2008-01-0382

The challenges of devising next generation automotive benchmarks

Patrick Leteinturier &
Laurent Beaurenaut
Infineon Technologies AG

Agenda

- Automotive System Trends
- Market requirements
- Benchmark requirements
- Workload characterization
- New Benchmark
- Summary

Key Market Drivers

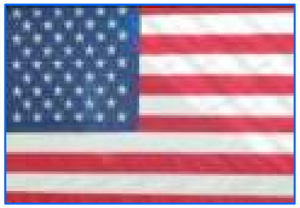
CO2 Reduction

Pollution Reduction

Safety

Fun to Drive

Global CO_2 Targets



It's the law: 35 mpg CAFE

Automotive News Dec. 19th 2007

Cars: 35 mpg by 2020

EU agrees to steep fines to cut car CO_2 from 2012

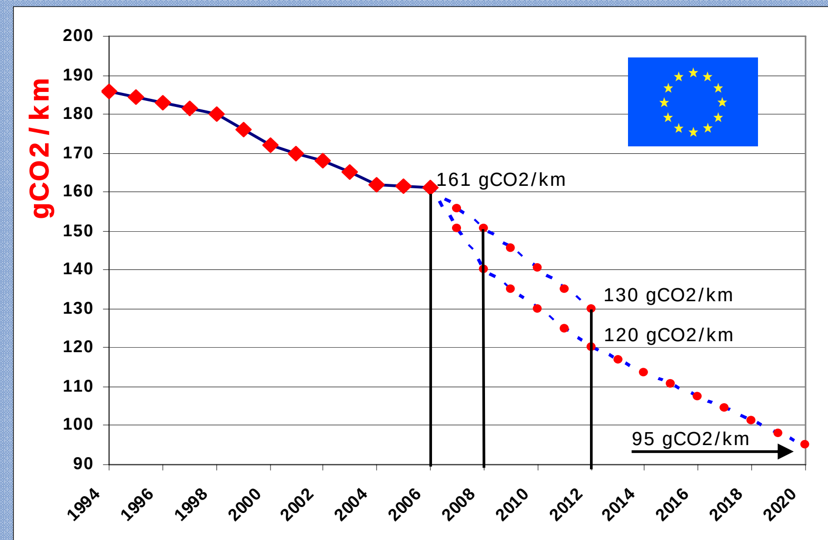
Automotive News Dec. 20th 2007

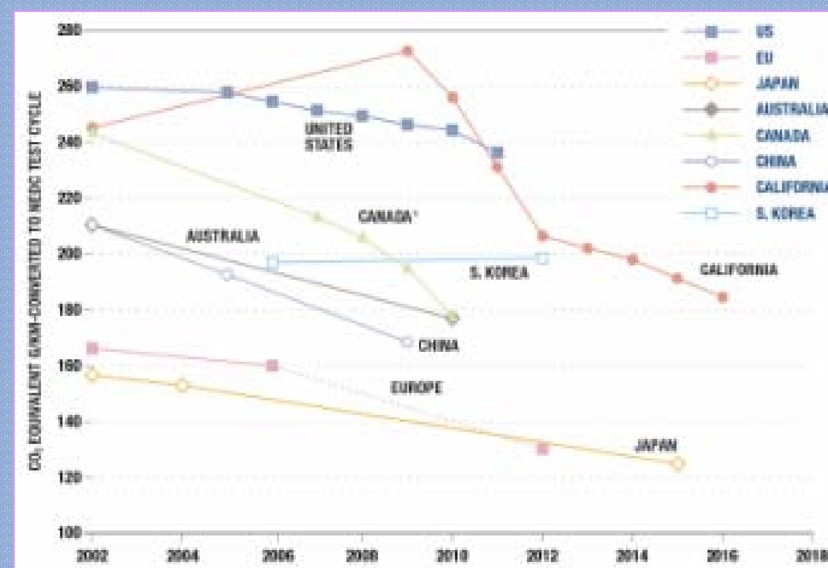
Cars: 120gCO2/km by 2012

+10 g coming from biofuels ...

