



Resolving the Grand Paradox: Low Energy and Full Programmability in 4G Mobile Baseband SOCs

Chris Rowen

Founder and CTO, Tensilica

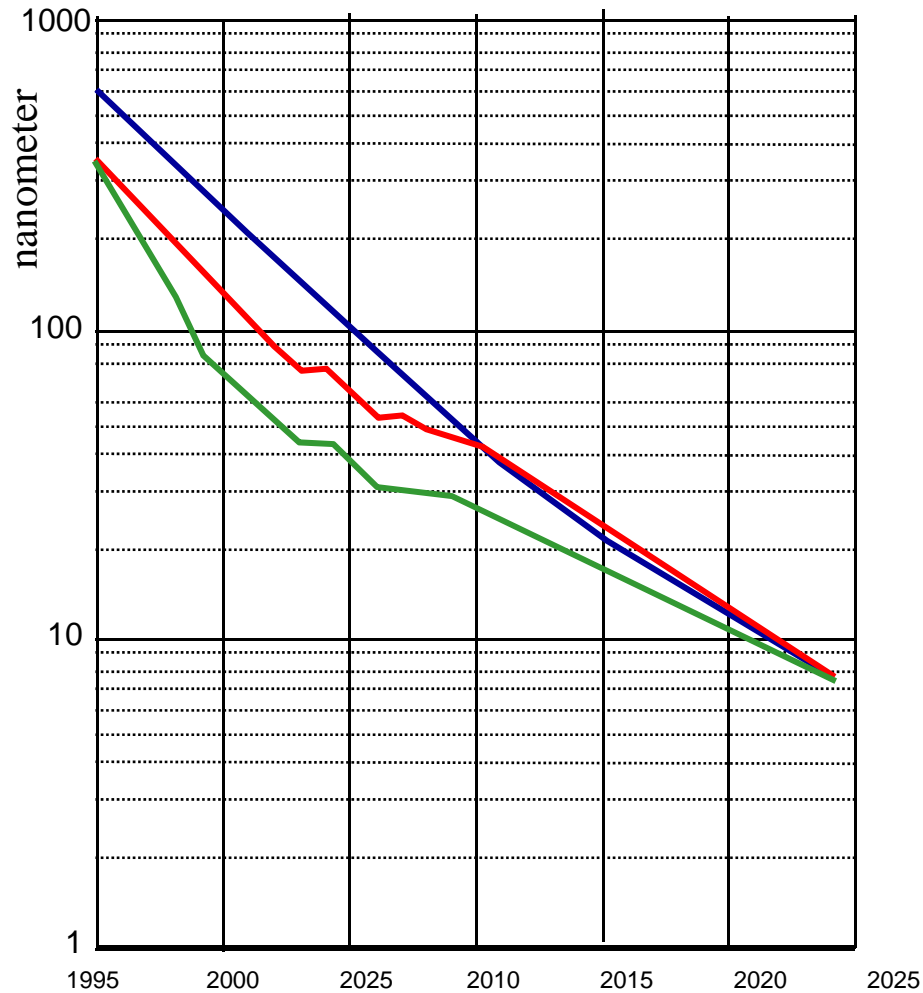
15 April 2010

Outline

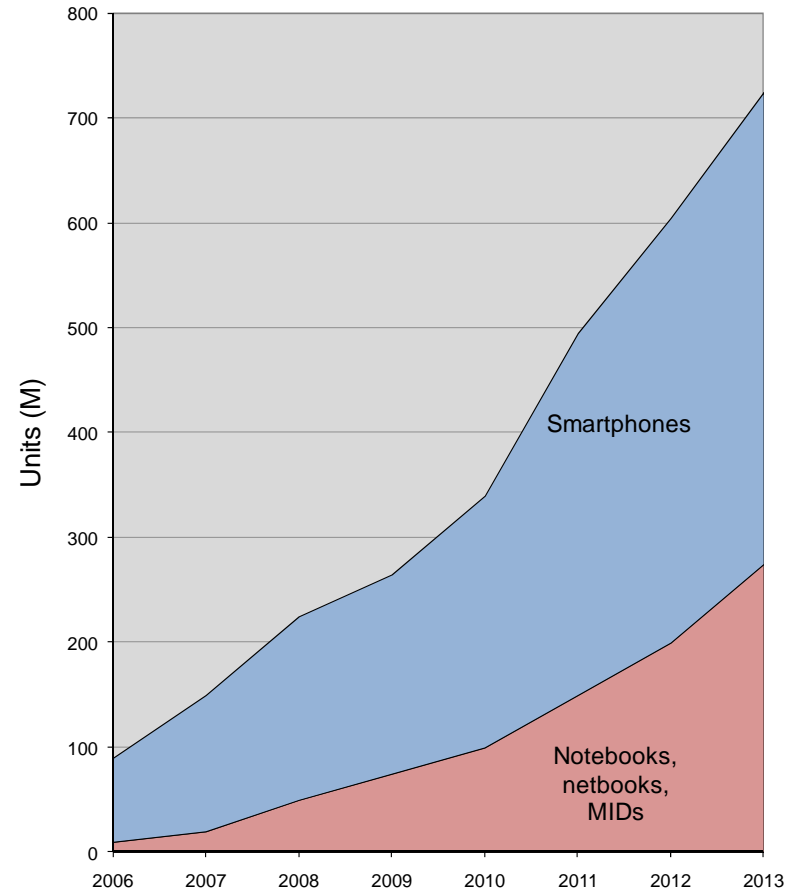
- **Two Exciting but Frightening Curves**
- **A Grand Challenge Problem: LTE Handsets**
- **A Family of Building Blocks**
- **Modular Approach**
 - Core1: BBE16
 - Core 2: SSP16
 - Core 3: BSP3
 - Core4: Turbo16
- **LTE Reference Architecture “Atlas”**
- **Atlas Area and Power Analysis**
- **Wrap Up**

Two Curves

2009 ITRS MPU/ASIC Metal 1 half pitch
 2009 ITRS MPU Printed Gate Length
 2009 ITRS MPU Physical Gate Length



Mobile Broadband Terminal Units



Two Curves

Why Are These Curves Exciting?

- Incredible density mean true single-chip integration: many coordinated functions for nearly unbounded mobile experience
- High volume on range of semiconductor designs
- New opportunities for processors: Bigger, smaller, faster, more specialized, lower power, many subsystems of many cores

Why are These Curves Frightening?

- Moore's Law enables high density, but how do we cope with the absolute complexity and rate of change in such integrated designs?
- Many required functions, but few viable single-function chips
 - Silicon platforms must integrate or die:
 - Example: "Single-chip phone"
 - apps processor + N basebands + audio + video + graphics+ networking+ memory+ more memory+N RF/transceivers+ sensors +..
- So many transistors, so little battery capacity
 - Steeper improvement in density than energy efficiency (new transistor types?)

Building Solutions to Solve Problems

- The raw material: more efficient processors
- A range of building blocks: baseband processors
- A solution architecture: LTE Reference Architecture



A Grand Challenge Problem: LTE Handsets

Design Goals

- High data rates: 150Mbps DL/50Mbps UL
- High spectral efficiency with
 - Orthogonal Frequency Division Multiplexing
 - Multiple-Input Multiple Output
- Scalable bandwidth: 1.25MHz to 20 MHz
- Both Frequency-Division and Time Division Duplex
- All IP Networks

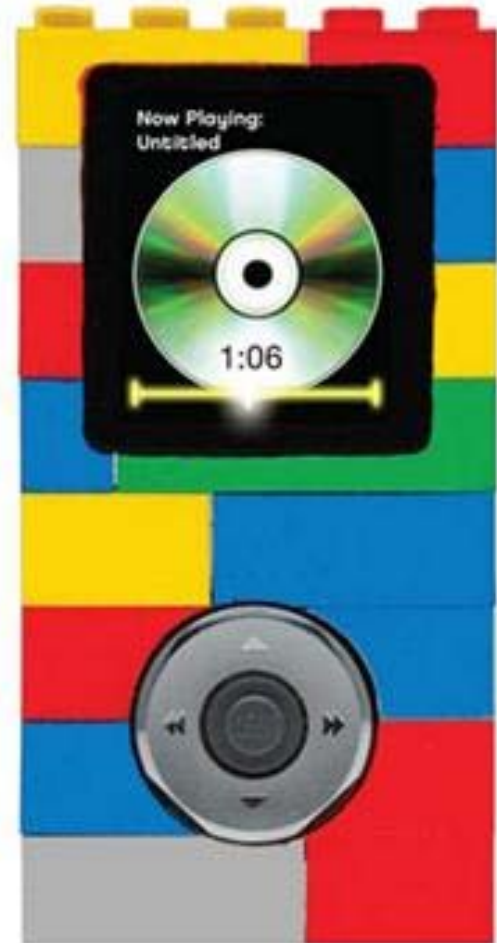
Network Layer Features

- Co-existence with legacy standards (e.g. call transfer)
- 200 active users in 5MHz bandwidth
- Sub 5-ms latency for small IP packets

Implementation Needs

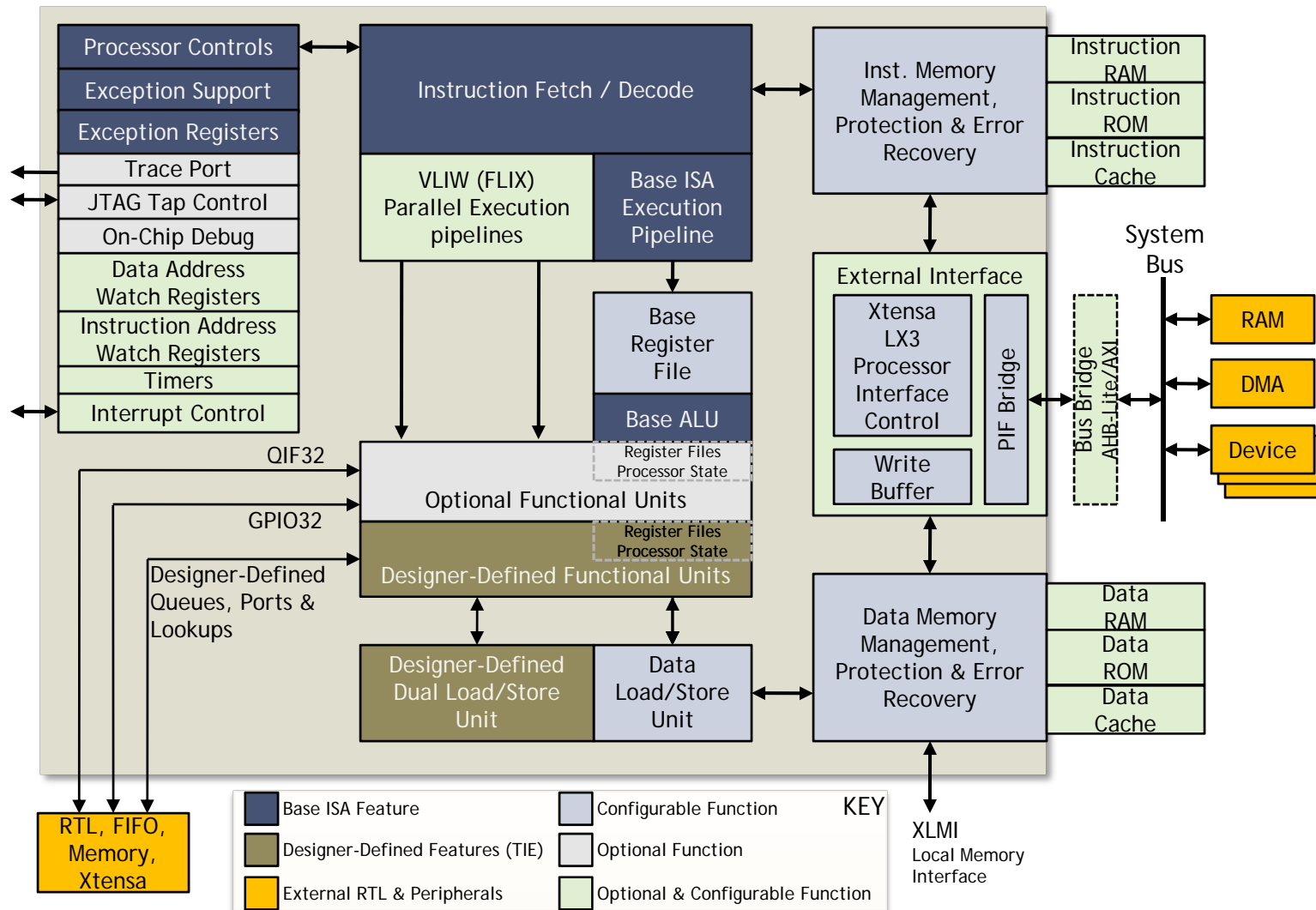
- Low silicon cost: $<20\text{mm}^2$ for digital PHY
- Low power: $< 2\text{mW}/\text{Mbps DL}$

