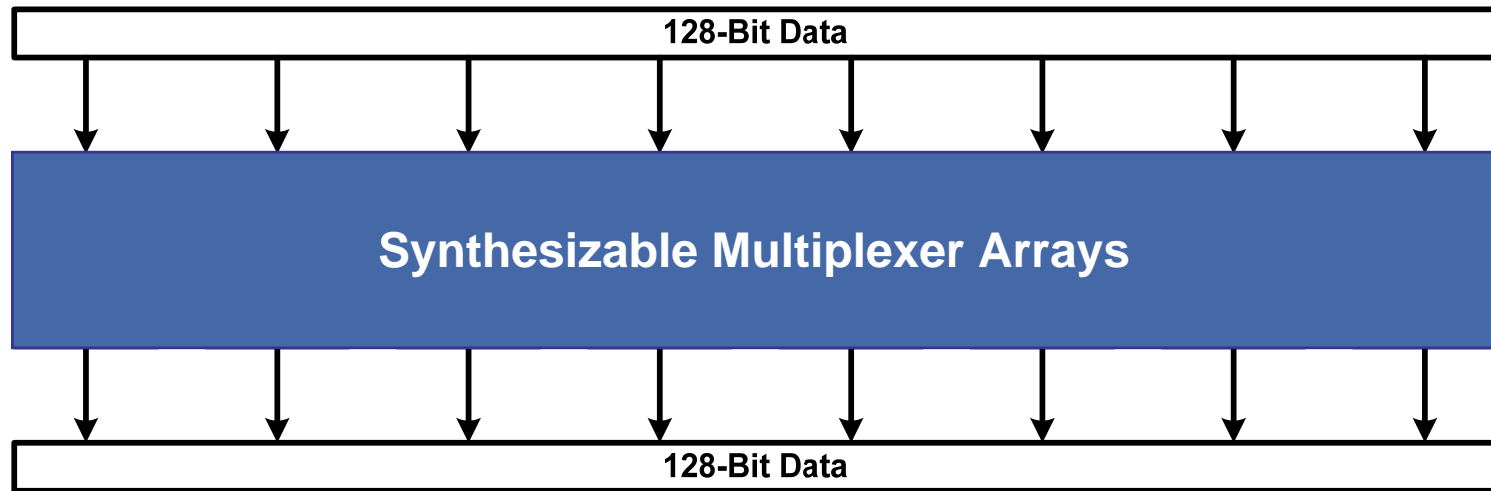
Use lookup tables to perform 16 simultaneous byte substitutions for the 16 data bytes



Saving System Power: Avoid memory and bus traffic associated with retrieving fixed substitution values by implementing the lookup tables as logic within the function unit.

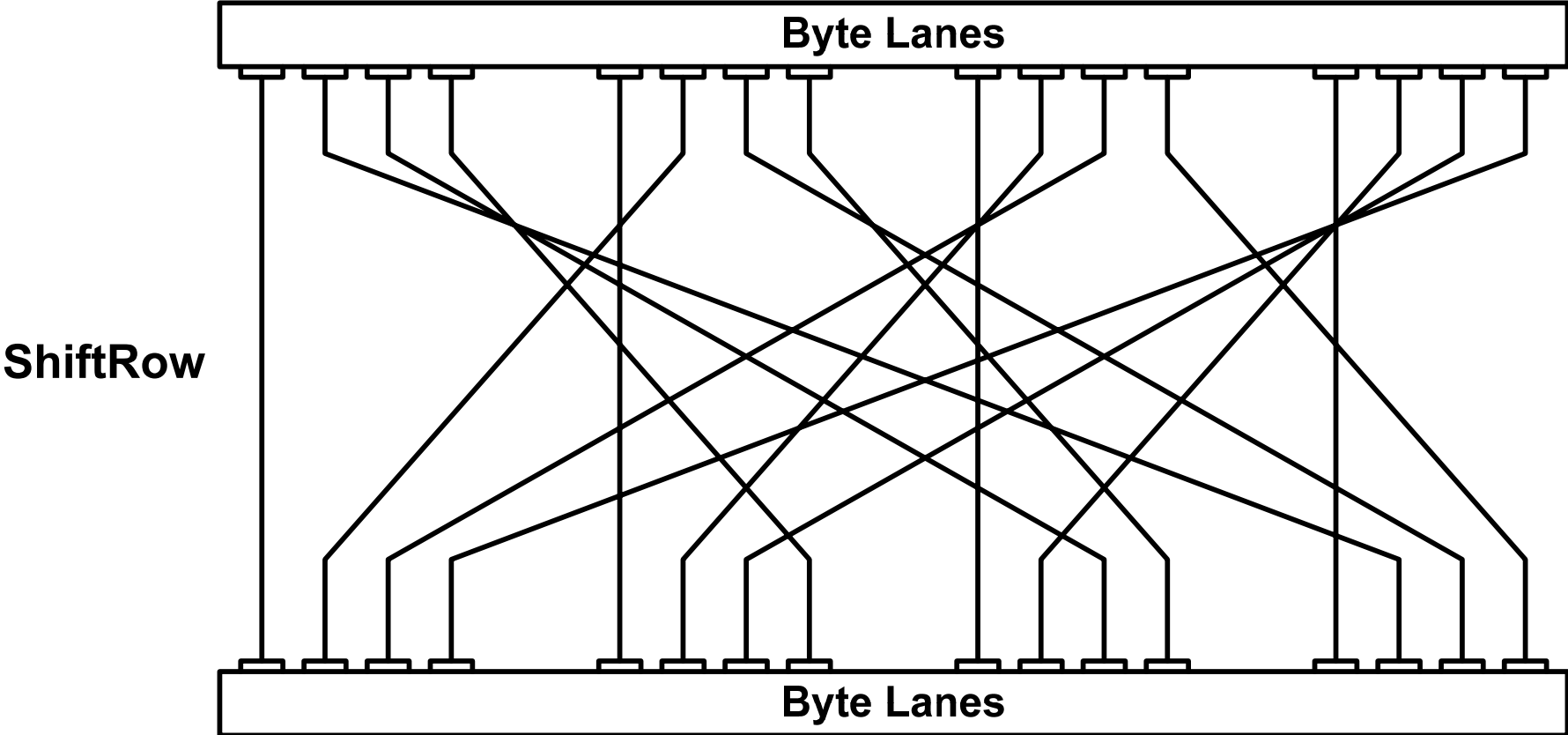
128-Bit SubBytes Transformation

Use lookup tables to perform 16 simultaneous byte substitutions for the 16 data bytes



Saving System Power: Avoid memory and bus traffic associated with retrieving fixed substitution values by implementing the lookup tables as logic within the function unit.