Conversion table for regular gasoline engine

gCO2/km	155	140	130	120	110	100	90
L / 100km	6.72	6.08	5.65	5.21	4.78	4.34	3.91
MPG	35.00	38.69	41.66	45.13	49.24	54.16	60.18





2008-01-0382

Agenda

Automotive System Trends

Market requirements

Benchmark requirements

Workload characterization

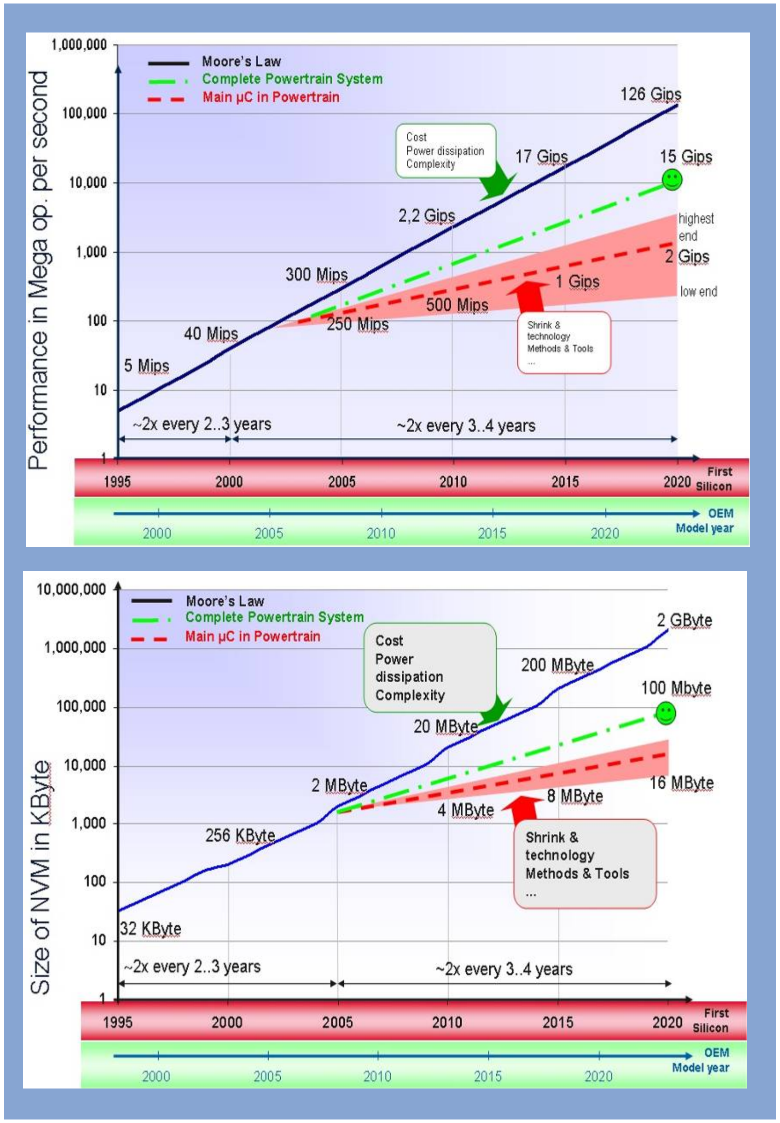
New Benchmark

Summary

General Architectural Trends and Requirements for Semiconductors

Trend	Application Examples		IC Requirements
Software-enabled functionality	Replacement of dedicated hardware with software algorithms running on µC	▶	Strong microcontroller cores with RT capabilities, broad peripheral set and eFlash
Decentralization to Centralization	µC-enabled global controls	▶	High performance processors with network connectivity
Centralization to Decentralization	Smart sensor networks, dedicated board nets	▶	Broad IP portfolio (sensors, µC, power) HVCMOS and advanced packaging technology
Analog/Digital Tradeoff	Replacement of signal processing and communication from analog to digital	▶	A/D and D/A conversion, signal conditioning and processing
X-by-Wire	Mechatronical solutions for steering, breaking instead of mechanical systems	▶	RT capabilities, failsafe electronics

Software-Enabled Functionality
Increased Microcontroller Performance



Semiconductor Industry provides a 30% to 60% annual performance increase at same cost

- Software platform, reuse of software modules across application and customers
- Migration of functions from hardware to software
- Hardware independent Software
- Wider use of automatic code generation
- Software standardization e.g. Operating system, Drivers with application level interfaces (OSEK, AutoSar, IEC61508, ISO26262…)
- Robust, transparent software e.g. encapsulation, software self test
- µC family concept with performance increase and easy migration path

Agenda

Automotive System Trends

Market requirements

Benchmark requirements

Workload characterization

New Benchmark

Summary

Benchmarks that Predict Performance in Real-world Applications

Not MHz

- Only provides relative performance analysis

Not Dhrystone

- No regulation
- No memory or cache effects
- Optimizes to nothing

Usage Models

- Is a substitute of the real application to model the system performance and memory size utilization
- Analyze, tune, and validate new processor architectures
- Design and analysis of system-level implementation
- Compare and select processors according to more important criteria than MHz