Paradox: Reach performance and power goals while building a programmable system



The Essential Building Block

Xtensa LX3 Extensible Dataplane Processor

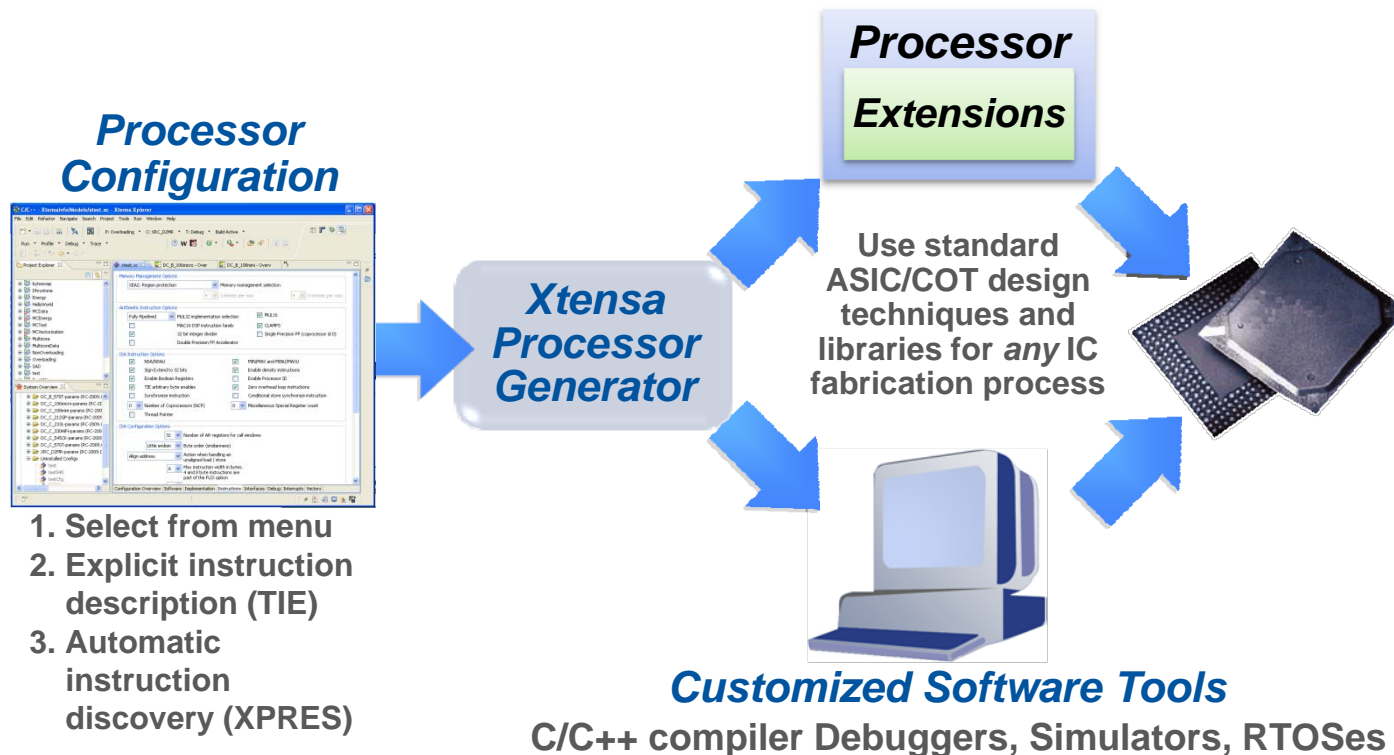


The Essential Automation Xtensa Processor Generator



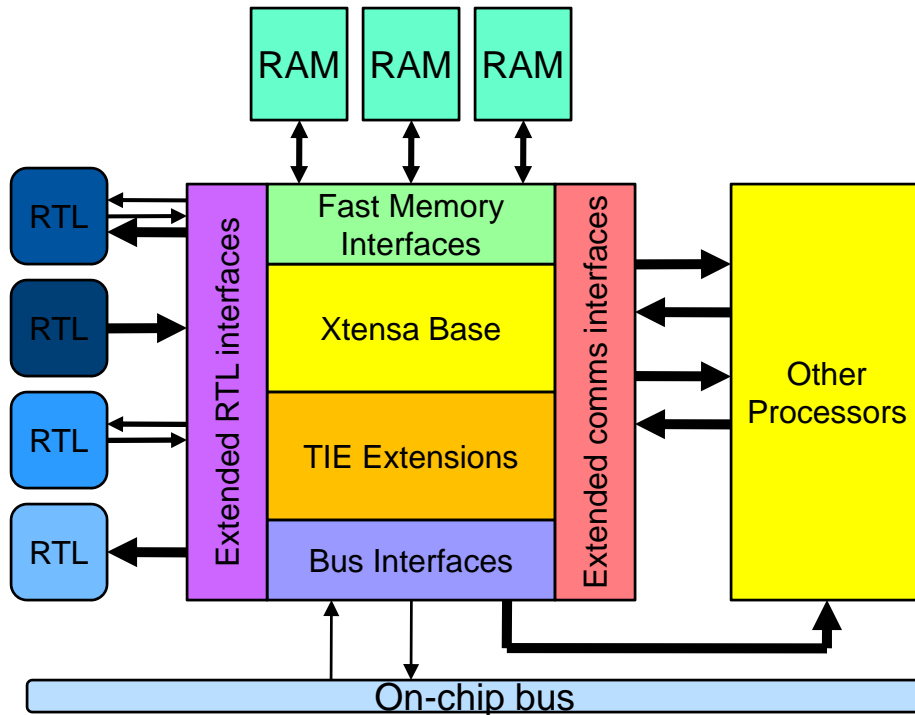
Complete Hardware Design

Source pre-verified RTL, EDA scripts, test suite



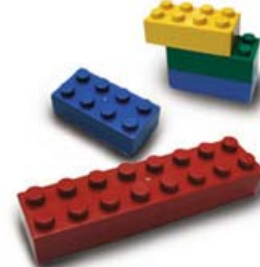
C/C++ compiler Debuggers, Simulators, RTOSes

Multiple Core Communication = Interconnect + Software



Software for Multi-core:

- Modeling at every level:
 1. Gate/RTL level
 2. FPGA netlist
 3. Pin exact
 4. Cycle accurate
 5. Instruction accurate
- Multi-core debug and analysis
- Communications APIs
 - Native message passing with hardware queues
 - Shared memory
 - Energy-optimized
- Complete DSP Libraries
- Solution stacks for LTE
 - L1 PHY
 - L2/L3 MAC, RLC



Scalable platform for all design approaches

Deeply integrated Task Engines

(Xtensas performs specific tasks in HW flow)

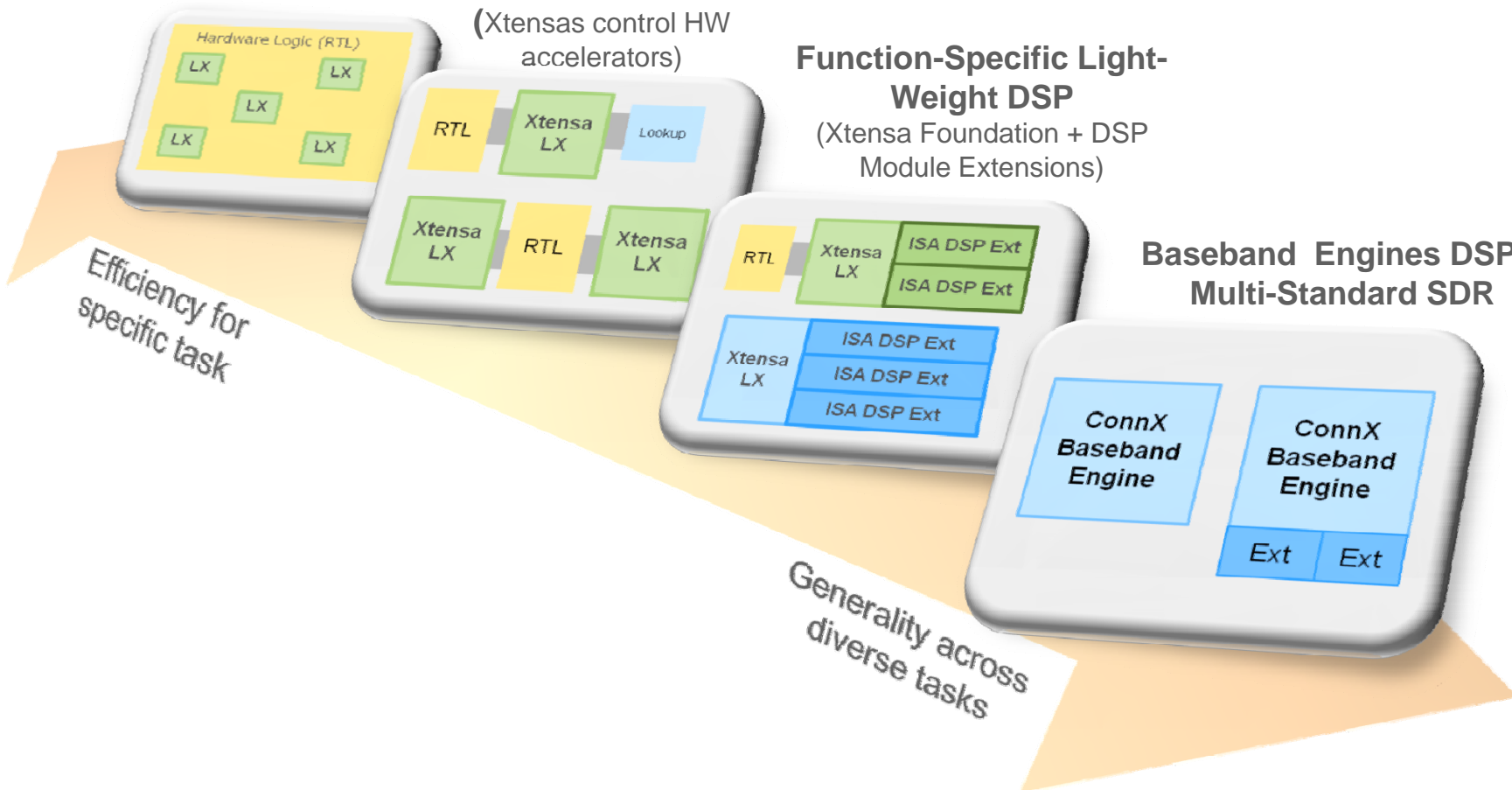
Small, programmable DPU/DSP

(Xtensas control HW accelerators)

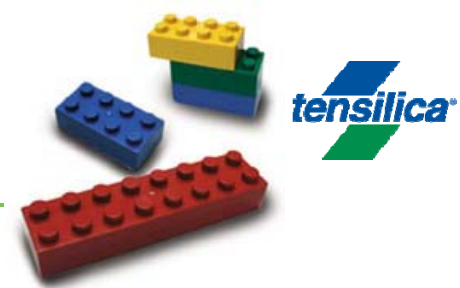
Function-Specific Light-Weight DSP

(Xtensa Foundation + DSP Module Extensions)

Baseband Engines DSPs, Multi-Standard SDR



A Range of Extensible Cores



Controller for RTL

Tiny: <16K gates
(25,000 μm^2)

Fast: >1GHz base

Configurable memory
interface, buses, interrupts,
timers, instructions

Unique extensible RTL
interfaces:

- **input/output queues**
- **input/output wires**
- **lookups**
- Moves error-prone state machines into C
- Lowest-energy connection between hardwired functions and processors