128-Bit ShiftRow Transformation



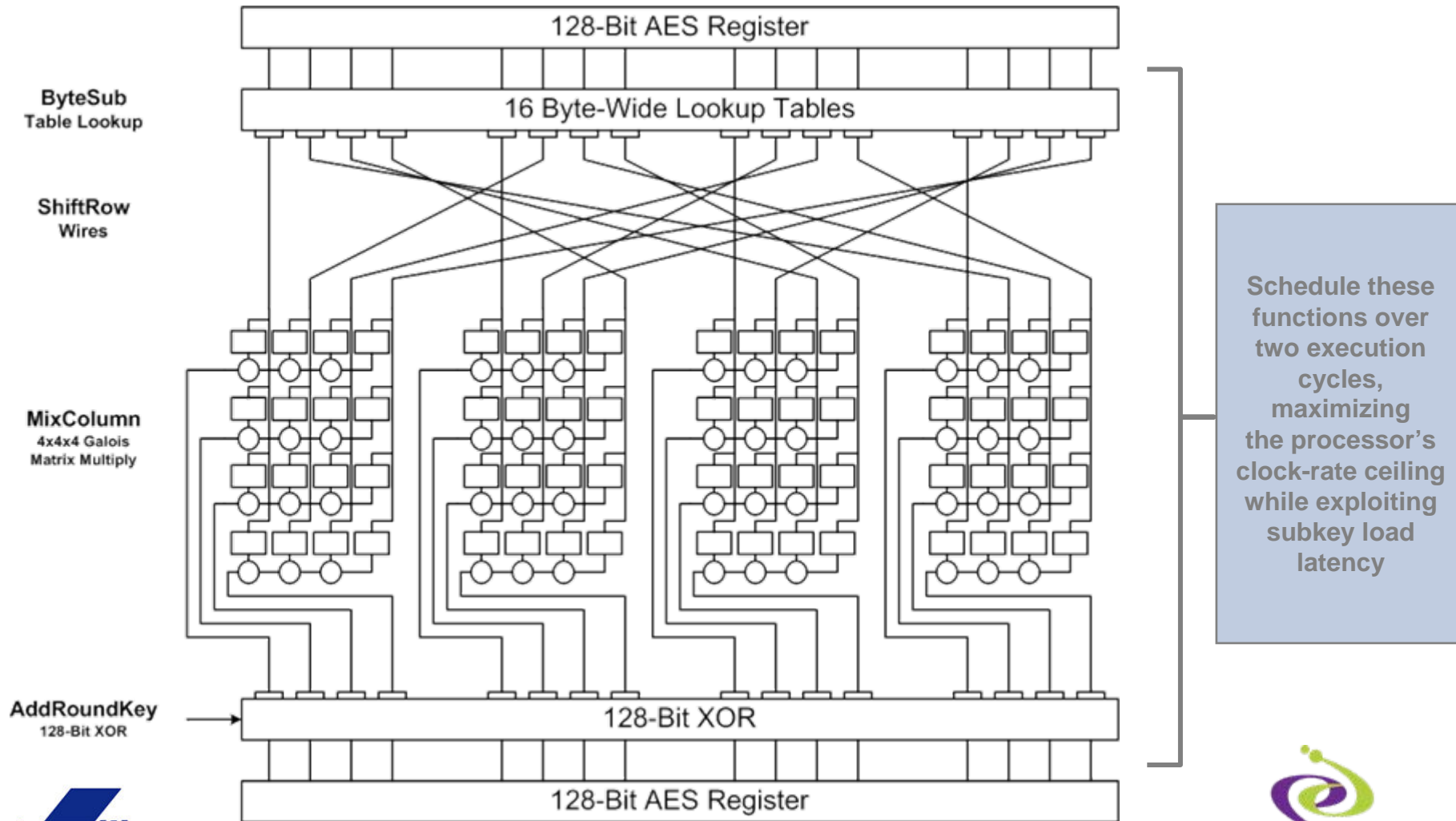
Byte-lane scrambling: No logic. Nothing but wires.

128-Bit MixColumn Transformation

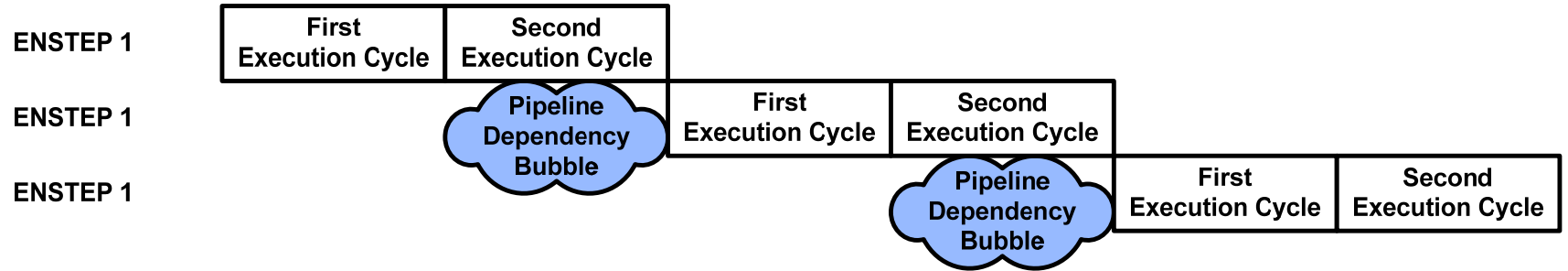
- Use 64 byte-wide Galois multipliers (which use logical operations) to perform the entire matrix-multiplication function in one operation
- Simplify each byte-wide Galois multiplier by exploiting the fact that the multiplicand is always 1, 2, or 3.
 - Multiplicand = 1: use the identity function.
 - Multiplicand = 2: shift the input value left by one bit.
If MSB = 1 after the shift, XOR the intermediate result with x01.
 - Multiplicand = 3: XOR the (multiplicand = 1) value with the (multiplicand = 2) value.

What have we built? An Optimized 128-bit AES Encryption Function Unit and Register

ENSTEP 1 Instruction

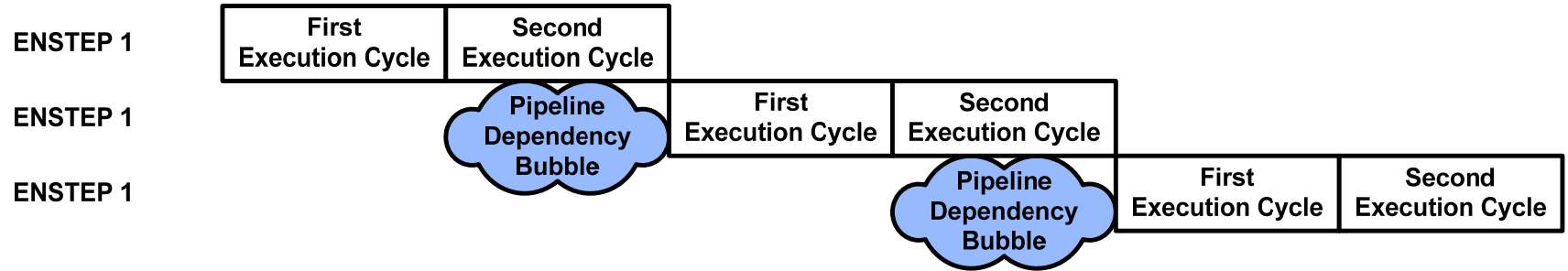


One More Efficiency Trick: Eliminate Execution Pipeline Bubbles

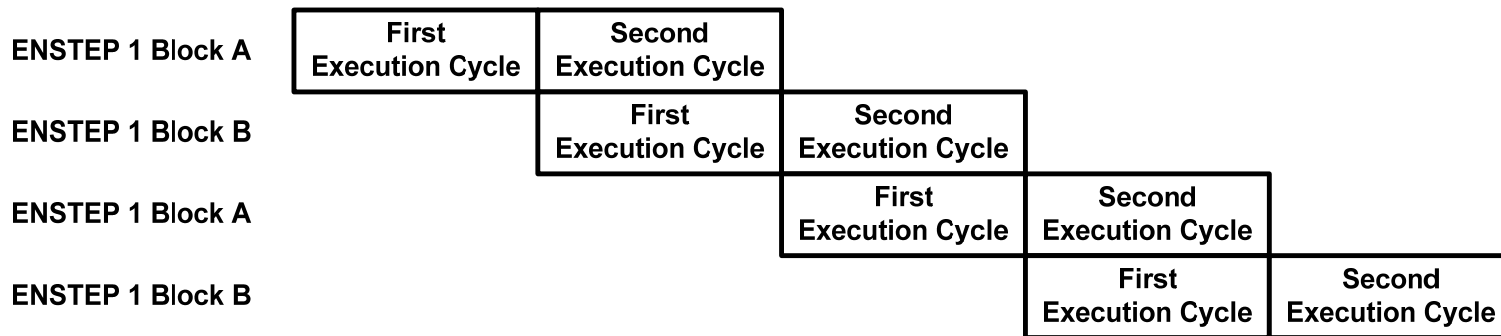


2-Cycle ENSTEP 1 instruction iteration (AES register dependency causes pipeline bubbles)

