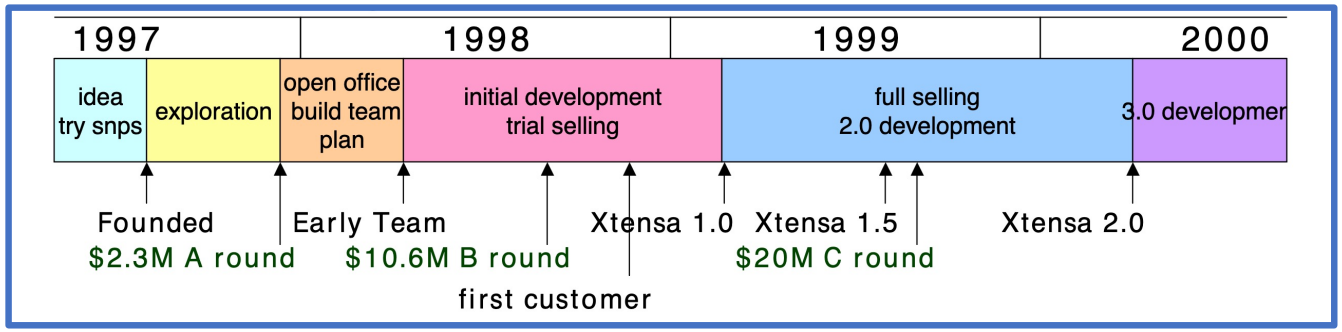
Automating Processor Design: Tensilica Technology and History

Dr. Chris Rowen

November 29, 2023
Stanford University

Founding of Tensilica

- Incorporated July 30, 1997
- Chris Rowen, later joined by
  - Bernie Rosenthal
  - Harvey Jones
  - Earl Killian
  - Gulbin Ezer
  - Ashish Dixit
  - Beatrice Fu
  - Dror Maydan
  - Albert Wang
  - and more



- A long build of customers, new technology and spreading use
- Acquired by Cadence, April 2013. Now a widely successful business unit inside Cadence

Mission of Tensilica

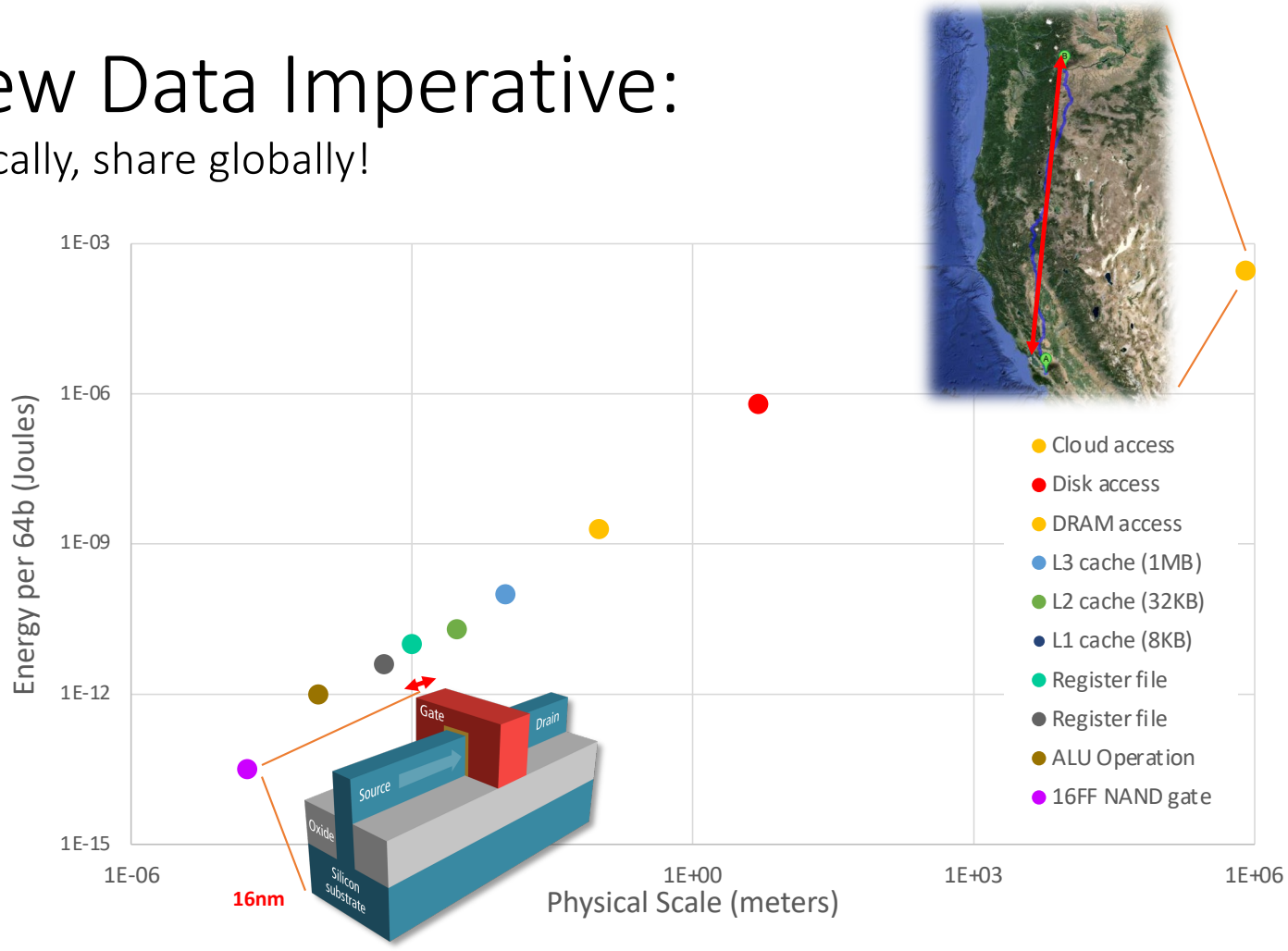
from an early corporate overview – 1998

To be the leading provider of
application-specific microprocessor solutions
by delivering
configurable, ASIC-based cores
and
matching software development tools

- Reality: Actually… that was it. With particularly strong success in audio, communications and video DSPs.