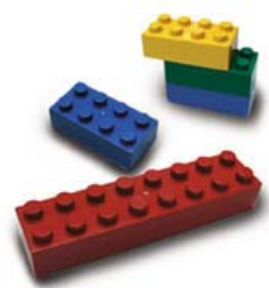
Specialized DSP

- High memory and operation bandwidth DSPs
- Exact fit to data types: 18b+18b, bytes, bits
- Combine programmable processing with RTL interface
- Standard DSP options +widely-used Tensilica Instruction Extension (TIE) format
- Co-generation of simulator, compilers, RTL, verification, multi-core models and energy profiles

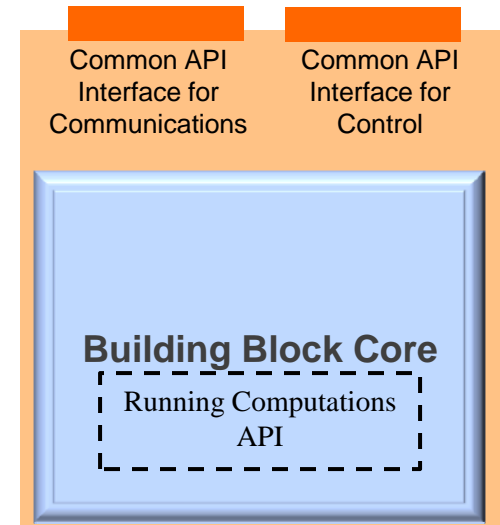
Baseband Engine DSP

- Fastest production DSP cores in industry
- High operation throughput: SIMD+VLIW
- Vectorizing compiler for native integer, fractional and complex data-types
- Broad libraries: common functions plus baseband stacks
- Further extension by customers

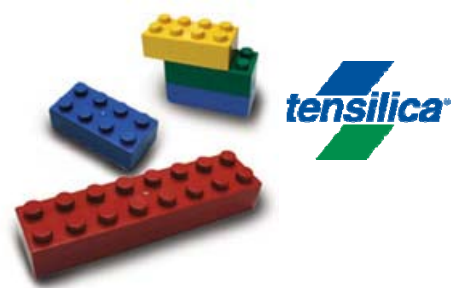
Modular Core Approach



- **Well-defined interfaces**
 - Hardware: Bus Interfaces, Queue Interfaces
 - Software: Shared memory and message passing Communications and Control API
- **Simple Software Use Model**
 - Initial software porting and debug on single core implementing superset of functions
- **Easier to Customize**
 - Individual processors can be replaced with users own implementation
 - Legacy Processor
 - Hardwired RTL blocks
 - Part of algorithm further accelerated with RTL
 - Direct attached to processor core for highest bandwidth

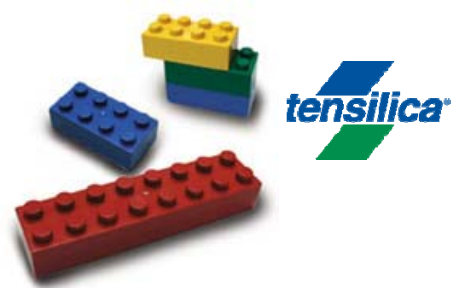


Integrated Power Management



- Clock gating and standby modes using integrated **Power Management Module (PMM)**
- Instruction and interface support for **deep sleep** modes
 - WAITI – Instruction control into Deep sleep. Wakeup on Interrupt
 - Runstall – External control into Deep sleep. External wakeup
- Small code size **reduces memory power**: modeless switching between among instruction sizes
- **Fine-grained power management** for control and data-oriented code types
 - Cycle-by-cycle enables for memory banks and load/store units
 - VLIW slot architecture shut down when not being used
 - Pipeline resource disabled when instruction state does not require
 - Instruction execution logic clock stopped when no in use
- Optimized Instruction Set Architecture: operation parallelism means instructions require less cycles to execute algorithms → **lower MHz**
- Multiple cores offer **better granularity** of switching off processors per operating mode
 - Demonstrated ease of dynamic voltage and frequency management in multi-core

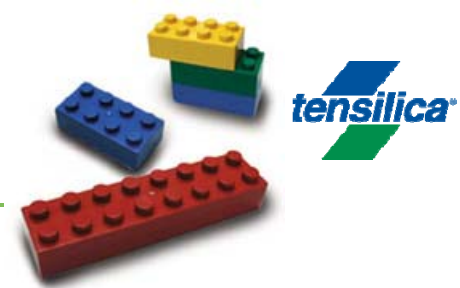
Core 1: ConnX BBE16 DSP Engine



Ultra-High Performance and Programming Ease

- 16-way MAC Architecture with hybrid SIMD / VLIW
 - 8-way SIMD offers high data computation rate for DSP algorithms
 - 3-way VLIW allows multiple independent operations to execute concurrently
- Optimized to give performance for DSP applications
 - Rich for most demanding DSP algorithms
 - OFDM based algorithms
 - Optimized FIR and Matrix computation
 - Full throughput with extended precision with 40-bit MAC operations
 - Optimized DSP Libraries available
 - C programmability through industry leading compiler technology
- Small size and low power
 - Fully optimized for size and power sensitive applications
 - 0.71mm² core area (40LP at 300MHz) for XRC_D16PM
 - 0.24mW/MHz for FFT computation (40LP at 300MHz) for XRC_D16PM

Core 1: ConnX BBE16 Baseband Engine

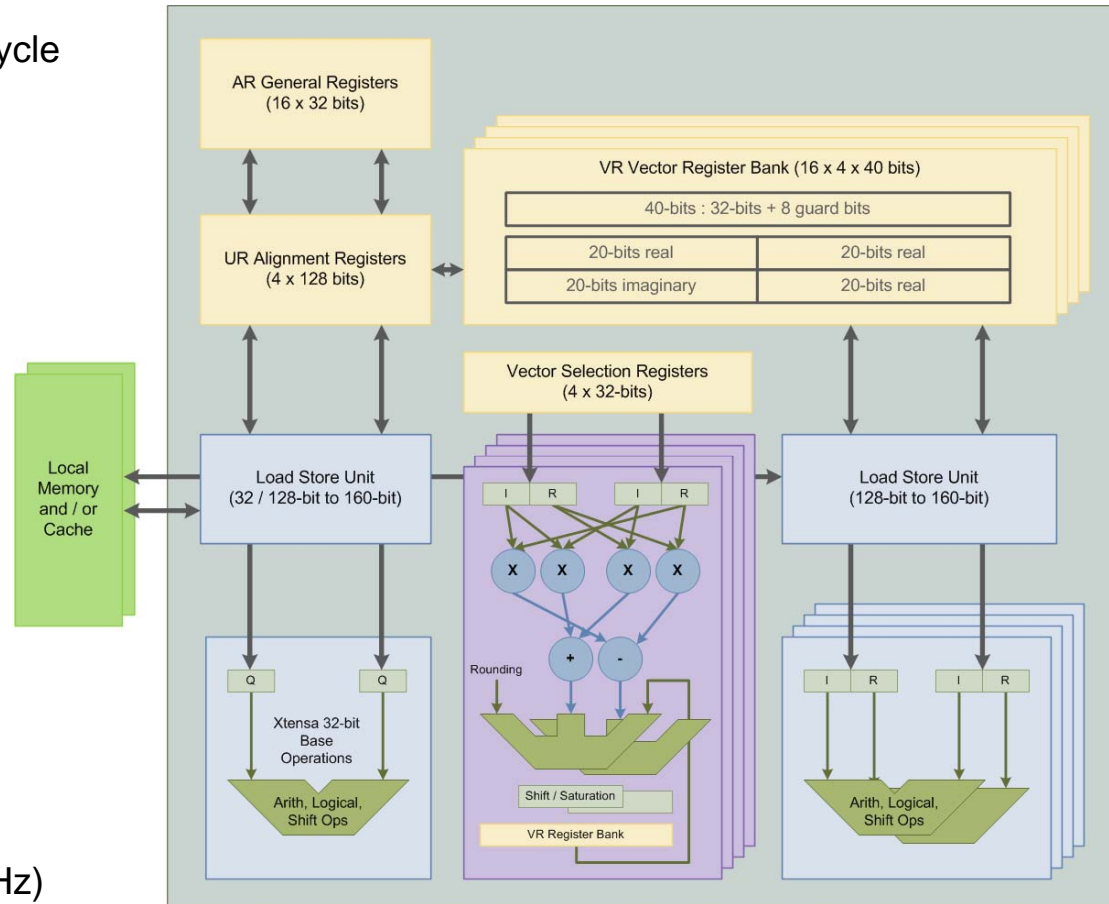


Architecture:

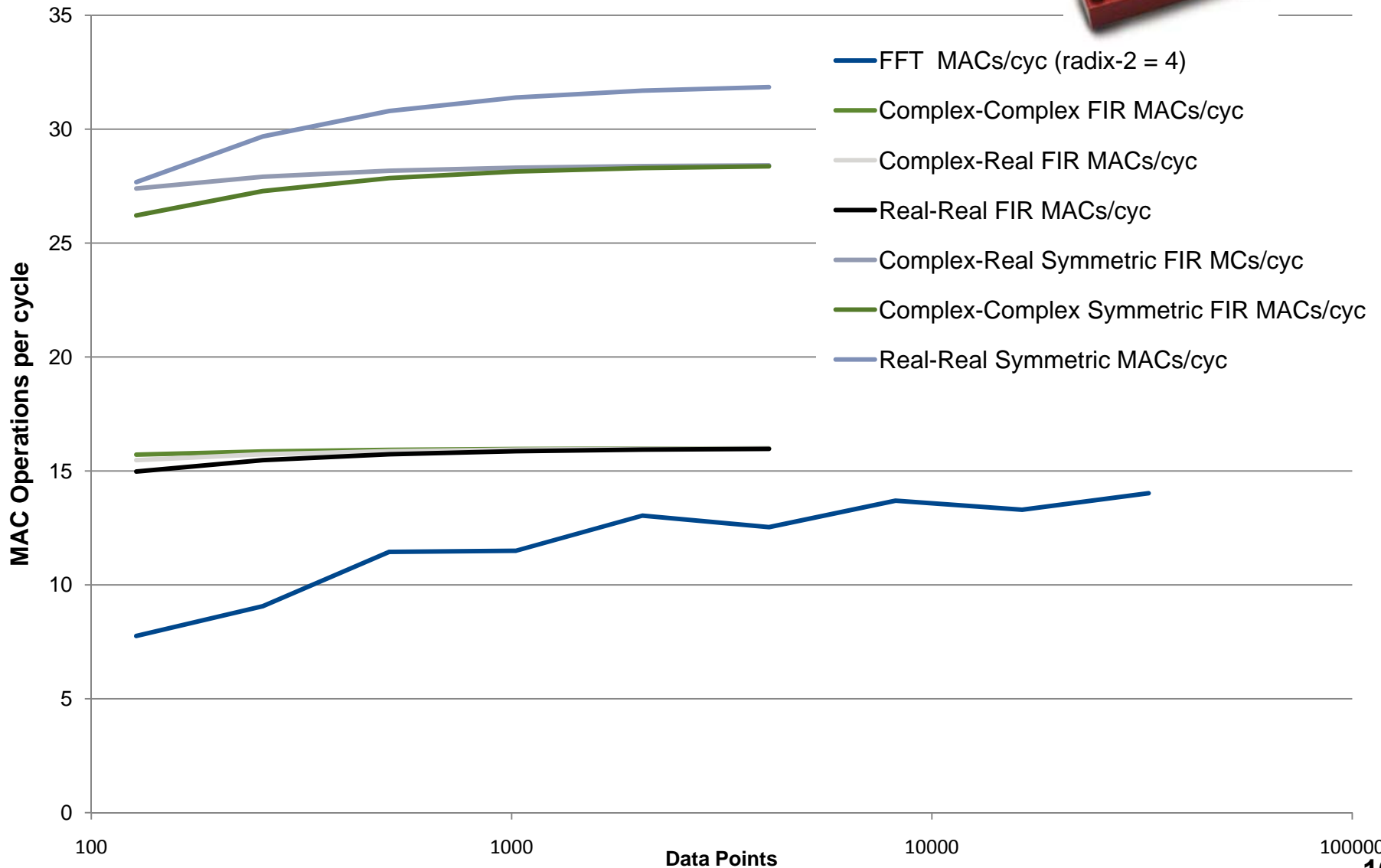
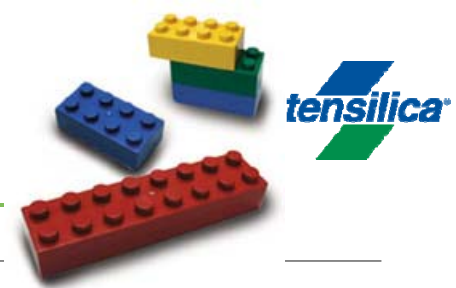
- 16 simultaneous 18x18b MAC per cycle
- 8 way SIMD + 3 way VLIW
- Dual load/store unit (128b wide)
- Scalar CPU pipeline with general 32b RISC instructions
- Rich vector operations with complex arithmetic support
- Extended precision with guard bits
- Efficient matrix operations
- 6 addressing modes