One More Efficiency Trick: Eliminate Execution Pipeline Bubbles

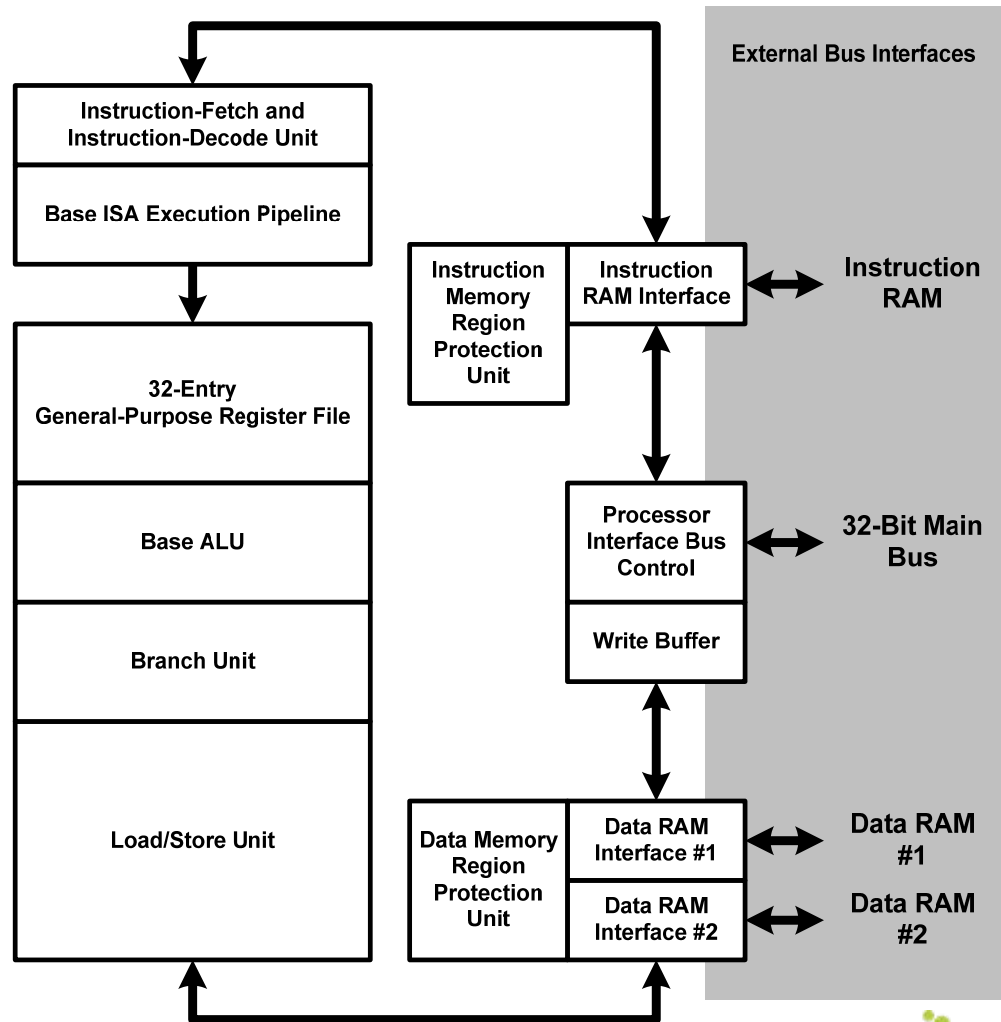


2-Cycle ENSTEP 1 instruction iteration (AES register dependency causes pipeline bubbles)



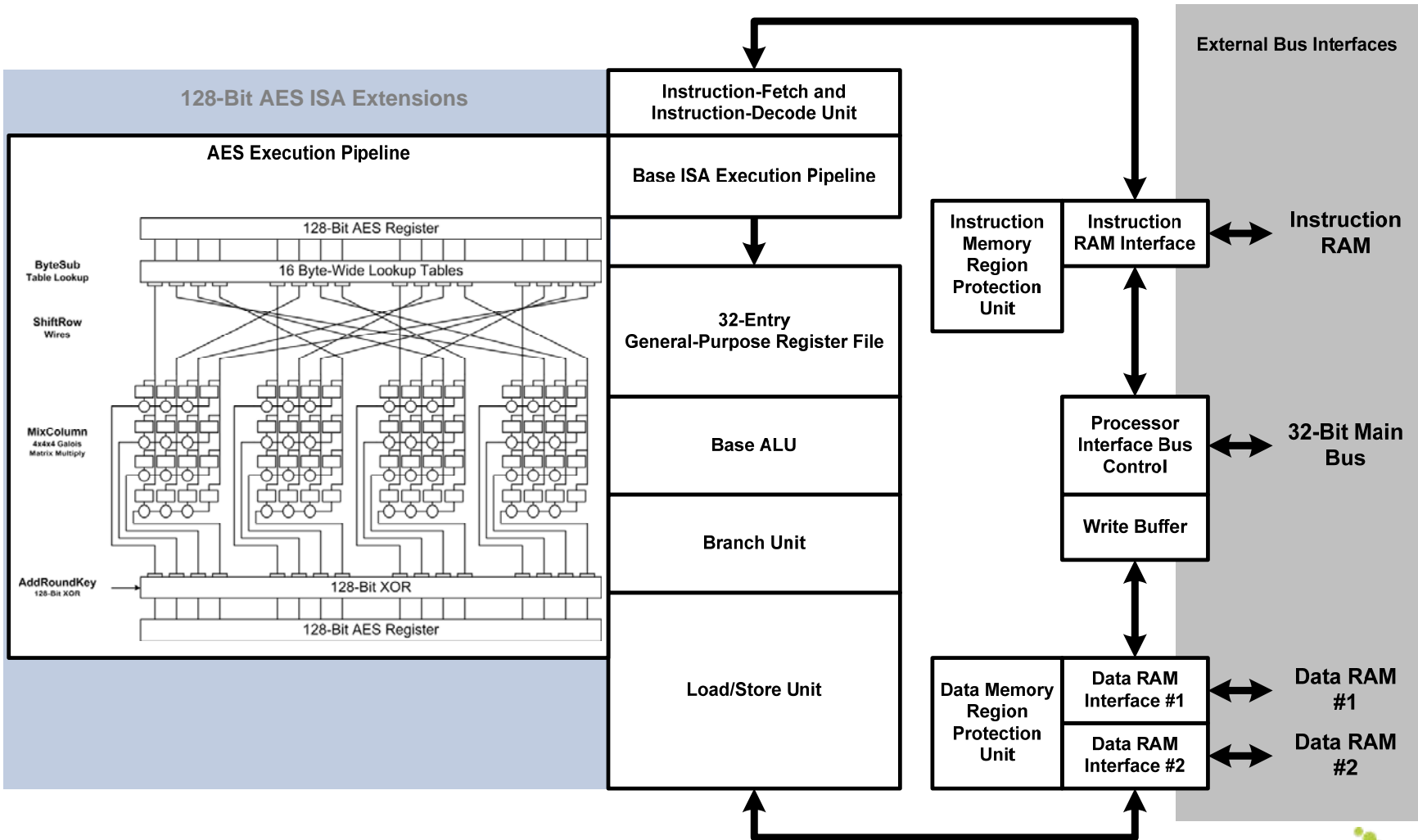
2-Cycle ENSTEP 1 instruction interleaving removes dependency and pipeline bubbles