The New Data Imperative:

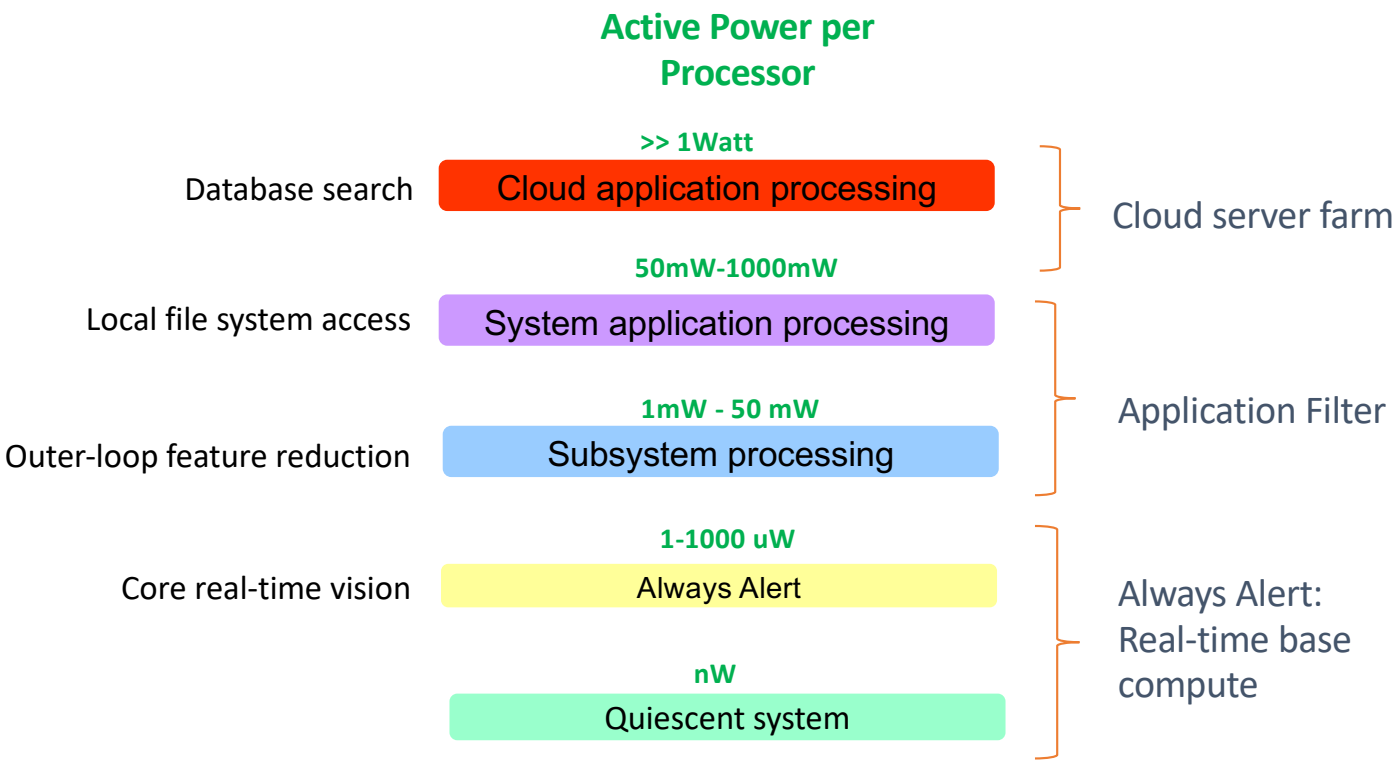
Compute locally, share globally!



Sources:

- Rick Zarr, TI, 2008- The True Cost of an Internet "Click" - estimate of transfer cost for 30KB page from server: http://energyzarr.typepad.com/energyzarrnationalcom/2008/08/the-true-cost-o.html
- J Kunkel et al, University of Hamburg 2010,Collecting Energy Consumption of Scientific Data
- Horowitz ISSCC 2014, 1300-2600 pJ per 64b access

Energy and layered cognition

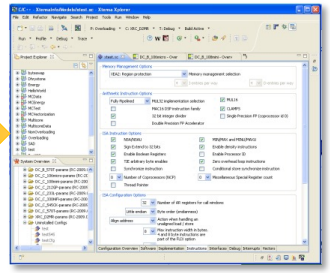


Tensilica automated hardware/software synthesis

Architects' needs:

- **Higher compute throughput**
- **More memory bandwidth**
- **Higher efficiency control structure update**
- **Continuous software upgrade of EVERYTHING**

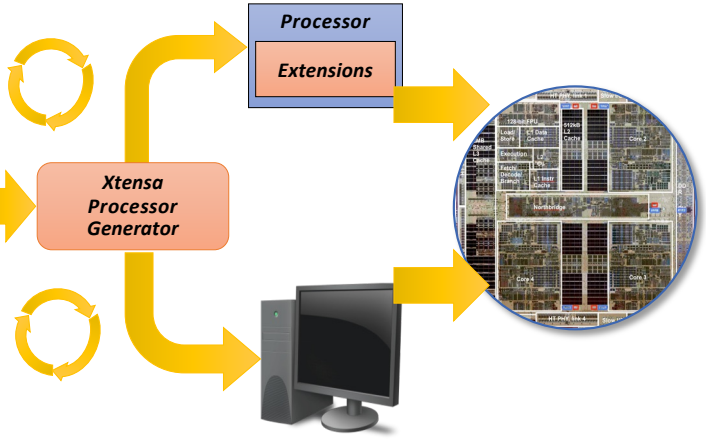
Processor configuration



1. **Select from menu**
2. **Explicit instruction description (TIE)**

Full hardware design

Source pre-verified RTL, EDA scripts, test suite



Customized software tools

C/C++ compiler Debuggers, Simulators, RTOSes

No manual intervention needed in hardware or software completeness

Xtensa: Two Part Architecture

Base + Extensions

Base Resources:

- Instruction fetch & decode –24b I formats
- 32b AR register file
- Control flow operations
- RISC operations on AR registers
- Exception and interrupt handling
- Data load/store unit

Common configuration options:

- 5 vs 7 vs long pipeline
- Endianness
- Zero-overhead loops
- Arithmetic units – multiplier, floating point
- Configurable load/store unit
- AR register windowing for call/return
- 16b format for common inst
- Configurable instruction width (FLIX)
- Various memory protection and TLB memory management options
- More compress instruction handling
- Different bus interfaces
- Many debug and trace options

TIE language (in order of addition)

- New operations on AR registers
- New state registers
- New register files – C type mapping, RTOS support
- New deeply pipelined operations on multiple register files. – C operator inference
- Ports and queue interfaces – huge I/O bandwidth with intrinsic synchronization
- New variable length instructions – automatic packing & scheduling
- Wide, complex memory operations – huge local bandwidth

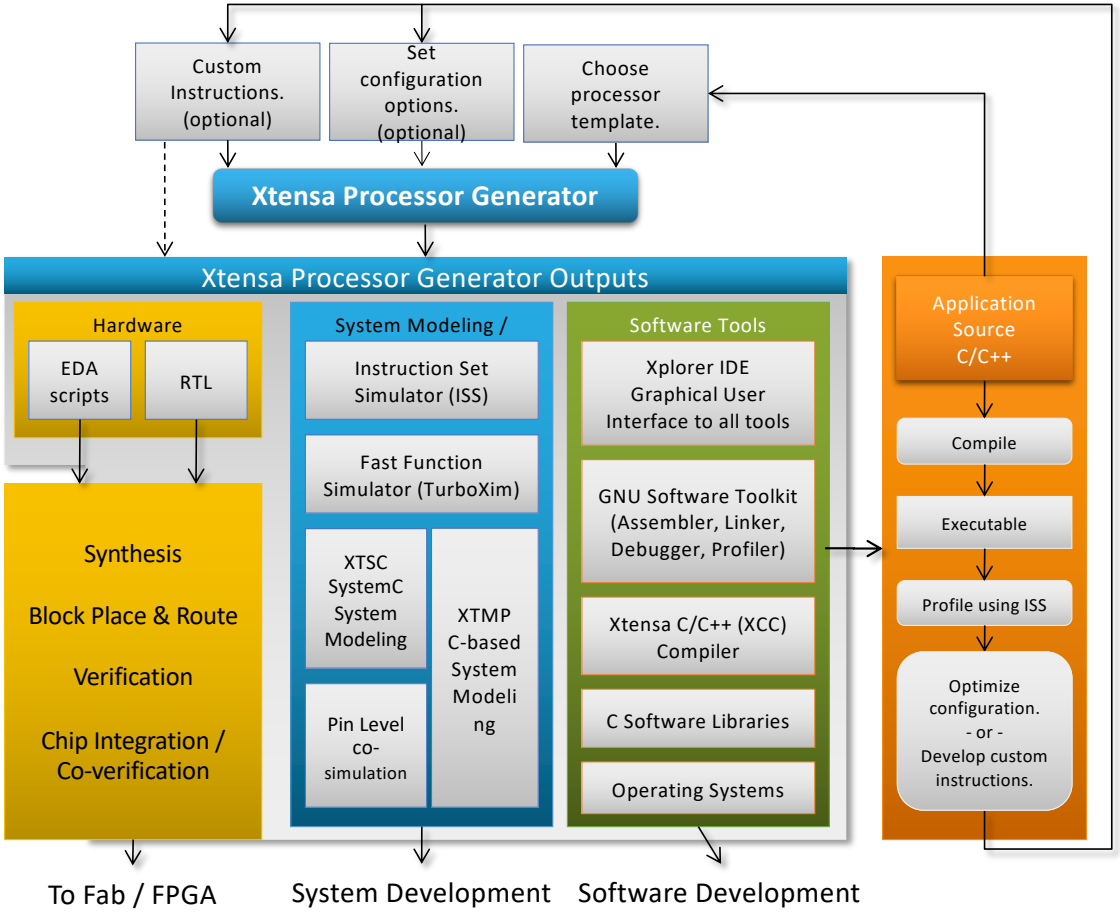
Fundamentally – a new kind of HW/SW design tool

Single source➔

RTL design
EDA tool support
fast cycle-accurate model
compiler mapping
OS support

The Xtensa Solution

Fully Automated Hardware and Software Tools Generation



Tensilica Instruction Extension

Comprehensive, robust ISA and interface extension

- Primitives:
  - **operation**: any number of input and output operands, any combinational Verilog as function
  - **regfile**: any width and depth, allocated by compiler
  - **state**: any width and depth, implicit operand to operations
  - **format** + **slot_opcodes**: specify VLIW bundling for compiler optimization
  - **proto**: association of C and extended types to operations and regfiles
  - **schedule**: inputs and outputs of operations at any stage (up to 32 stages)
  - **semantic**: optional hardware sharing of operation bodies
  - vectorization hints: automatic vectorization of C and extended types