Performance:

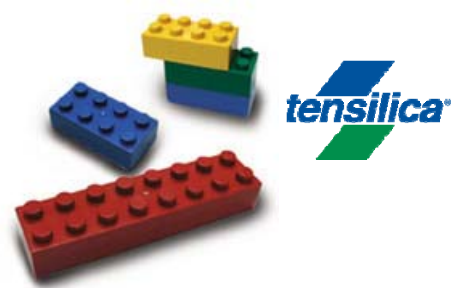
- 16 multiply-adds per cycle
- Three 8-way ops per cycle
- 4 complex FIR taps / cycle
- 1 Radix-4 FFT butterfly / cycle
- 17GB/s memory bandwidth(@ 550MHz)
- 40-bit accumulation on all MAC operations without performance penalty



Core 1: Kernel Performance of BBE16



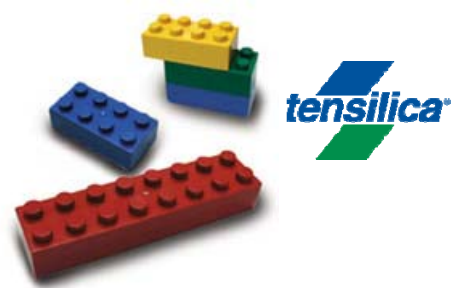
Core 1: DSP Kernels Library for BBE16



- 73 Kernels for generic DSP functions
 - FIR kernels: Real, Complex, Symmetrical), n-tap, 8-tap and 16-tap [11]
 - IIR kernels: Block, Real, Complex, Double Precision [6]
 - FFT kernels: Real and Complex, Forward and Inverse, Precision 16x16, length from 16 to 32K in steps of 2^n [7]
 - Correlation [2]
 - Linear algebra: Large variety of Matrix operations [25]
 - Transcendental (exp, log, sine, cosine, tangent) [18]
 - Communications kernels [4]
- Defined API Interface
 - Common across ConnX product line

Core 2: ConnX SSP16

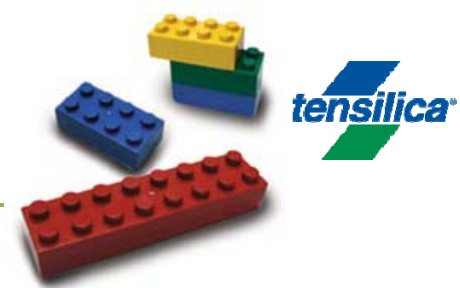
“Soft Stream Processor”



Purpose:

- Processing streams of soft bits (4 to 8-bit representation) highly intensive in LTE applications
- More power-efficient than if executed on big DSP like BBE16
- Architecture
 - **16-way SIMD Architecture with SIMD+VLIW**
 - 16-way SIMD computation for arithmetic operations
 - 2-way VLIW allows parallel instruction execution on SIMD and scalar code
 - **Optimized for LTE Soft Bit Processing**
 - Extended Instruction Set for Soft Bit Processing
 - 8-bit and 10-bit optimized
 - Automatic vectorization for ANSI C signed/unsigned char datatypes
- **Small size:** 150-200K gates (configuration dependant)

Core 2: ConnX SSP16 Processor



Data-types:

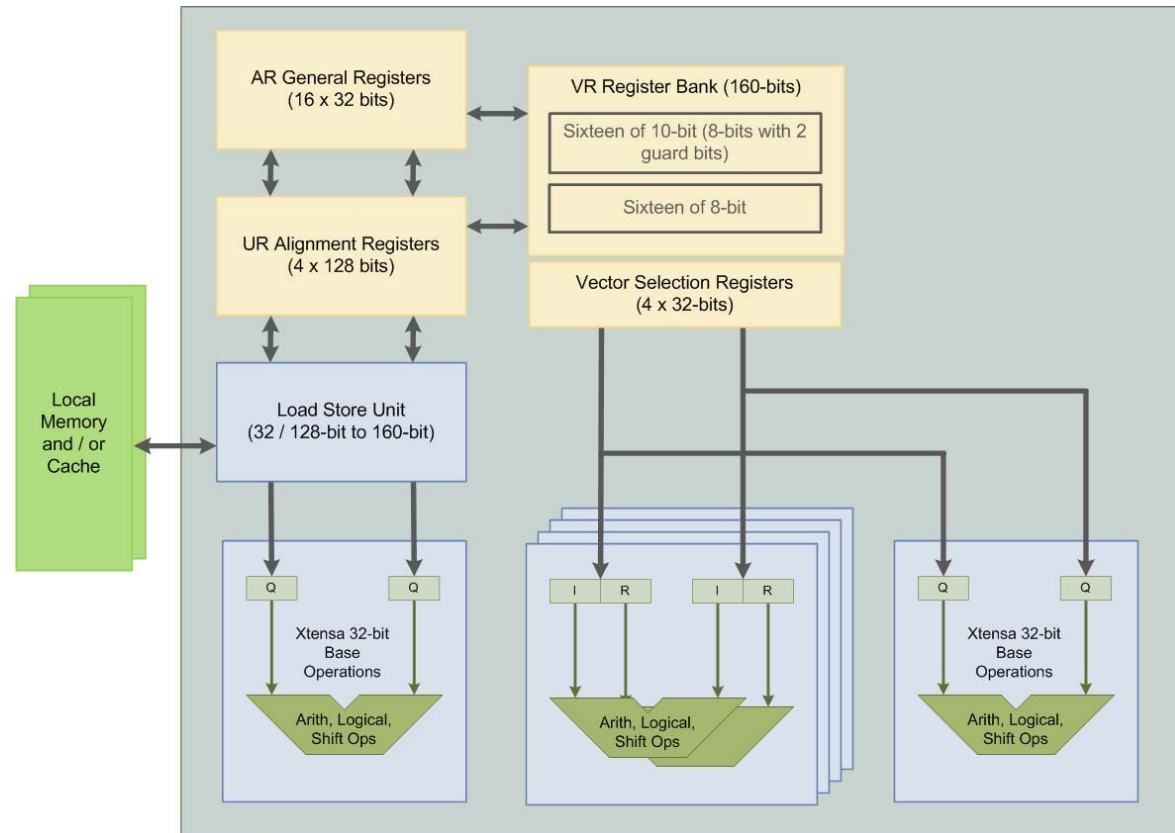
- 10-bit and 8-bit Vectors
- 8-bit, 16-bit and 32-bit Scalars

Performance:

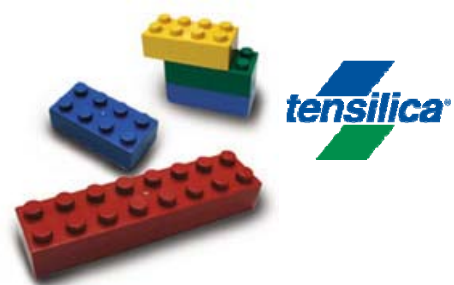
- 16 arithmetic operations per cycle
- 2 issue VLIW architecture
- 600 MHz in 45nm
- ~10GB/s memory bandwidth

Software:

- Vectorizing compiler
- Multi-core debugger
- Fast, cycle-accurate simulator
- Energy models
- Function libraries and 3rd party cellular stacks



ConnX SSP16 ISA Overview



Standard C Operations (8-bit, 16-bit, 32-bit)

- Add, Sub, Neg, ABS, CMP, CMOV, Shift, Reduced ADD and Logical Operations

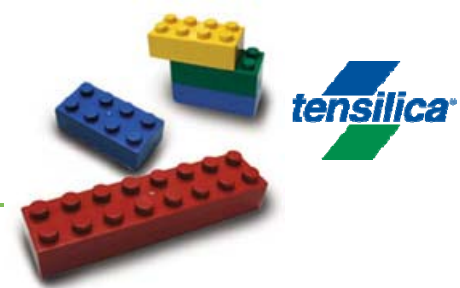
Data Types

- 8-bit, 10-bit Vectors
- 8-bit, 16-bit and 32-bit Scalar types

Loads and Stores Addressing Modes

- Index, Updating, Aligned, Non-aligned and Saturation Modes
- Load 16-bit / 32-bit scalars and vectors
- Store 16-bit / 32-bit scalar, vectors, transposed
- Load/store unaligned and masked

Core 2: ConnX SSP16 : Instruction Set Organization



16-bit Instructions
Base ISA

24-bit Instructions
Base ISA or ConnX SSP16

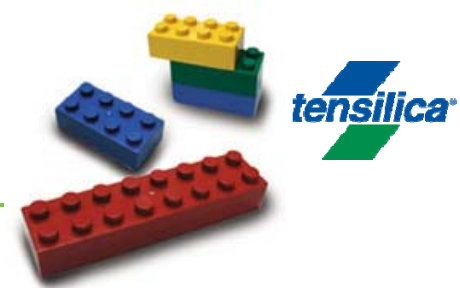
Slot 0
Base ISA
(core or vector Load/Stores)

Slot 1
ConnX SSP16 or Base ISA (Scalar core,
Vector SIMD operations)

VLIW
Instructions
(64-bits)

- **Flexible allocation of instructions available to compiler**
 - Optimum use of 2 VLIW slots (ConnX SSP16 or base ISA instructions)
 - Improved performance and no code bloat (reduced NOPs)
 - Reduce code size when algorithm is less performance intensive

Core 2: SSP16 ISA Summary



Slot0: SIMD loads, stores and general-purpose Xtensa operations

Loads: LVH8.I, LALIGN.I, LSS8.I, LSS8.IU, LSS8.X, LSS8.XU, LSU8.I, LSU8.IU, LSU8.X, LSU8.XU, LVS8.I, LVS8.IU, LVS8.X, LVS8.XU, LVU8.I, LVU8.IU, LVU8.X, LVU8.XU

Stores: SALIGN.I, SSS8.I, SSS8.IU, SSS8.X, SSS8.XU, SSU8.I, SSU8.IU, SSU8.X, SSU8.XU, SVH8.I, SVS8.I, SVS8.IU, SVS8.X, SVS8.XU, SVU8.I, SVU8.IU, SVU8.X, SVU8.XU, MOV160

Plus Xtensa base operations

Slot1: SIMD ALU operations

Arithmetic: ADD10, NEG10, SUB10, AND160, EQ10, LE10, LT10, NE10, ABS10, MAX10, MIN10, RADD10

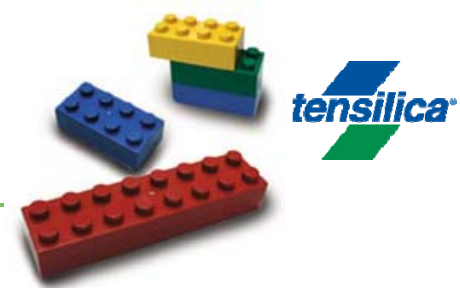
Logical: XOR160, MOV160, OR160, NAND160

Data arrangement: MOVF10, MOVT10, ESHFT10, SHFT10_8, SHFT10_4, SHFT10_2, SEL10

Shifts: SLLI10, SRAI10, SRAV10, SLLV10, SRA10, SLL10

Plus Xtensa base operations

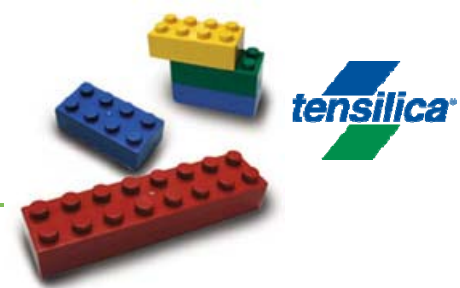
Core 3: ConnX BSP3 - Overview



“Bit Stream Processing at minimal size and power consumption”

- **Advanced Bit Manipulation and Control Instruction Set**
- **3-issue FLIX machine**
 - Parallel execution of instructions across 3 VLIW (FLIX) slots
 - Compiler allocates instructions across available slots for maximum performance
- **Dual Load/Store Architecture (Dual 32-bit data)**
 - High data bandwidth
 - High throughput on complex non-vectorizable code