Resulting Processor Block Diagram



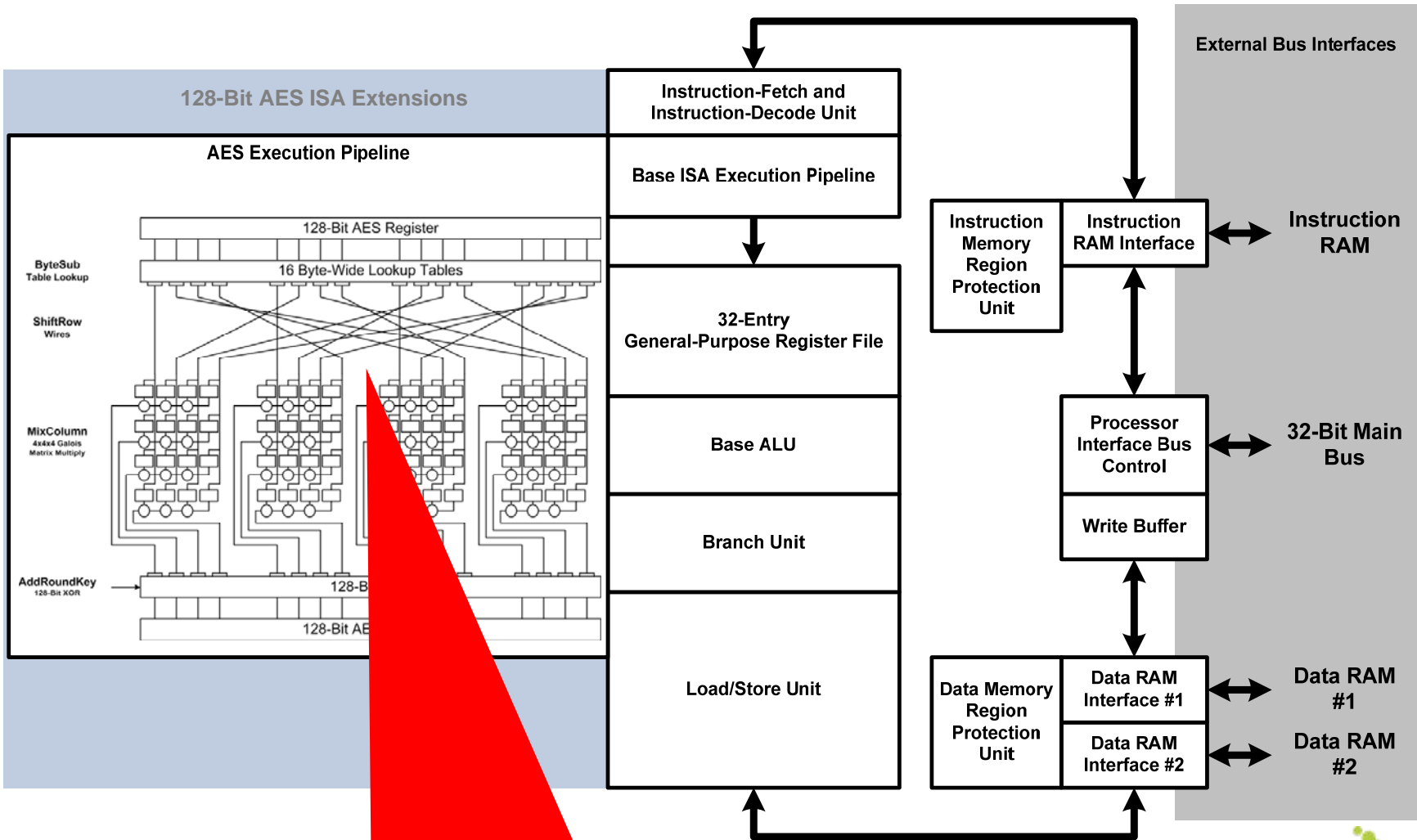
* Caches optional

Resulting Processor Block Diagram



* Caches optional

Resulting Processor Block Diagram

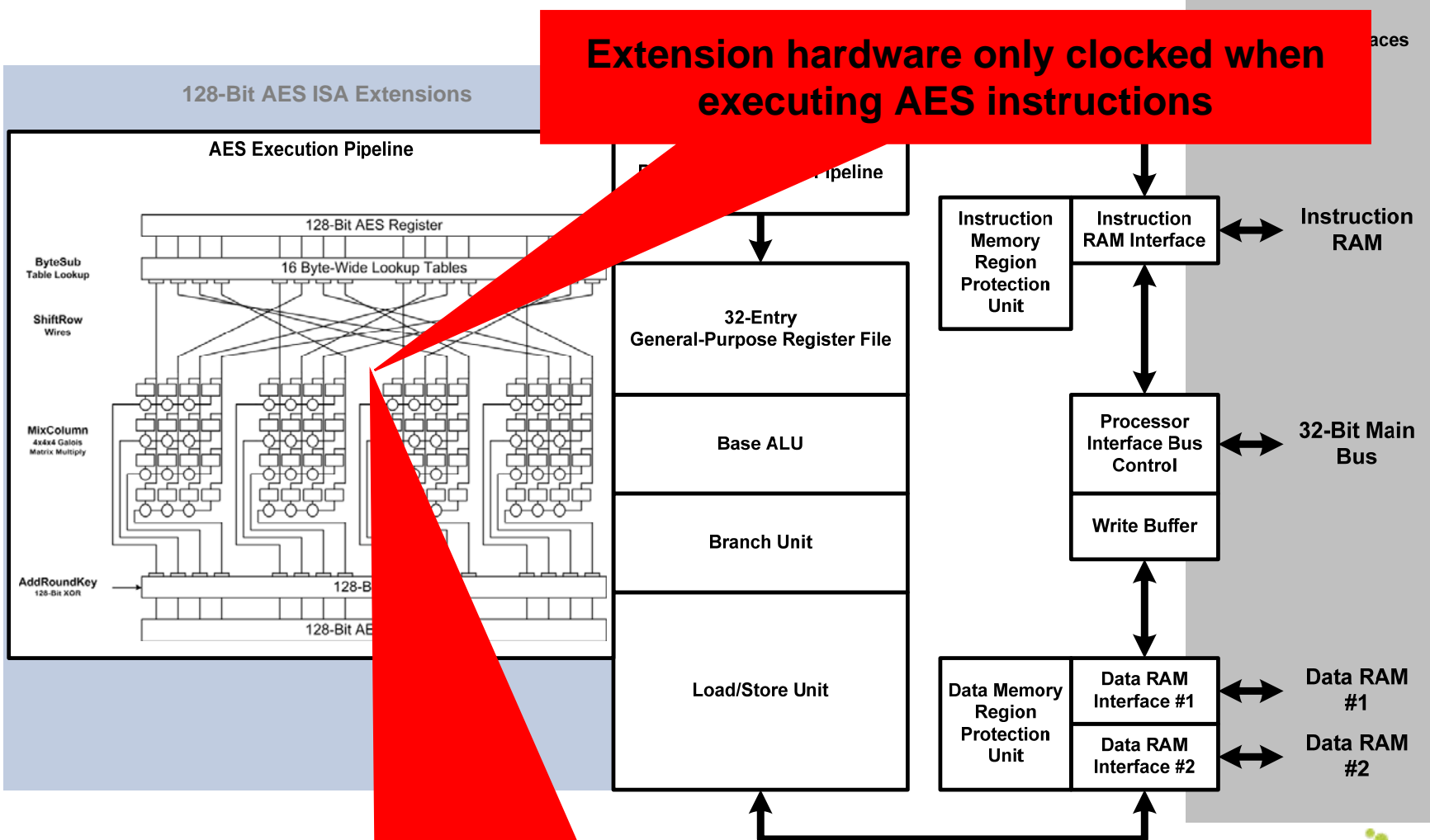


Extension hardware automatically generated from instruction descriptions

* Caches optional



Resulting Processor Block Diagram



Extension hardware automatically generated from instruction descriptions



* Caches optional



Results of ISA Optimization

	Cycle Count (per 10 blocks)		Estimated Energy (uJ)		Estimated Instantaneous Power (mW @ 100 MHz)	
	Straight C	ISA Optimized	Straight C	ISA Optimized	Straight C	ISA Optimized
AES Encryption	353,493	10,713	57.03	1.745	16.13	16.28
AES Encryption and Decryption	679,517	16,966	108.18	2.889	15.92	17.03

Estimates generated by Xenergy energy estimator with the Xtensa ISS.

Notes:

1. Instantaneous power (**power is instantaneous**) increases slightly due to more gates but energy consumption decreases due to fewer cycles.
2. **Estimates do not account for lower core operating voltage due to lower clock rate.**

Results of ISA Optimization

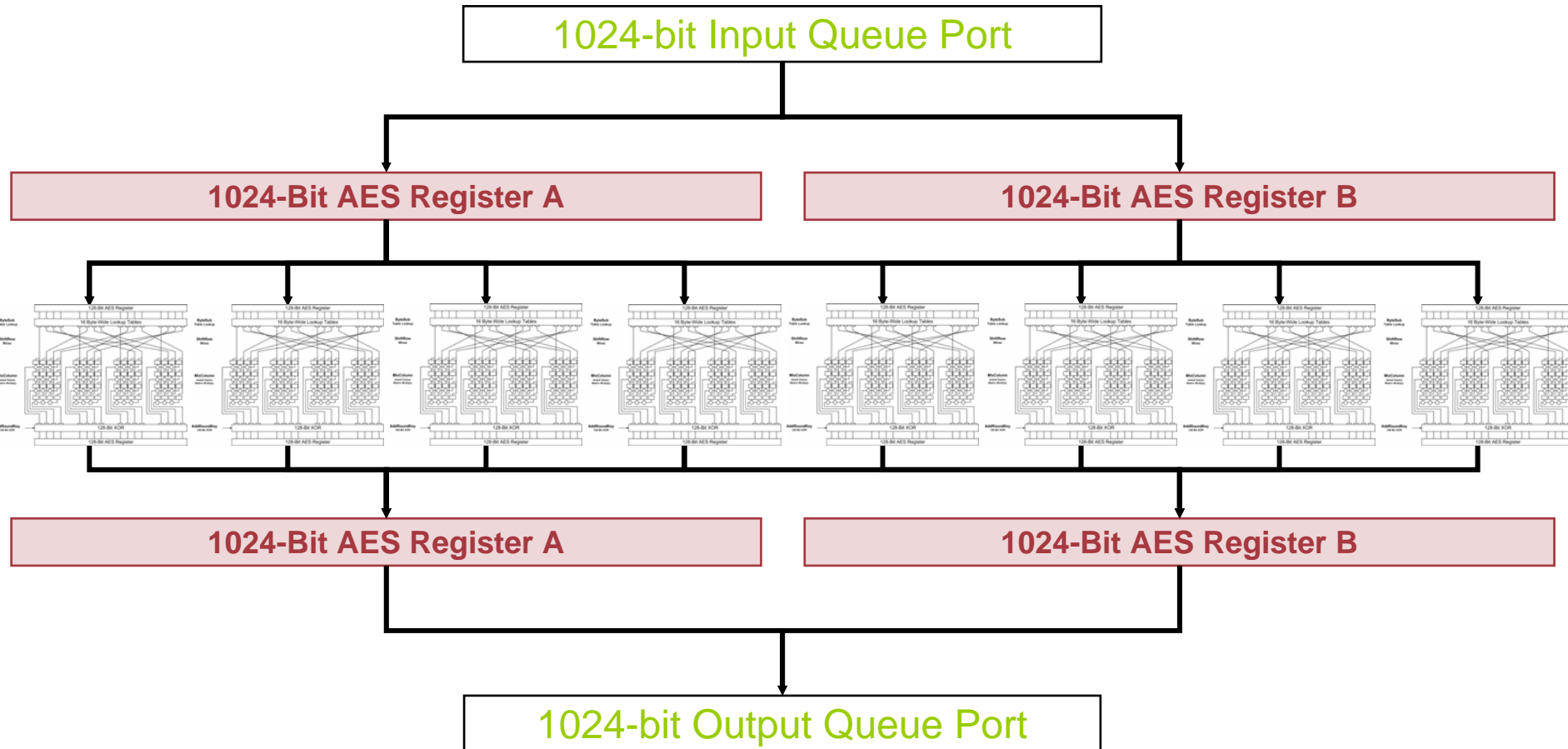
	Cycle Count (per 10 blocks)		Estimated Energy (uJ)		Estimated Instantaneous Power (mW @ 100 MHz)	
	Straight C	ISA Optimized	Straight C	ISA Optimized	Straight C	ISA Optimized
AES Encryption	353,493	10,713	57.03	1.745	16.13	16.28
		30-40x		30-40x		~1x
AES Encryption and Decryption	679,517	16,966	108.18	2.889	15.92	17.03

Estimates generated by Xenergy energy estimator with the Xtensa ISS.

Notes:

1. Instantaneous power (power is instantaneous) increases slightly due to more gates but energy consumption decreases due to fewer cycles.
2. Estimates do not account for lower core operating voltage due to lower clock rate.