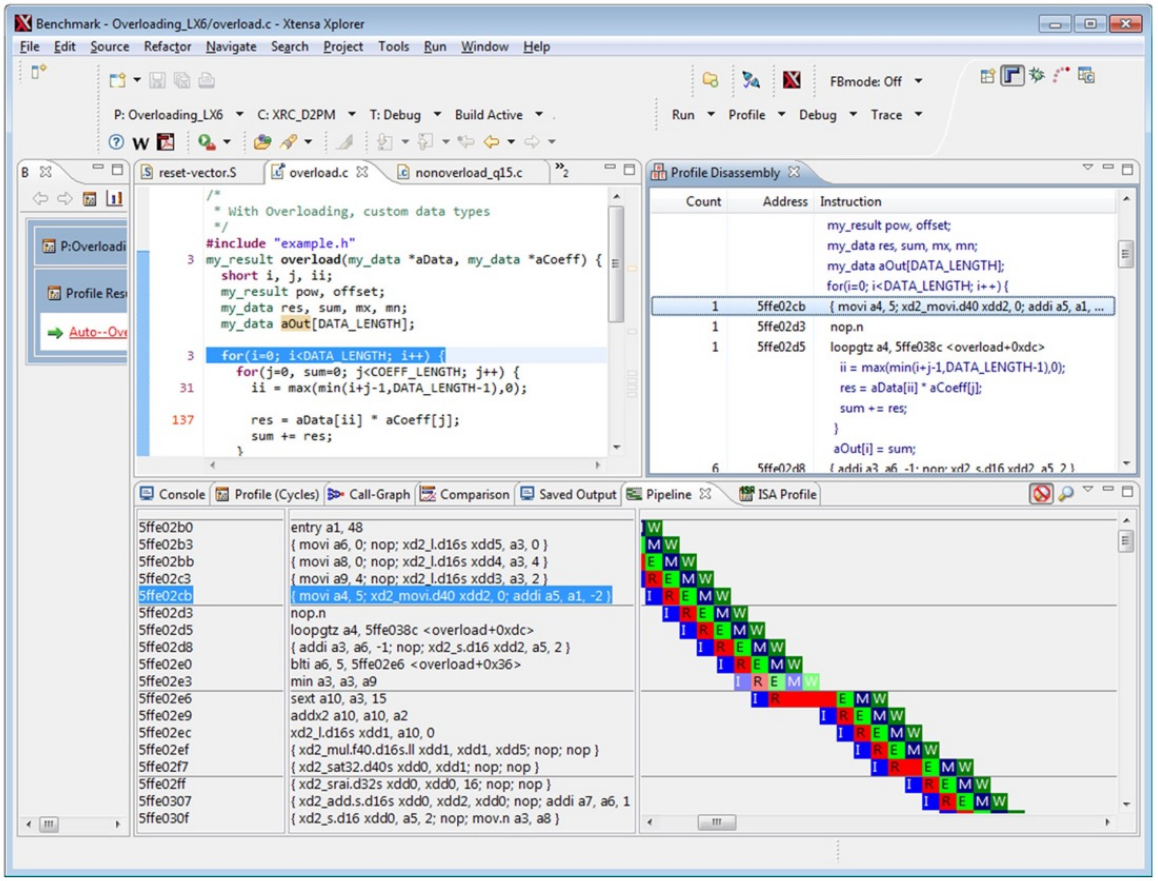
```
regfile vec 512 16 v
operation add32v {out vec z, in vec x, in vec y} {} {
; for ($i (0..15) {
    wire[31:0] z`$i` = x[`$i*32+31`:`$i*32`] + y[`$i*32+31`:`$i*32`]'
; }
    assign z = {` join(",",map("z$_".reverse 0..15))`}; }
format three_way 64 {ls_slot, ls_slot, op_slot}
slot_opcodes ld_slot {ld.vec,st.vec,l32i,s32i,l16si, s16i}
slot_opcodes op_slot {add32v,add, sub, addi, movi, mov}
```

C/C++ Compilers for auto-generated processors

Generating complete compiler chain for each configuration and ISA

- TIE Complier generated configuration-specific dll used by all phases of software generation system
- Every new operation available as intrinsic function
- Every new register file accessible through new datatypes
- Memory operations on new datatypes generate load/store or compound memory ops
- Automatic scheduling/packing of operations into variable-length VLIW packets
- Mapping between scalar and vector operations marked in TIE – enables automatic vectorization
- Graph-based matching to infer fused base and extended instructions
- Operator fusion + vectorization + SIMD packing
- Linker optimizations to handle user-defined instructions with relocation types and arbitrary-length fields
- Automaticallly-generated libdb contains all configuration info – used by xt-gdb and third-party debuggers