Core 3: ConnX BSP3 Processor



Data-types:

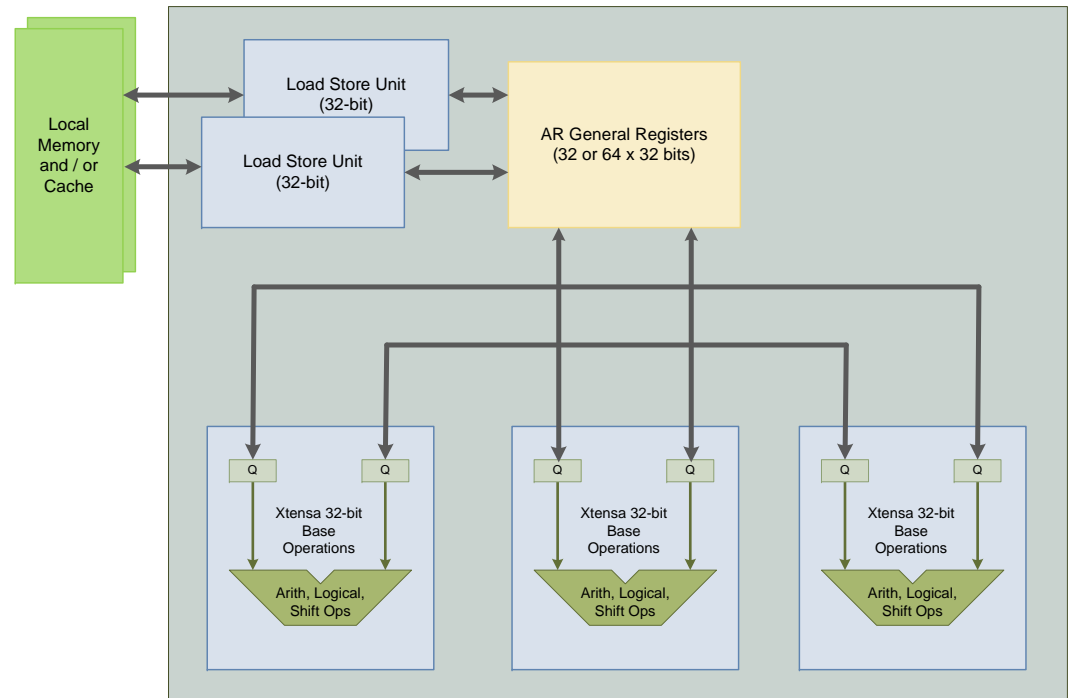
- 8-bit, 16-bit and 32-bit Scalars

Performance:

- Dual Load/Store
- 3 issue VLIW architecture
- 600MHz in 45nm
- >4GB/s memory bandwidth

Software:

- C/C++ compiler
- Multi-core debugger
- Fast, cycle-accurate simulator
- Energy models
- Function libraries and 3rd party cellular stacks





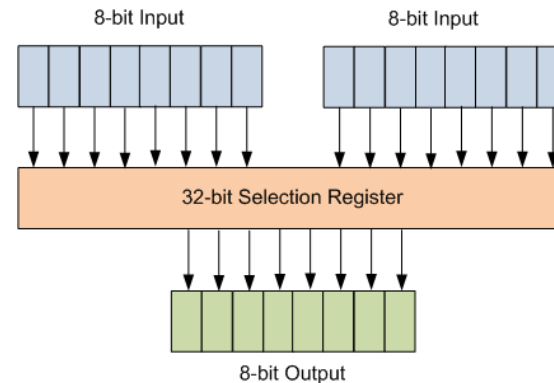
Core 3: ConnX BSP3 Bit Processing

Bit Insertion

- Instruction (INJUI) to insert bits into 8, 16 or 32-bit words

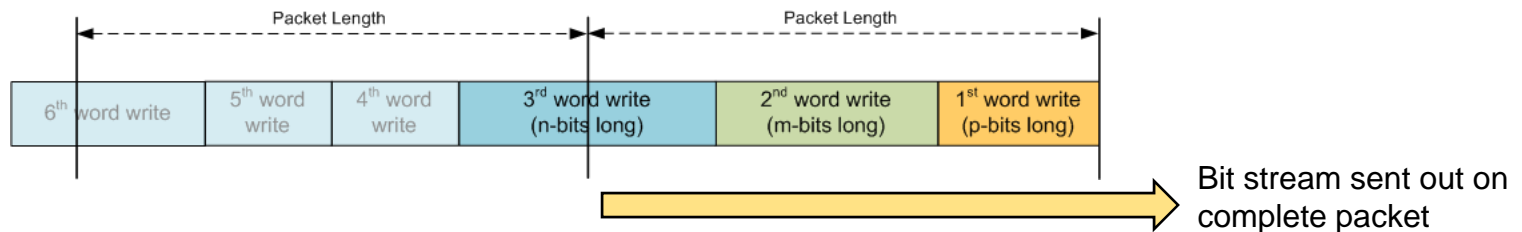
Bit Selector

- Two 8-bit inputs
- 32-bit selection register

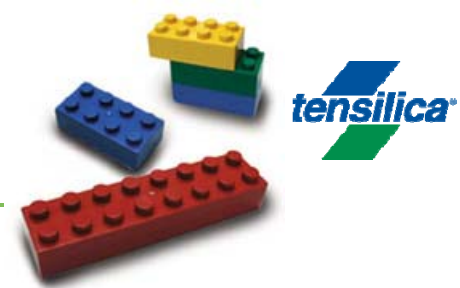


Bit Stream Writer

- Allows loading of different length words
- Once packet length is reached, sent down bit stream
- Fully synchronized operation of loading and sending
- Overflow bits allocated to next packet

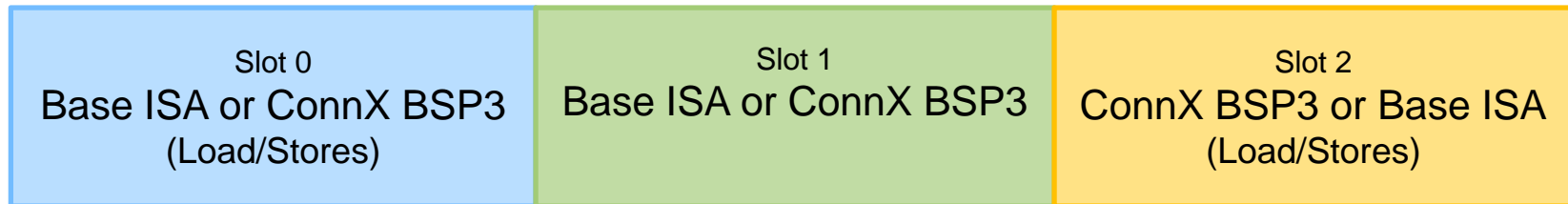


Core 3: ConnX BSP3 : Instruction Set Organization



16-bit Instructions
Base ISA

24-bit Instructions
Base ISA or ConnX BSP3

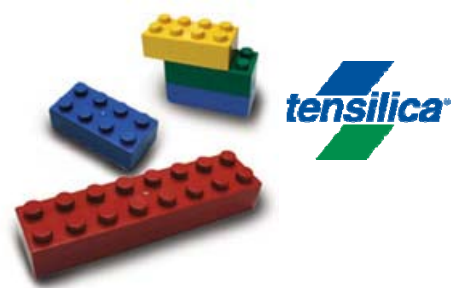


VLIW
Instructions
(64-bits)

- **Flexible allocation of instructions available to compiler**
 - Optimum use of 3 VLIW slots (ConnX BSP3 or base ISA instructions)
 - Improved performance and VLIW no code bloat (reduced NOPs)
 - Reduce code size when algorithm is less performance intensive
 - Modeless switching between instruction formats

Core 4: ConnX Turbo16

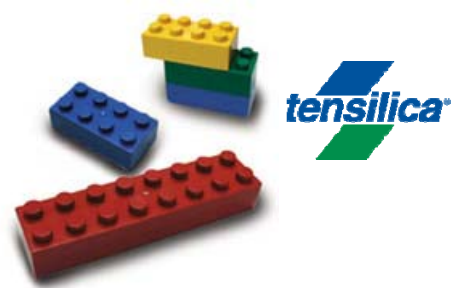
“LTE Turbo Decoder”



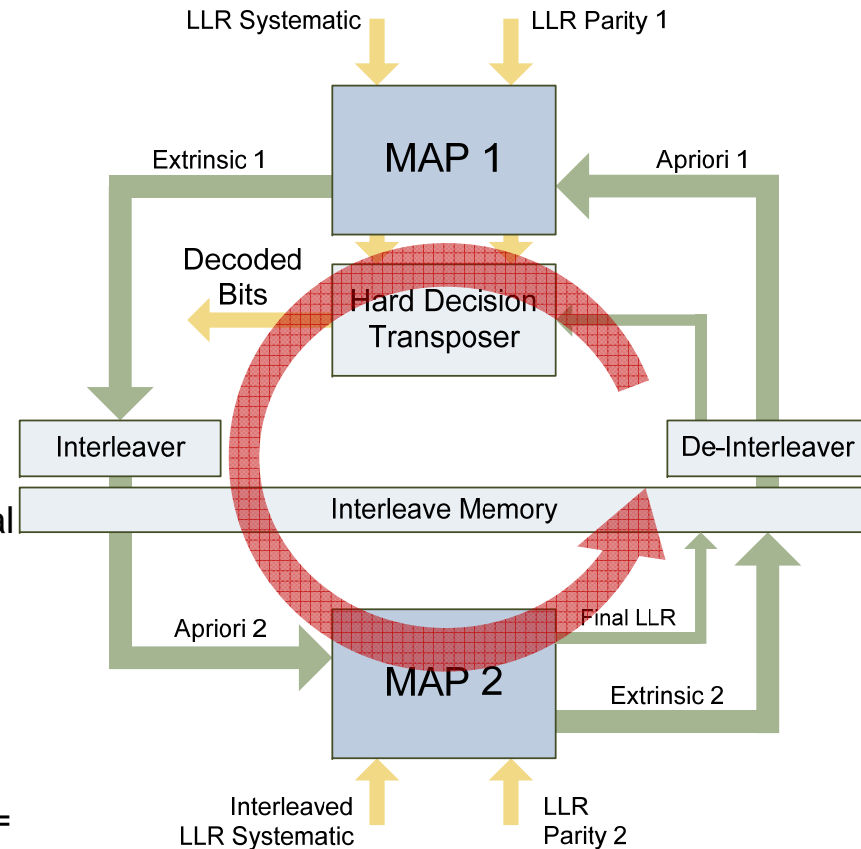
- **Customized Processor based solution for LTE Turbo decoding at 150 Mbps**
 - Matches throughput of hardwired RTL solution with similar area, power
- **Why choose ConnX Turbo16 engine ?**
 - Flexibility provided by programmability especially for on-the-fly adaptation to channel conditions:
 - SW tuning to provide smallest error rate for current channel conditions
 - SW tuning to minimize power while maintaining block error rate below acceptable threshold
 - Future extensibility to LTE + HSPA Turbo

Core 4: ConnX Turbo16

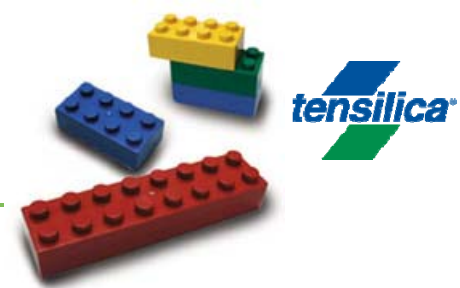
Turbo Decoding Algorithm Structure



- MAX-Log-MAP based decoding
 - Forward ALPHA pass and backward BETA pass
 - Uses: $\exp(\log(a) + \log(b)) \sim \max(a,b)$
- 8-parallel windows based decoding
 - Two bits from each window processed per update
 - MAP-1 decoder operates on LLR Systematic & LLR Parity1
 - MAP-2 decoder operates on interleaved LLR systematic & LLR Parity2
 - Interleaving and deinterleaving operations are integrated within the load/ store operations to the local memories (Odd and Even D-RAM)
- Confidence estimator for software-based early-termination → lower power
 - Under good channel conditions one need not run MAP1 and MAP2 decoders for ITER_MAX (typically = 8)
 - Often, error correction is achieved in < 5 iterations