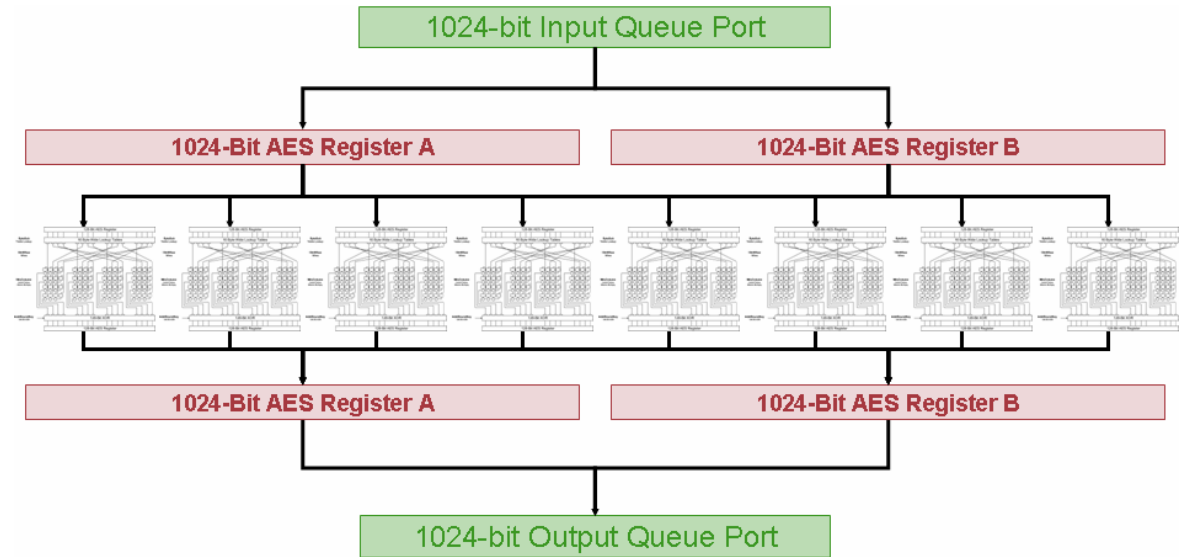
Further 8x Cycle-Count Reduction: 1024-bit AES Registers Fed by External FIFO Queues



Throughput: Unrolled inner loop encrypts ~ 8.5 bytes/cycle
(versus 1.07 bytes/cycle for the bus-based version)

The Outer Limits: Two more Orders of Magnitude

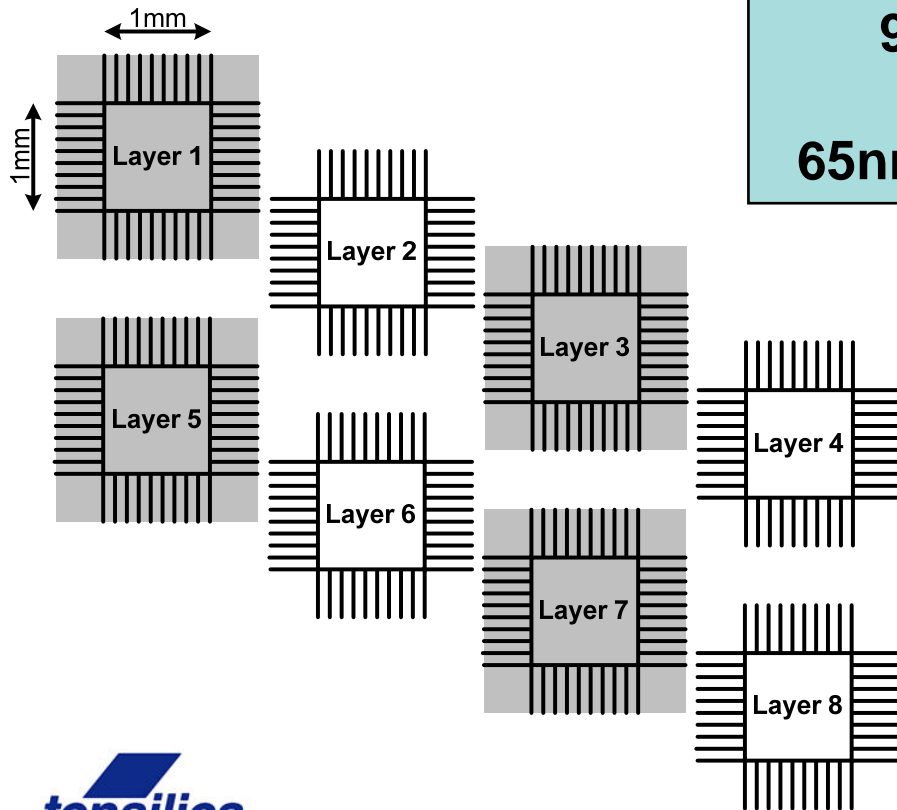
100 x



Throughput: Unrolled inner loop encrypts ~ 850 bytes/cycle
You still need only three new instructions.

Note: Practical gate-count and routing limit is $1 < X < 100X$

Wide Interconnect and Wire Density: What's Practical? Do the Math



8-10 Metal Layers, ITRS 2006 wire spacing

90nm: 100,000+ wires/square mm

65nm: Almost 200,000 wires/square mm

Conclusions: Key Points for AES

- ✓ Two new 128-bit AES registers and three new instructions reduce energy consumption by 30-40x for the AES encryption application
 - ✓ **Scheduled one instruction over two cycles so as not to decrease processor's maximum possible clock rate after synthesis**
 - ✓ **Added second 128-bit AES data register to maintain throughput of one AES instruction per cycle by interleaving operations on two data blocks**
- ✓ More improvement possible if you bypass the processor's bus using wide, direct-access ports to the 128-bit AES registers
- ✓ The RISC processor retains ability to run other application code even with AES-specific ISA extensions

Final Conclusions: Key Points

- ✓ Many, many algorithms lend themselves to these energy-saving design techniques
- ✓ Applying domain-specific expertise nets big reductions in energy used to execute a task
- ✓ Think different: the smallest core doesn't necessarily use the least energy—cycle counts rule!
- ✓ Stay off the bus!!!

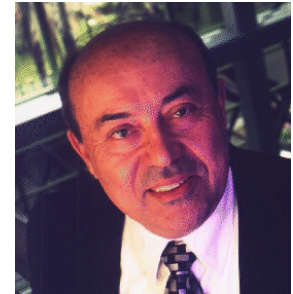
Bonus Example #1: Viterbi Decode

Viterbi Coding

- Determines the most likely path through a state sequence given the presence of noise
- Overcomes noise through spread-spectrum, redundancy, and convolutional coding
- Used for cellular phone reception