Cycle-accurate simulation and visualization

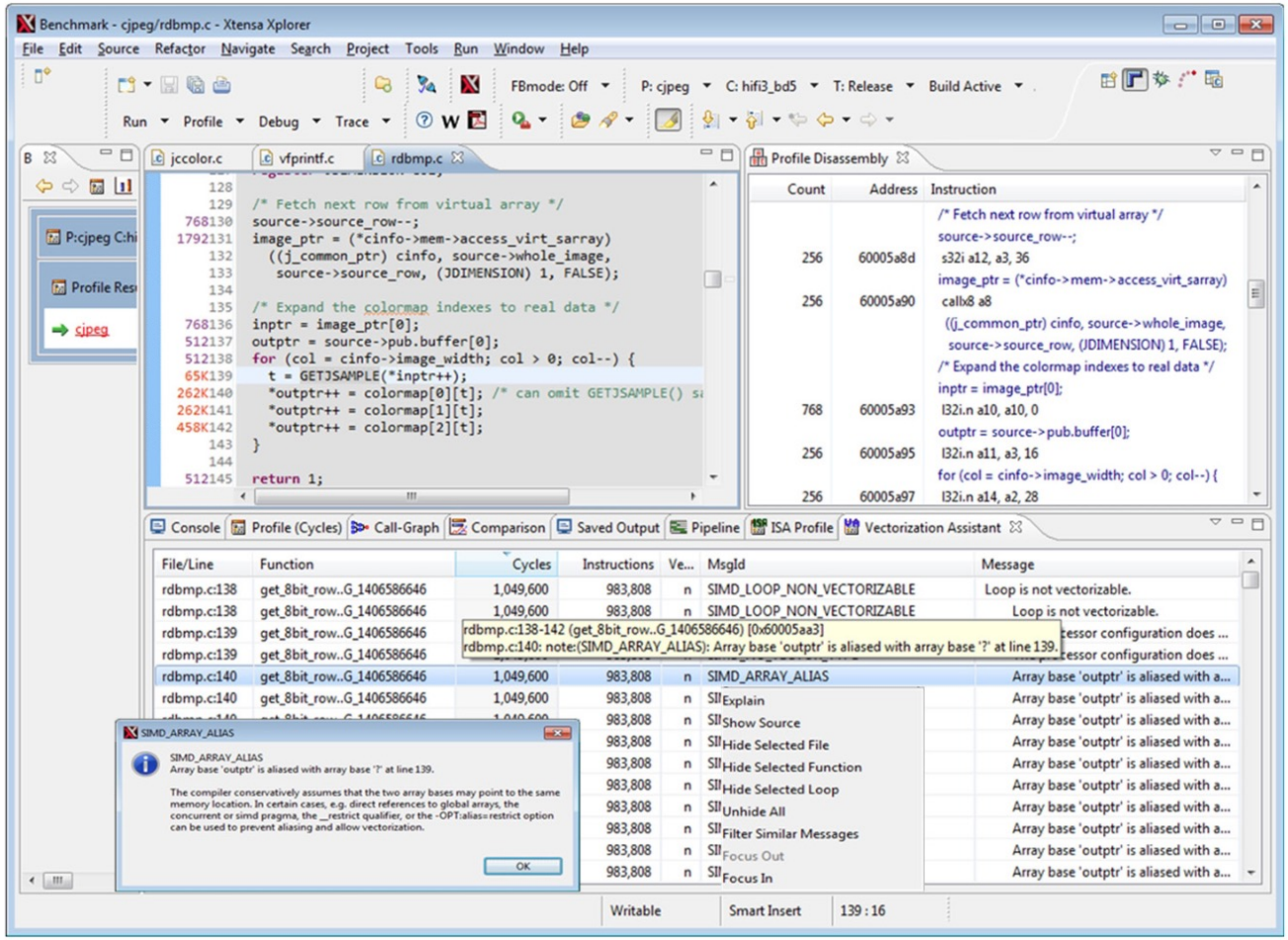


Faster

More accurate

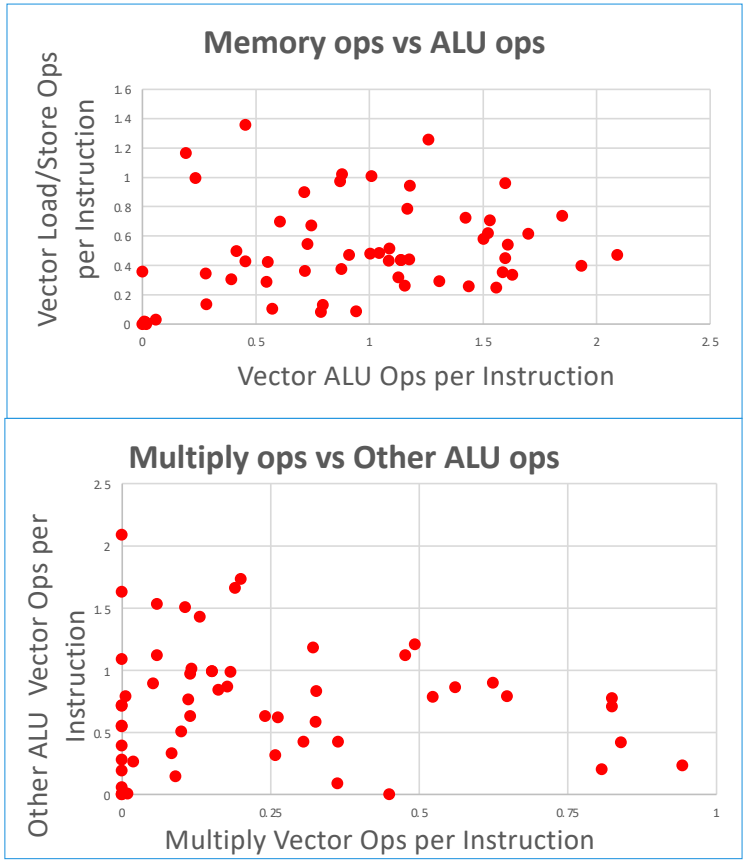
Simulation Speed	Modeling Tool	Benefits
20 to 50 MIPS[1]	Standalone ISS in TurboXim mode	Fast functional simulation for rapid application testing
1.5 to 40 MIPS[1,2]	XTMP or XTSC in TurboXim mode	Fast functional simulation for rapid application testing at the system level
800K to 1,600K cycles/second	Standalone ISS in cycle-accurate mode	Software verification and cycle accuracy
600K to 1,000K cycles/second[2]	XTMP in cycle-accurate mode	Multi-core subsystem modeling and cycle accuracy
350K to 600K cycles/second[2]	XTSC in cycle-accurate mode	Multi-core subsystem modeling with SystemC interfaces and cycle accuracy
1K to 4K cycles/ second	Verilog RTL simulation	Functional verification, pipeline-cycle accuracy and high visibility and accuracy
10 to 100 cycles/ second	Verilog gate-level simulation	Timing verification, pipeline-cycle accuracy and high visibility and accuracy.