Core 4: ConnX Turbo16 Architecture

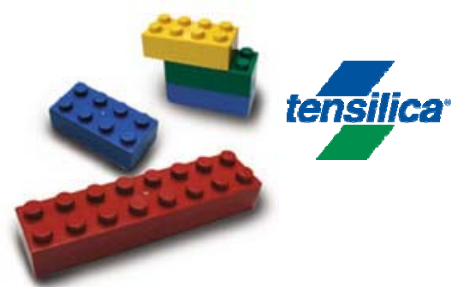


- **4-issue VLIW Machine**
 - Parallel operation of dedicated, optimized tasks
 - Up to 4 memory accesses/ cycle occur (in ALPHA, BETA operations)
 - Two 128-bit accesses to Local D-Ram (via Dual Load/Stores)
 - Dual 64-bit interleaved Apriori memory
 - Or, Single 640-bit access to State Memory
- **Massively parallel SIMD engine with multiple memory loads/stores**

Xtensa VLIW (FLIX) for parallelism	ALU Slot 0 Load/Store 0	ALU Slot 1 Load/Store 1	ALU Slot 2 Read/Write State	ALU Slot 3 Alpha Beta
Instructions per slot	LD_WD128.I LD_WD128.IU	LD_WD128.I LD_WD128.IU	RD_STATEU RD_MEM, RD_INTER	ALPHA, BETA BETA_INTER DECISION WR_INTER
Operation	128-bit Load of Systematic Data from local D-RAM	128-bit Load of Parity Data from local D-RAM	Load of state from local TIE Memory Or Read apriori LLR from dual 64-bit apriori memory	ALPHA and BETA executes 16-way SIMD Writes to dual 64-bit apriori memory

Core 4: ConnX Turbo16

Code Implementation



Implementation of Alpha and Beta passes in C-code (MAP2 example)

```
for (k=kq/2-1;k>=0;k--) { //ALPHA Loop
    ALPHA(RD_INTER(),*(--SystemP),*(--ParityP));
}

for (k=kq/2-1;k>=0;k--) { //BETA Loop
    BETA(RD_STATEU(),*(--SystemP),*(--ParityP));
}
```

Compiler maps C-code routines to most efficient execution, using FLIX for parallel operation and code scheduling for enabling alpha/beta instructions every cycle

Compiler-generated inner loop: Automatic register allocation, software pipelining, loop unrolling (4x) due to resource requirements and code scheduling

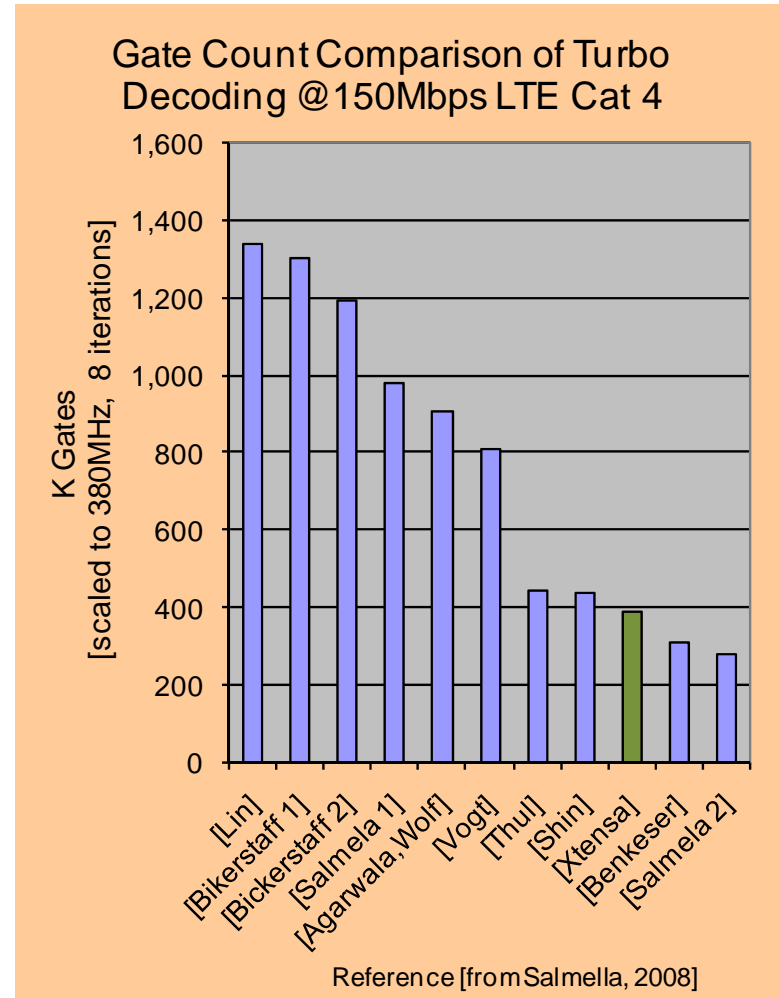
```
2fd: loopgtza3, 320 <main+0x320> //ALPHA Loop
300: { ld_wd128.iu w0, a2, 16; ld_wd128.iu w1, a9, 16; rd_inter ap0; alpha ap0, w0, w1 }
308: { ld_wd128.iu w2, a2, 16; ld_wd128.iu w3, a9, 16; rd_inter ap1; alpha ap1, w2, w3 }
310: { ld_wd128.iu w4, a2, 16; ld_wd128.iu w5, a9, 16; rd_inter ap2; alpha ap2, w4, w5 }
318: { ld_wd128.iu w6, a2, 16; ld_wd128.iu w7, a9, 16; rd_inter ap3; alpha ap3, w6, w7 }

3fd: loopgtza3, 320 <main+0x320> //BETA Loop
400: { ld_wd128.iu w0, a8, -16; ld_wd128.iu w1, a9, -16; rd_state st0; beta_inter st1, w2, w3 }
408: { ld_wd128.iu w2, a8, -16; ld_wd128.iu w3, a9, -16; rd_state st1; beta_inter st2, w4, w5 }
410: { ld_wd128.iu w4, a8, -16; ld_wd128.iu w5, a9, -16; rd_state st2; beta_inter st3, w6, w7 }
418: { ld_wd128.iu w6, a8, -16; ld_wd128.iu w7, a9, -16; rd_state st3; beta_inter st0, w0, w1 }
```



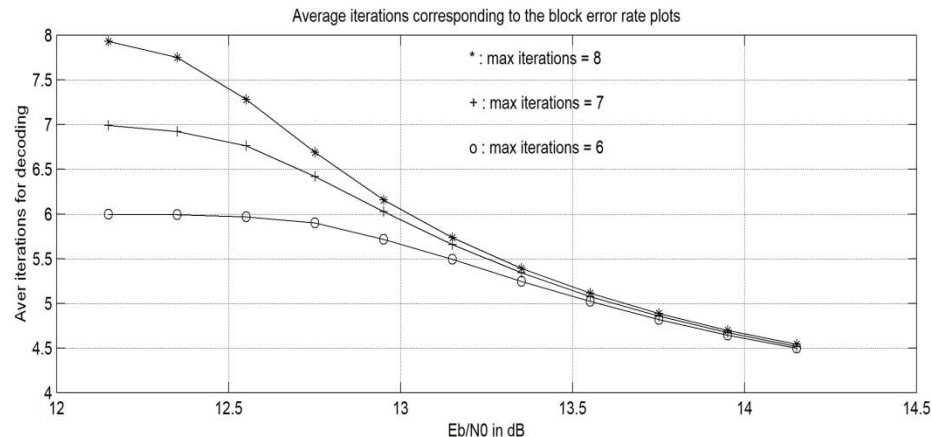
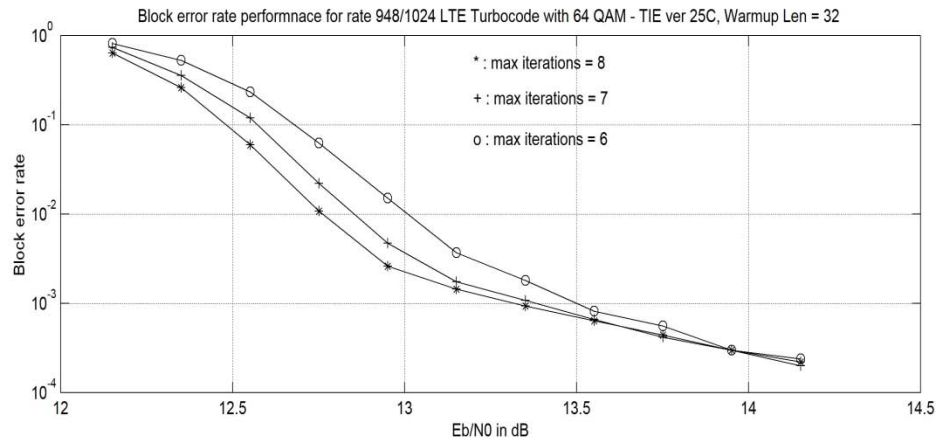
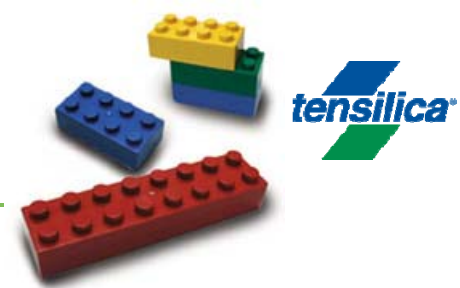
Core 4: Turbo16 as Efficient as RTL

- Enables RTL-like performance and size with processor programmability
- Good flexibility
 - Arbitrary memory referencing
 - Software-based control structures around optimized inner-loop instructions
 - FLIX technology enables compiler scheduling of parallel operations
 - Instruction set optimized for software-based early terminations
- Processor generator creates compiler and processor together
 - Advanced features all accessible from C-code
 - C-programming Ease of use model
- Small Size
 - 375K gates + 88KB memory – comparable to hardwired Turbo decoder