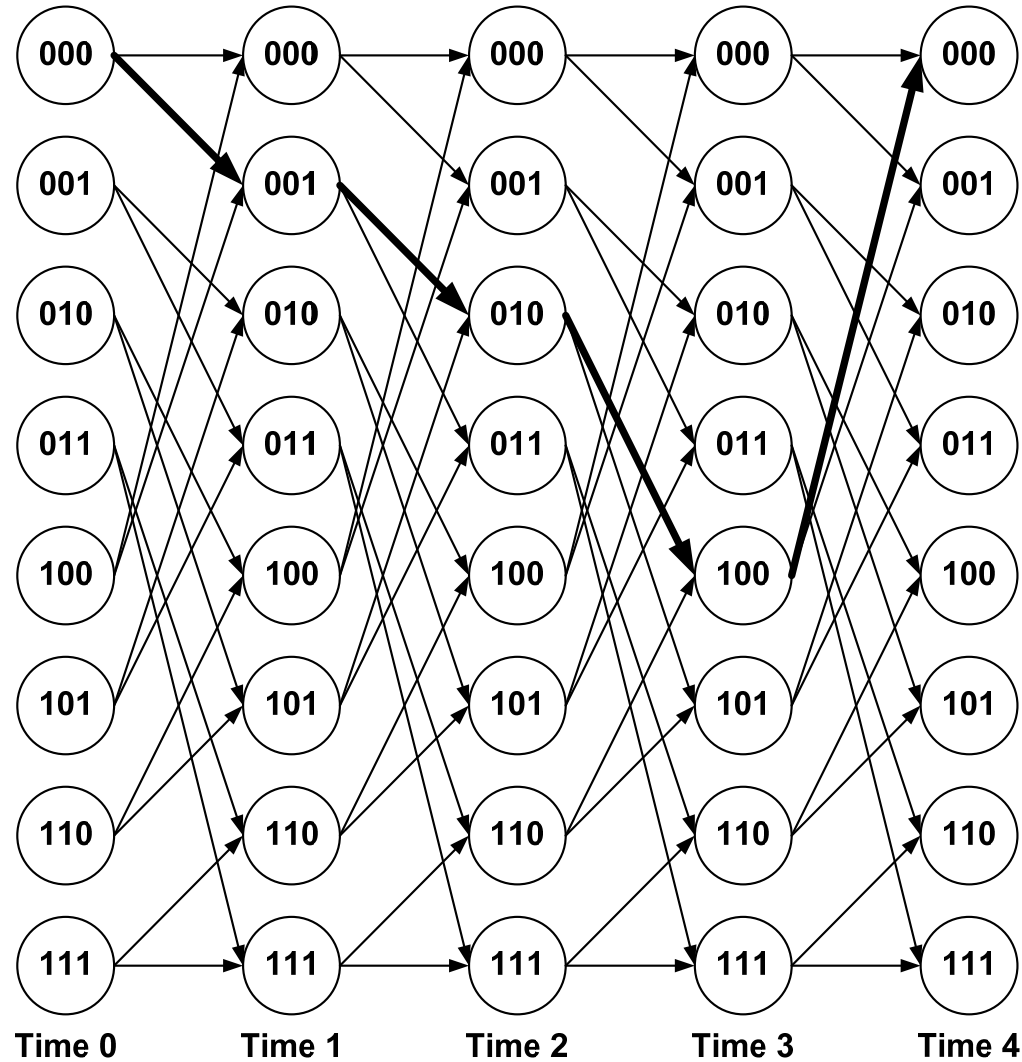


Hedy Lamarr
Invented Spread-Spectrum Communications
in 1942. Declassified in 1981.



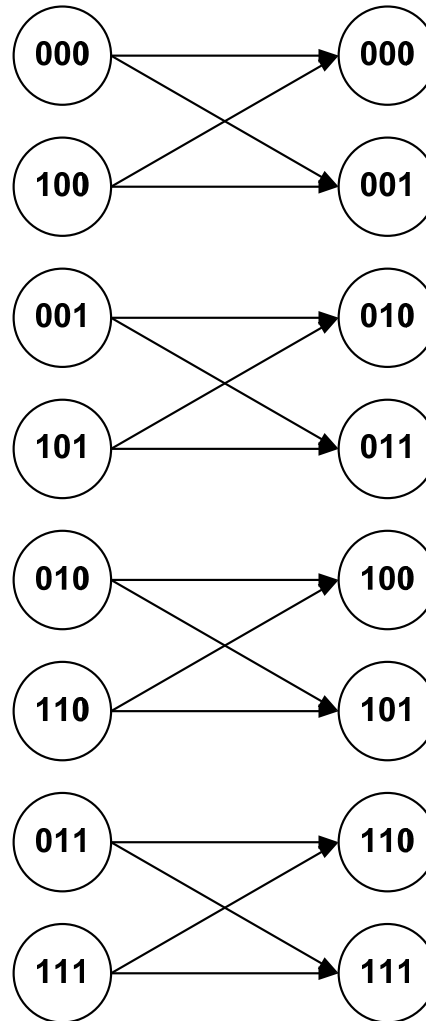
Dr. Andrew Viterbi
Co-Founded QUALCOMM in 1985

Viterbi Trellis: Traversing the States



Tracing the Most Likely Path Through States

Viterbi Butterfly: Two Sources, Two Destinations, Four Arcs per Group



**Select next
state using a
computed
distance
metric**

Results of ISA Optimization

	Cycle Count		Estimated Energy (uJ)		Estimated Instantaneous Power (mW @ 100 MHz)	
	Straight C	ISA Optimized	Straight C	ISA Optimized	Straight C	ISA Optimized
Viterbi Decode	279,537	7632	65.69	2	23.5	26.2

Estimates generated by Xenergy energy estimator with the Xtensa ISS.

Notes:

1. Instantaneous power (**power is instantaneous**) increases due to more gates but energy consumption decreases due to fewer cycles.
2. **Estimates do not account for lower core operating voltage due to lower clock rate.**

Results of ISA Optimization

	Cycle Count		Estimated Energy (uJ)		Estimated Instantaneous Power (mW @ 100 MHz)	
	Straight C	ISA	Straight C	ISA	Straight C	ISA
Viterbi Decode	279,537	7632 37x	65.69	2 33x	23.5	26.2 ~1.15x

Estimates generated by Xenergy energy estimator with the Xtensa ISS.

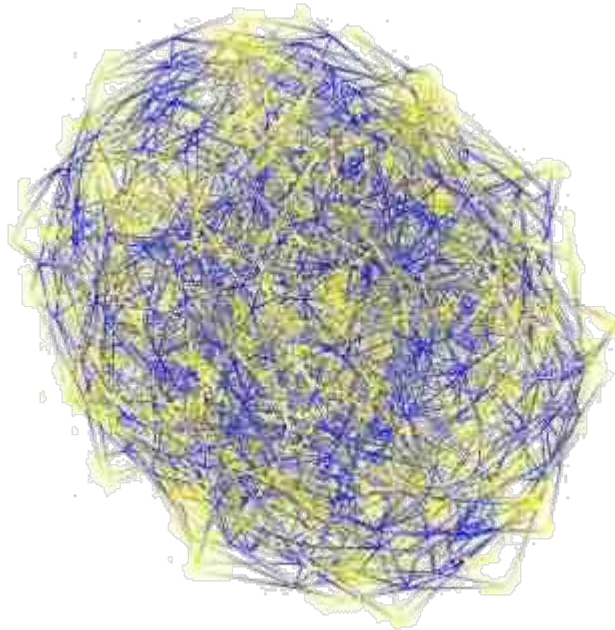
Notes:

1. Instantaneous power (power is instantaneous) increases due to more gates but energy consumption decreases due to fewer cycles.
2. Estimates do not account for lower core operating voltage due to lower clock rate.

Other Coding Algorithms are Equally Targetable

Related convolutional codes

- Turbo (3G cellular, deep-space communications)
- LDPC (Low-Density Parity Check, 3GPP cellular)



LDPC Minimum Distance Graph
Professor Oscar Y. Takeshita
Ohio State University

Bonus Example #2: FFT

Fourier Transform

- Decomposes signals into frequency components
- Very “mathy”
- Foundation of DSP