Interactive vectorization hints



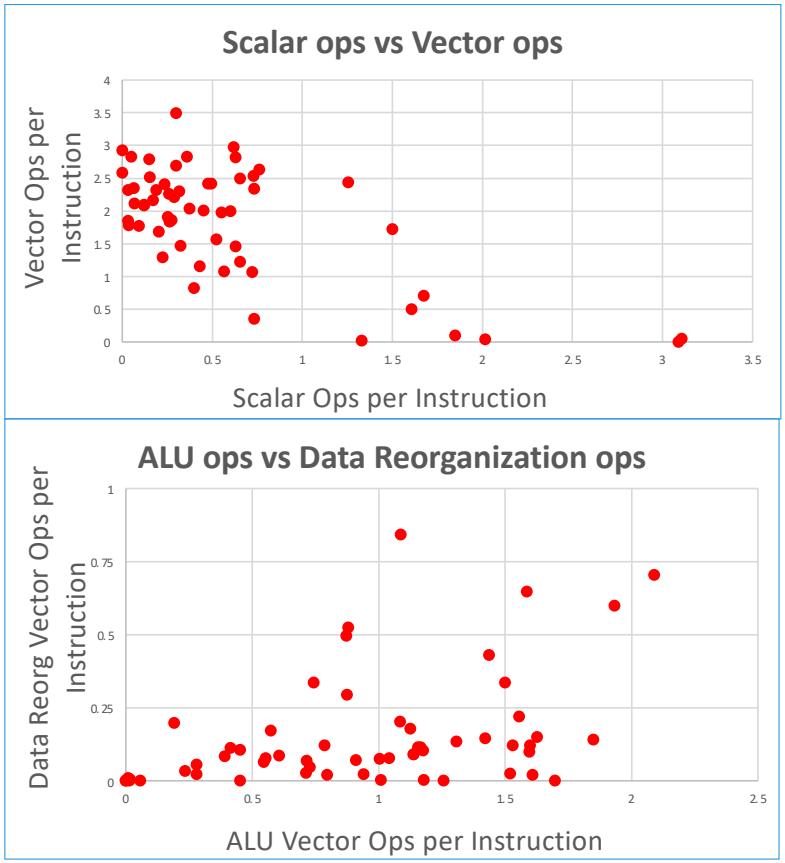
Application Diversity drives ISA flexibility

A look at 45 vision application kernels



- Typically several ALU ops per load operation
- Wide range of ALU : Load/store ratio (1:2 to 5:1)
- Many important functions don't do multiplies
- A fraction have very heavy multiply usage – e.g. convolutions
- ISA should handle wide range of ratios efficiently

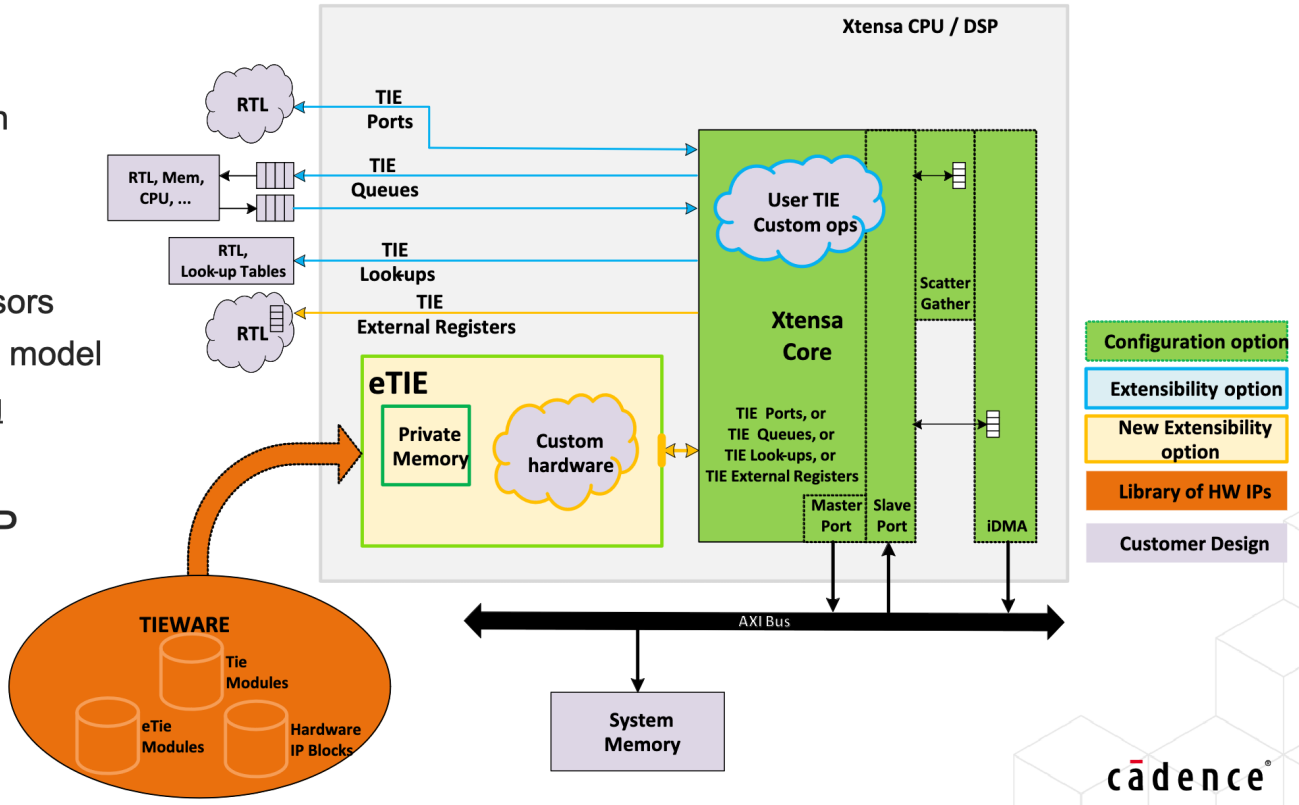
Application Diversity drives ISA flexibility