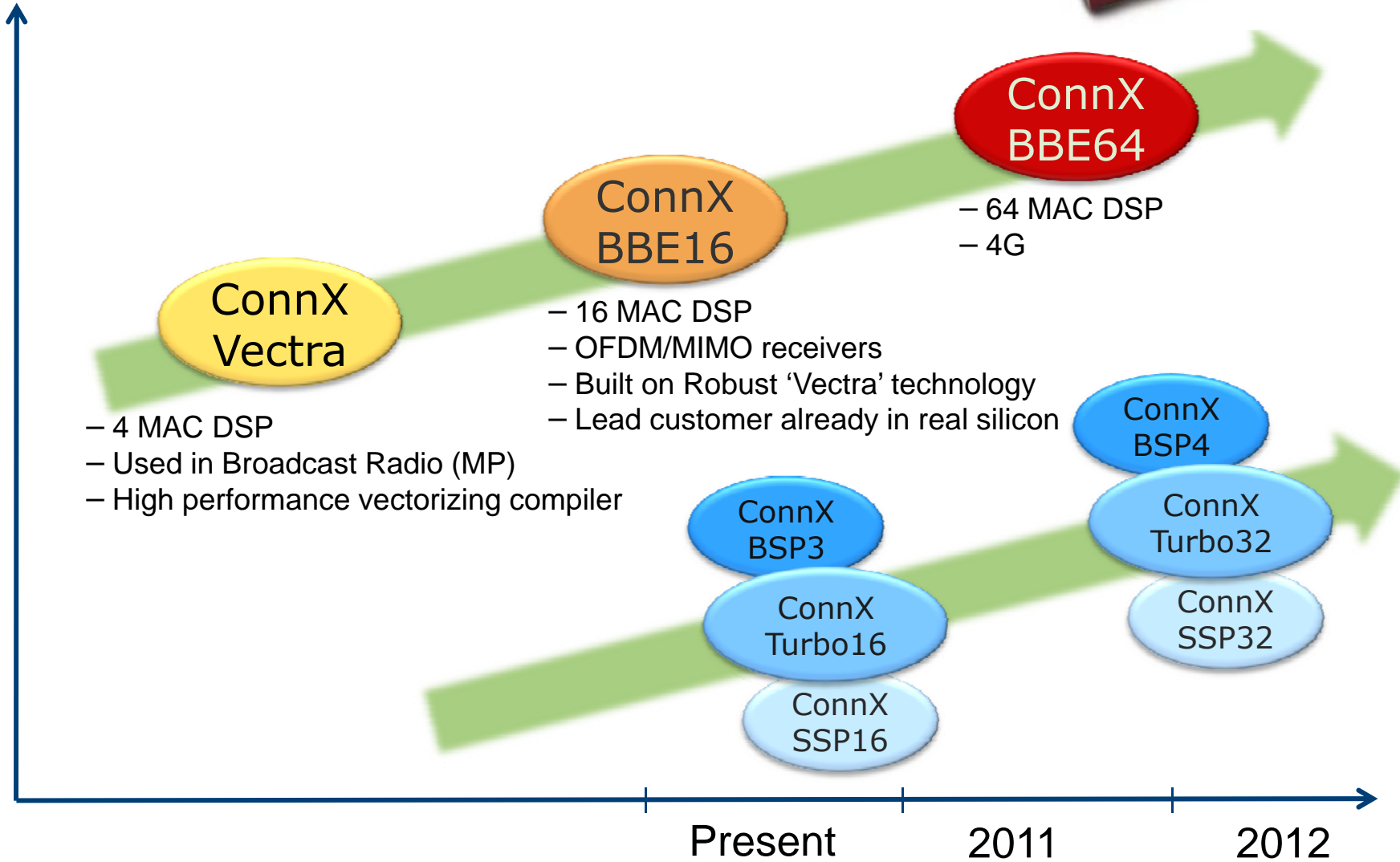
Core 4: ConnX Turbo16

Turbo Decoding Performance



- Max clock speed for ConnX Turbo16: 350 MHz
- MCPS requirement for decoding 150 Mbps data stream
 - 8 iterations: 382
 - 7 iterations: 341
 - 6 iterations: 300
- Block-Error Rate (BLER) and average iterations versus Eb/N0 (shown on left)
- Average iterations significantly smaller than max iterations due to early termination using confidence estimators
- For BLER < 1% (typical operation point), even with 8 max iterations, average iterations still < 7
- Max iterations = 8, sustainable at 150 Mbps for BLER < 1% using early termination

Roadmap for ConnX Building Block Family



LTE Reference Architecture “Atlas”



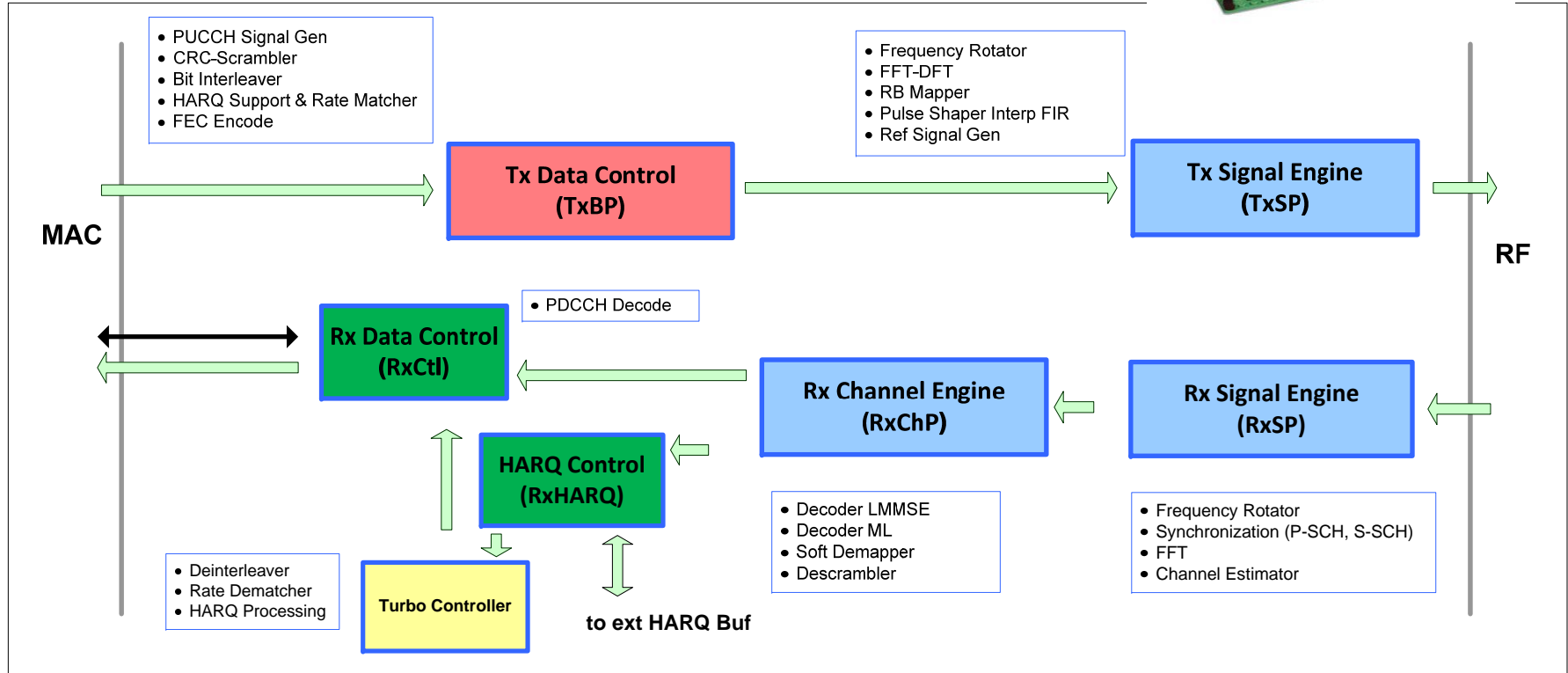
- Demonstration of best practices for low-cost/low-power multi-core-based digital PHY sub-system
 - 100% processor-based – no RTL blocks
 - Highlights agile configuration of ConnX processors
- Proves efficiency of digital-PHY solution with processors
 - Reference architecture implements all blocks, memories and interconnect
 - Port of complete PHY software solution: mimoOn mi!MobilePHY™ 3GPP Fully Compliant LTE PHY
- 7 core solution for Cat 4 UE (150Mbps DL/50Mbps UL 20MHz):
 - 3 x BBE16
 - 2 x SSP16
 - 1 x BSP3
 - 1 x Turbo16
- Adaptable to mix-and-match with existing hardwired functions
- Scalable across performance: Cat 2 – Cat 5 and eNodeB base-station

Atlas UE Platform Functionality and Scope



- Support from MAC transport block to front-end filters
- Slave to MAC Controller
- Communication through an user defined API
- MAC provides
 - Scheduling information
 - Transmit & Receive Parameters: Modulation and Coding Information
- PHY provides
 - uE: Reception Quality Information
 - uE Channel Condition Information
- PHY System maintains HARQ Buffers
 - Tighter latency control
 - HARQ ACK/NACK mapping

ATLAS LTE-UE Platform: Functional Block Diagram

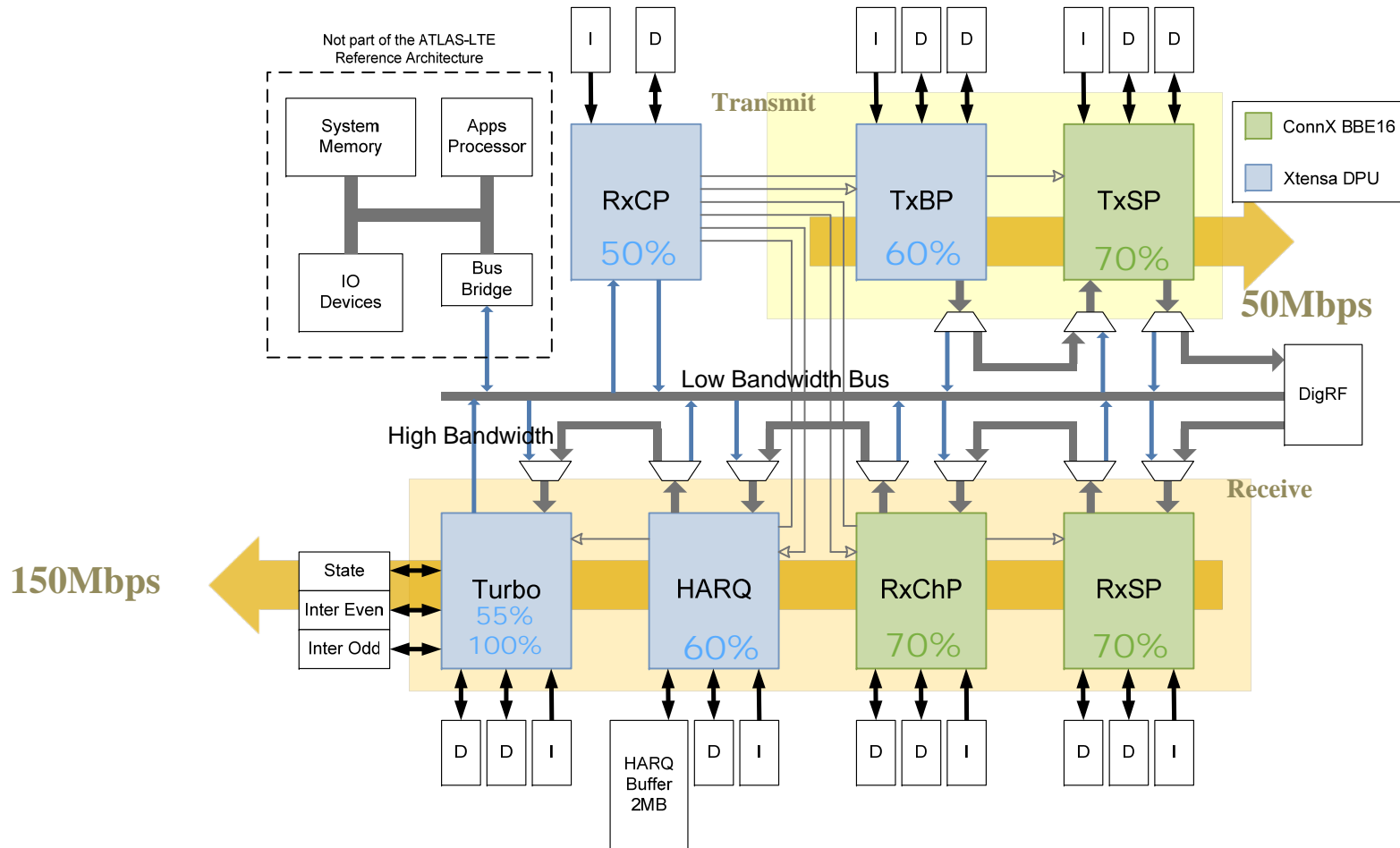


- Signal Domain (Time -- Frequency):
- Matrix Domain (Frequency -- Soft Bits)
- Soft-Bit Domain (Soft Bits-Bits)
- Control Domain
- Bit Domain (Soft Bits-Bits)

I-Q Ops: -FFT, DFT, Synch, Channel Estimation
Matrix Ops (I-Q Data): QR, LMMSE, ML
Soft Descramble-De-interleave, HARQ, Turbo Dec
Config and Control, Communicate with MAC Host
FEC Encode, Bit Scramble-Interleave, Rate Match