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt,$$



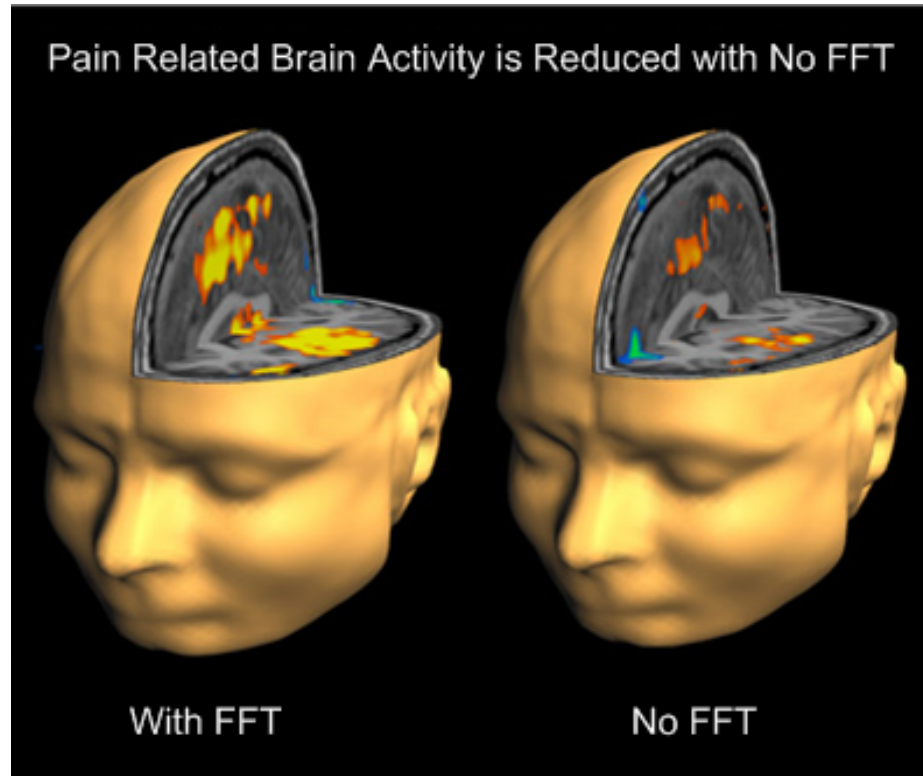
Joseph Fourier
French Transformer

Fast Fourier Transform

- Fast algorithm for computing discrete Fourier transform

$$X_{\mathbf{k}} = \sum_{\mathbf{n}=0}^{\mathbf{N}-1} e^{-2\pi i \mathbf{k} \cdot (\mathbf{n}/\mathbf{N})} x_{\mathbf{n}}$$

FFTs Hurt My Brain

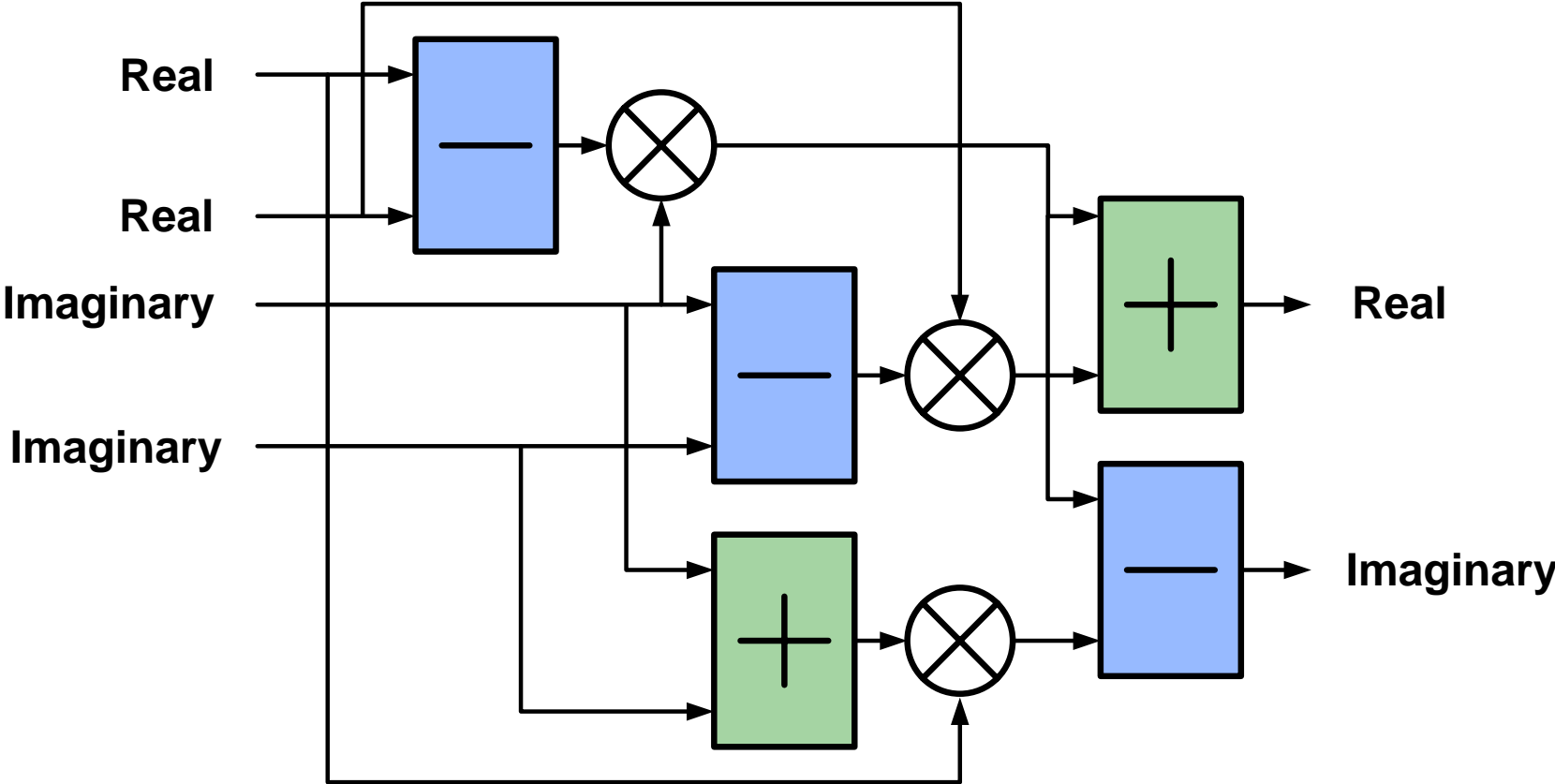


FFT for 802.11g Wireless PHY

- Each FFT to be completed in 3.2 usec
 - 64-point
 - decimation-in-frequency
 - 16-bit real/16-bit imaginary complex data
- Radix-4 FFT Butterfly requires:
 - Twelve 16x16-bit multipliers
 - More than twenty 16-bit adders



16-Bit FFT Complex Multiplier (need 4)



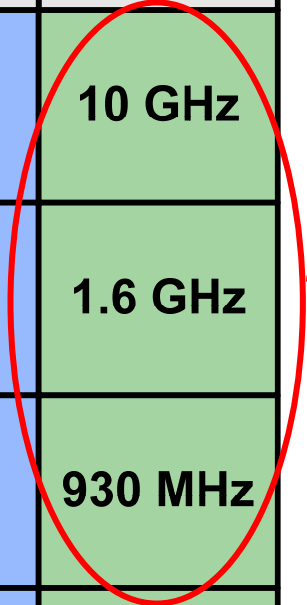
Results of ISA Optimization

64-point, 16-bit Decimation in Frequency Complex FFT	Cycle Count		Estimated Energy (uJ)
	One FFT in 3.2 usec	Required Clock Rate	
Straight C	32187	10 GHz	3,450
Add 32-bit Multiplier	5071	1.6 GHz	575
Multiple Instruction Issue and 32-bit Multiplier	2975	930 MHz	620
Radix-4 FFT ISA Extension	146	46 MHz	56



Results of ISA Optimization

64-point, 16-bit Decimation in Frequency Complex FFT	Cycle Count		Estimated Energy (uJ)
	One FFT in 3.2 usec	Required Clock Rate	
Straight C	32187	10 GHz	3,450
Add 32-bit Multiplier	5071	1.6 GHz	575
Multiple Instruction Issue and 32-bit Multiplier	2975	930 MHz	620
Radix-4 FFT ISA Extension	146	46 MHz	56



Impossible using synthesized logic even with 65nm ASIC fabrication technology



217x

62x