- A successful architecture maximizes the fraction of kernels that can be vectorized
- A small number of functions may still use scalar ops heavily
- On-the-fly data reorganization may be important in a few kernels
- ALU : Reorg ratio varies from 10:1 to 1:1
- Efficient data reorganization boosts benefit of vectorization

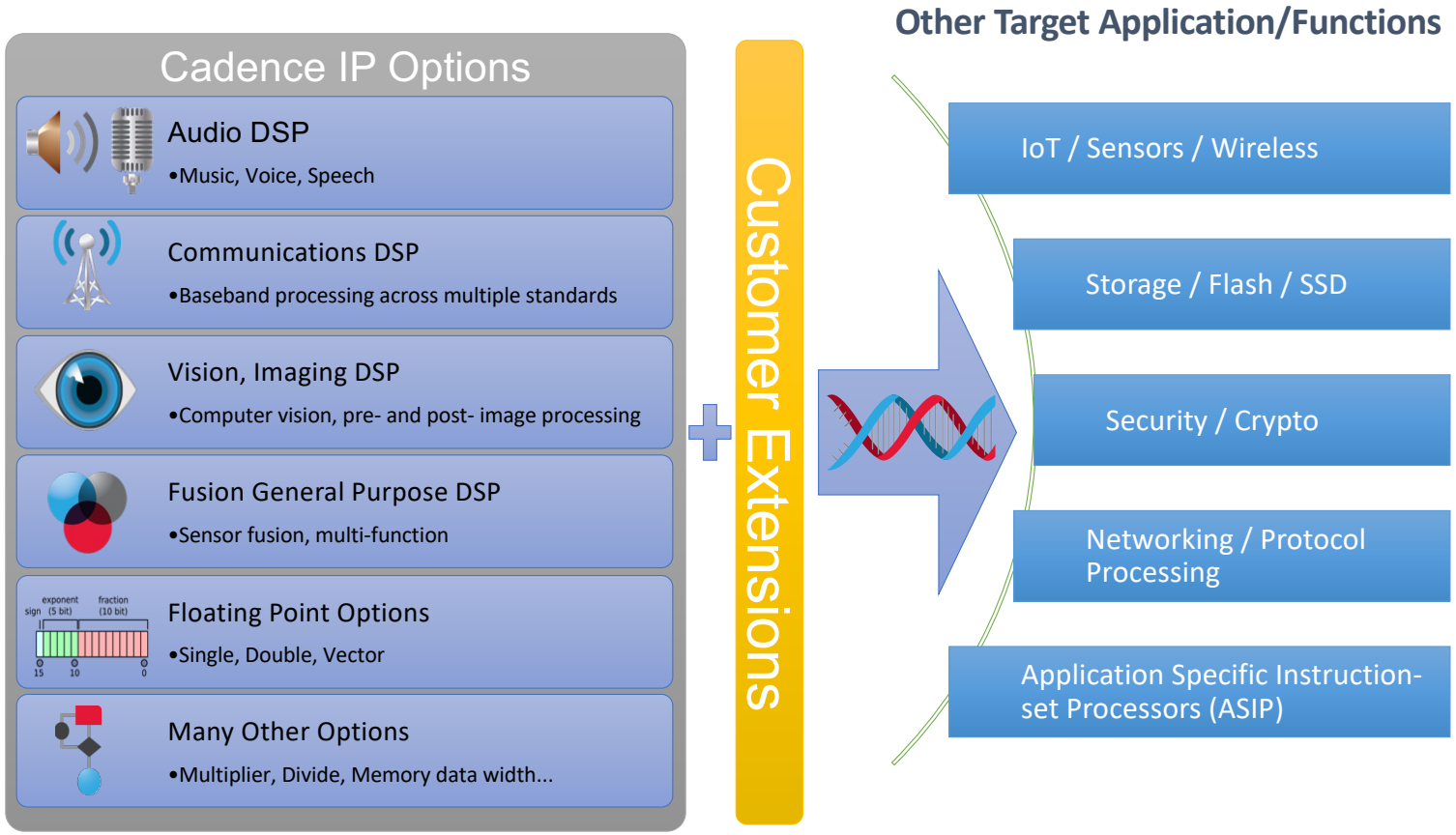
eTIE – New Extensibility option with Xtensa

- Enables generation of tightly-coupled hardware coprocessors
  - Specialized functions
  - Massive operational throughput
  - Intelligent IOs of increased dimension
- Auto-generation of development environment and tools
  - Cycle accurate C-model of coprocessors
  - Integrated coprocessor programming model
  - All with compiler, software and debug development tool chain support
- Library of pre-verified hardware IP components (TIEWare)