ATLAS-LTE UE

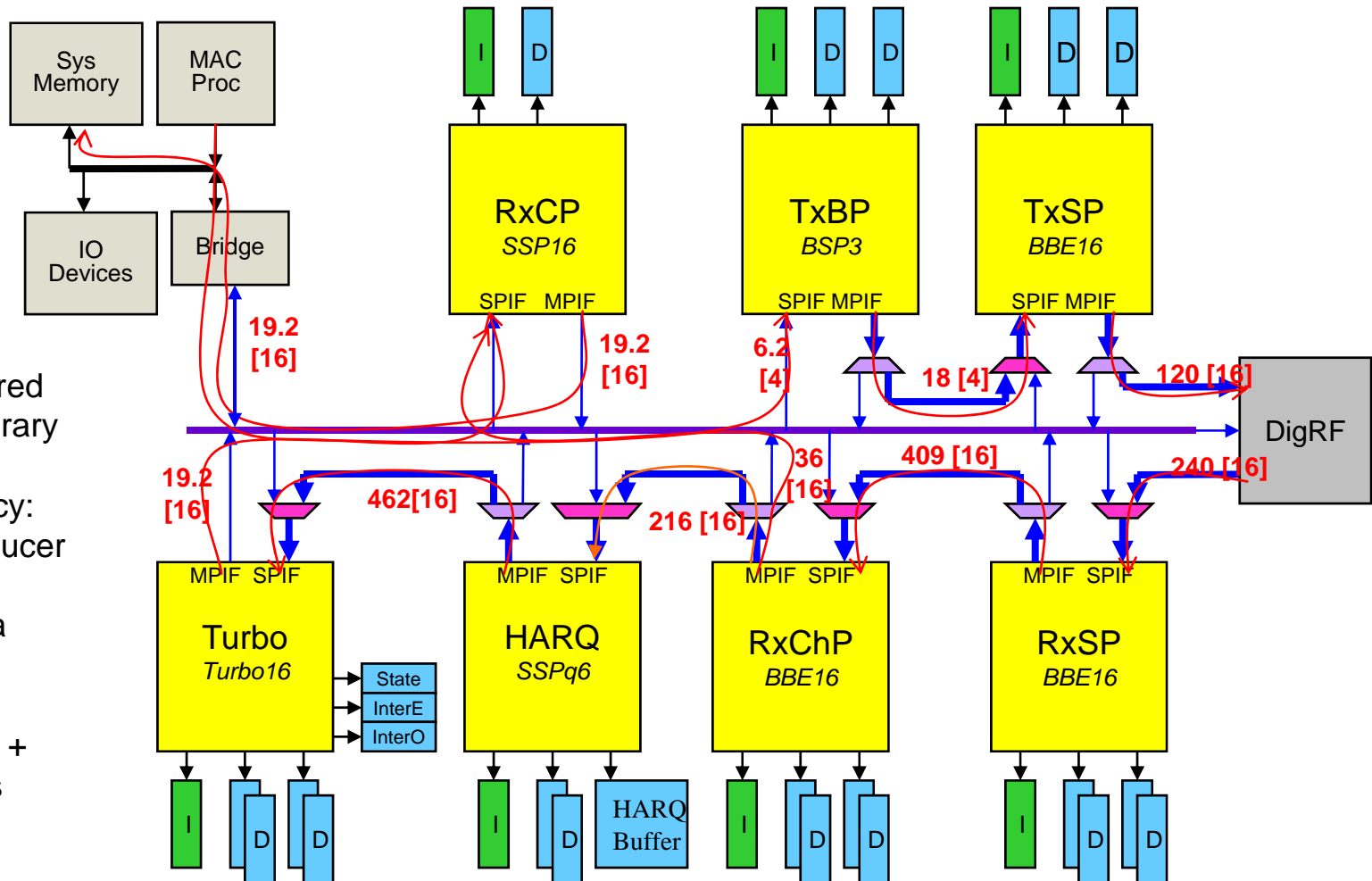
CAT4 @ 350MHz



Data Communications in Atlas



Key: Data transfer rate in Mbytes per second [transfer size in bytes]



- Consistent shared memory SW library
- Minimum latency: Each data producer writes to local memory of data consumer
- Global 32b bus + local 128b links

System Configuration



Processor	Type	Inst RAM/Cache Size	Data RAM Size	Configurations + TIE
RxSP	ConnX BBE16	32K + 4K cache	80 K	
RxChP	ConnX BBE16	32K + 4K cache	64 K	MIMO + Soft Demod
RxHARQ	ConnX SSP16	32K + 4K cache	2MB	Byte Transpose
Turbo	ConnX Turbo16	16K + 4K cache	64 K + 48 K	Turbo
RxCtl	ConnX SSP16	64K + 4K cache	24 K	Viterbi, CRC
TxBP	ConnX BSP3	64K + 4K cache	24 K	Bit Transpose, CRC
TxSP	ConnX BBE16	32K + 4K cache	48 K	
Total		272K + 28K cache	2352 K	



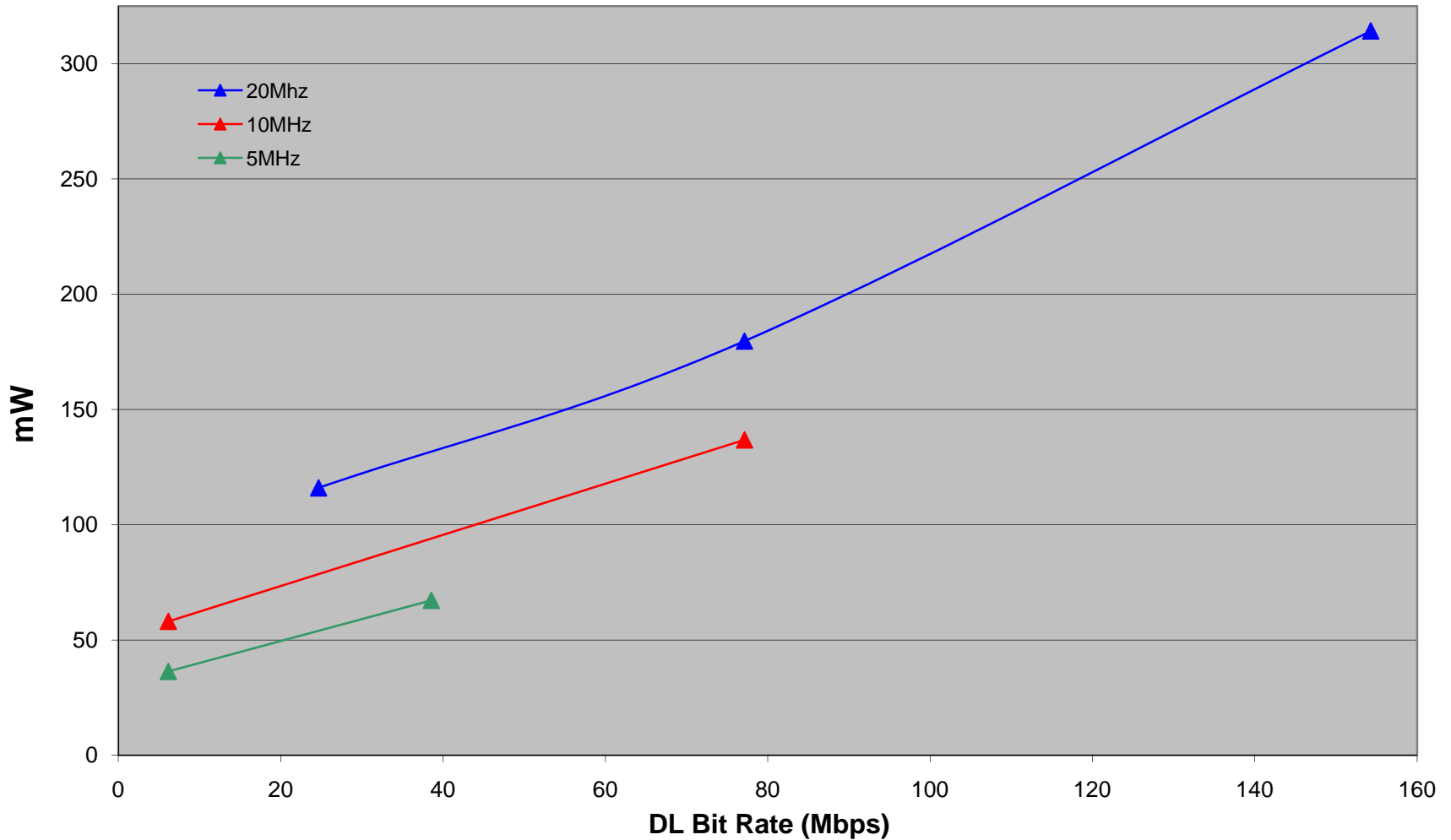
Area Analysis (40LP)

Proc	Logic Area	RAM Area	Total Area (mm ²)
RxSP	1.02	0.57	1.59
RxChP	1.13	0.50	1.62
RxHARQ	0.37	0.61	0.98
Turbo	0.79	0.71	1.50
RxCP	0.31	0.43	0.74
TxBP	0.21	0.43	0.67
TxSP	1.02	0.43	1.45
HARQ Buffer		8.00	8.00
Buses	0.13		0.13
Total	5.0	11.7	16.7

Atlas Power Dissipation



LTE Power Dissipation Estimate (Turbo early termination, 40LP)



Atlas Power Analysis Methodology



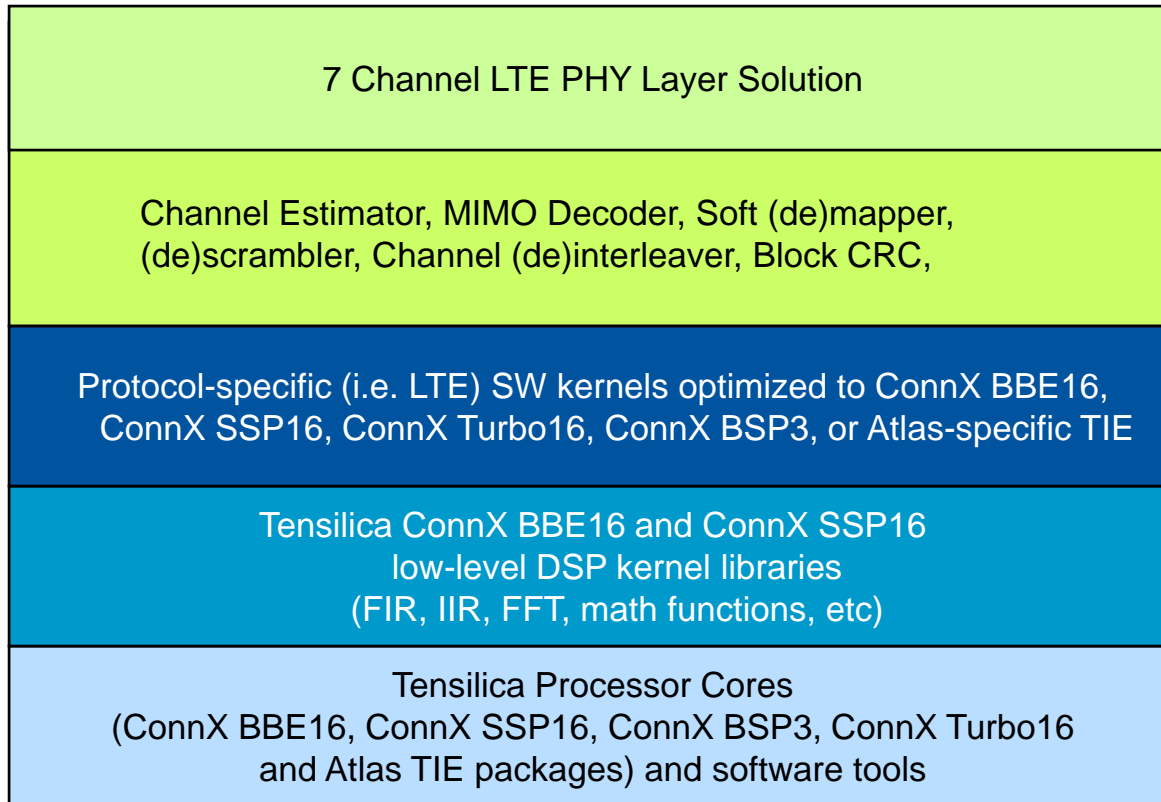
▀ System Level Assumptions

- Based on BW, allocated RBs, QAM Level
- RB Allocation contiguous in frequency and time
- Accounts for PDSCH, PDCCH, PUSCH
- Duty Cycled Operation for low data rates
 - “ON” Subframe: PDSCH, PDCCH, PUSCH
 - “OFF” Subframe: PDCCH (Header Processing only), PUSCH
 - Simulates realistic MAC scheduling for <10% resource allocation

▀ Power estimates include

- Core Logic
- Local Memories
- Active and Leakage power components
- HARQ Buffer
- Subsystem bus activity
- Does not include: L2/L3 stack processing, apps subsystem power

LTE Software Stack



Wrap-up



1. Conflict between flexibility and efficiency is resolvable
2. Data-plane processors combine three characteristics
 - **Adaptable:** interface, instruction set, memory system automatically fit need
 - **Programmable:** proven instruction sets, compilers, libraries, SW stacks
 - **Efficient:** Rivals hardware area and power dissipation on complex problems
3. LTE handset challenge shows need ... and possibility ...for better solutions
4. Even bigger challenges ahead” - SOC complexity in 2010 = 0.0

Relative SOC Complexity
M1 features/mm²