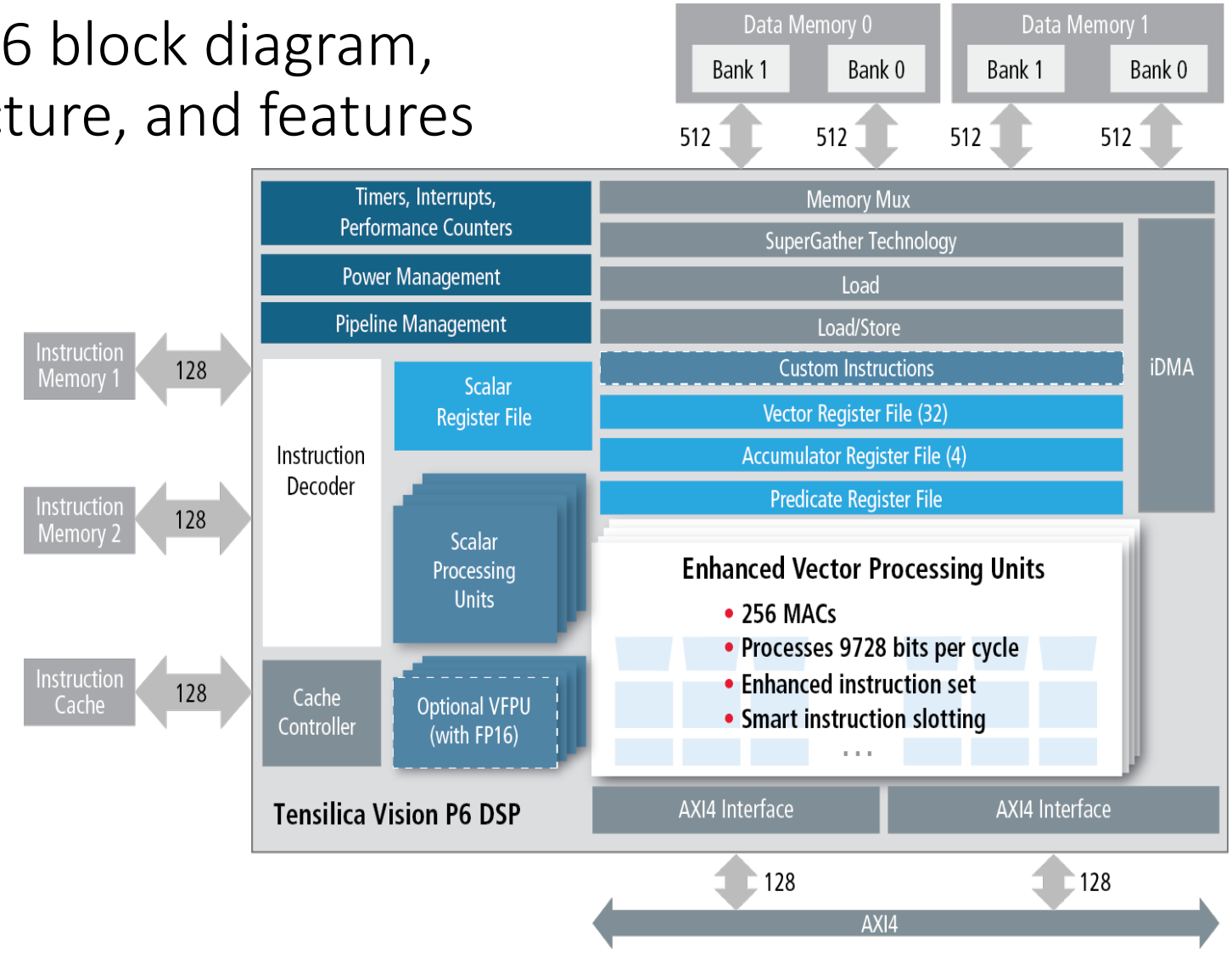


Tensilica Optimized Processors

Performance & Energy Efficiency… Differentiation



Vision P6 block diagram, architecture, and features



The most widely used DSP through the wealth of Standard Options
off-the-shelf with rich software environments

Tensilica ConnX DSPs for comms

	Tensilica ConnX DSPs			
Throughput	110	120	B10	B20
Vector/Memory width (b)	128	256	256	512
MACs 8bx8b	32	64	64	128
MACs 16bx16b	32	64	64	128
MACs 32bx32b	8	16	16	32
SP FP FMA	8	16	16	32
HP FP FMA	16	32	32	64
DP FP FMA	4	8	8	16
ALU 8b	48	96	96	192
ALU 16b	24	48	48	96
ALU 32b	16	32	32	64
Divider 16b	1	2	2	4
Divider 32b	0.5	1	1	2

Tensilica Vision DSPs

Tensilica DSP		Vision P1	Vision P6	Vision Q7	Vision Q8
Use Case					
MAX SIMD Width		128	512	512	1024
Xtensa Processor		LX	LX	NX	NX
MAC For AI	8x8	128	256	512	1024
	8x16	32	128	128	256
	16x16	32	64	128	256
	32x32	2	8	16	32
VFPU	16-bit Half-Precision	8	32	64	128
	32-bit Single-Precision	4	16	32	64
	64-bit Single-Precision	NA	NA	16	32
SuperGather		Yes	Yes	Yes	Yes
SLAM Acceleration		NA	NA	Yes	Yes

Tensilica HiFi DSPs for audio

Features		HiFi 1	HiFi 3	HiFi 3z	HiFi 4	HiFi 5
VLIW Slots		2	3	3	4	5
Fixed-Point MACs per Cycle	32x32	1	2	2	4	8
	32x16	2	4	4	up to 8	16
	16x16	4	4	up to 8	up to 8	16
Accumulator		64-bit	64-bit	64-bit	64/72-bit	64/72-bit
FPU (optional)		Integrated 2-way SIMD Vector FPU (VFPU)	Integrated 2-way SIMD VFPU	Integrated 2-way SIMD VFPU	2 integrated 2-way SIMD VFPU	2 integrated 4-way SIMD VFPU
ITU Intrinsic Support		Yes				
Circular Buffer Support		1	1	1	2	3
Bitstream VLE/VLD Support		VLD	Yes	Yes	Yes	Yes
User-defined instructions		Yes	Yes	Yes	Yes	Yes
Additional features		Special NN acceleration ISA				32 MACs/cycle NN MACs (optional) half-precision FPU

A closing thought

Specialization ➔ proliferation of designs

Specialized low-cost, low-energy platforms: wearables, smart-home, medical, industrial, mil-aero, sensor-rich **extreme fit** SoCs

All SOC Designs

General-purpose all-in-one system platforms: cellphone, PC, server, gateway – **extreme scale** SoCs