Vision of the benchmark

The vision is to establish this new benchmark as an automotive tool to specify and measure performance between:

1. OEMs
2. Tier1s
3. Silicon vendors
4. 3rd party tool vendors

The relevance of the benchmark

The automotive is more going toward the system benchmark

1. Static Benchmark
   - Test of given algorithm
2. Dynamic Benchmark
   - Test of response, switch context...
3. Functional Benchmark
   - Test of complete function CAN, LIN, PWM, ADC
4. Auto code Benchmark

Agenda

Automotive System Trends

Market requirements

Benchmark requirements

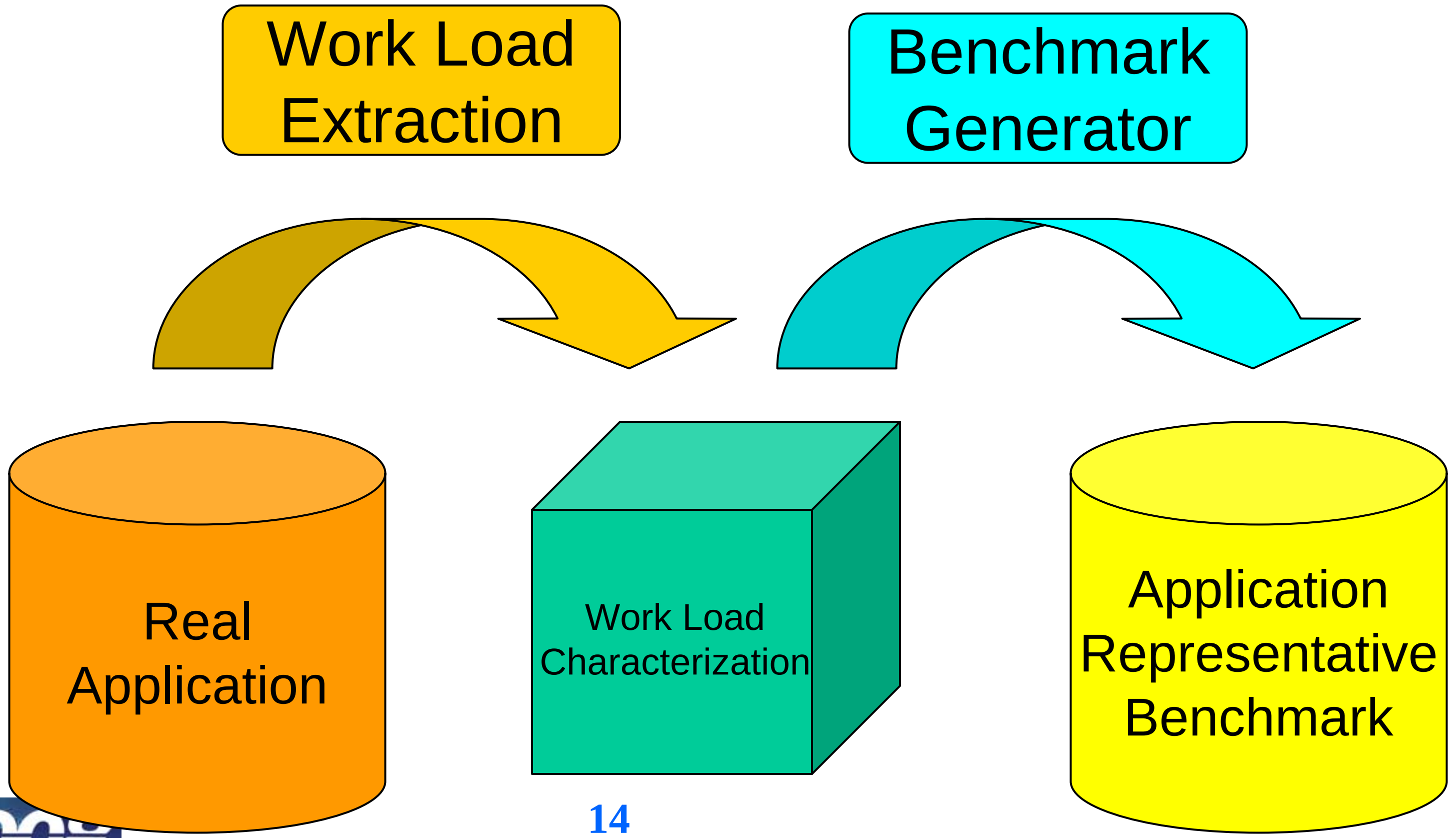
Workload characterization

New Benchmark

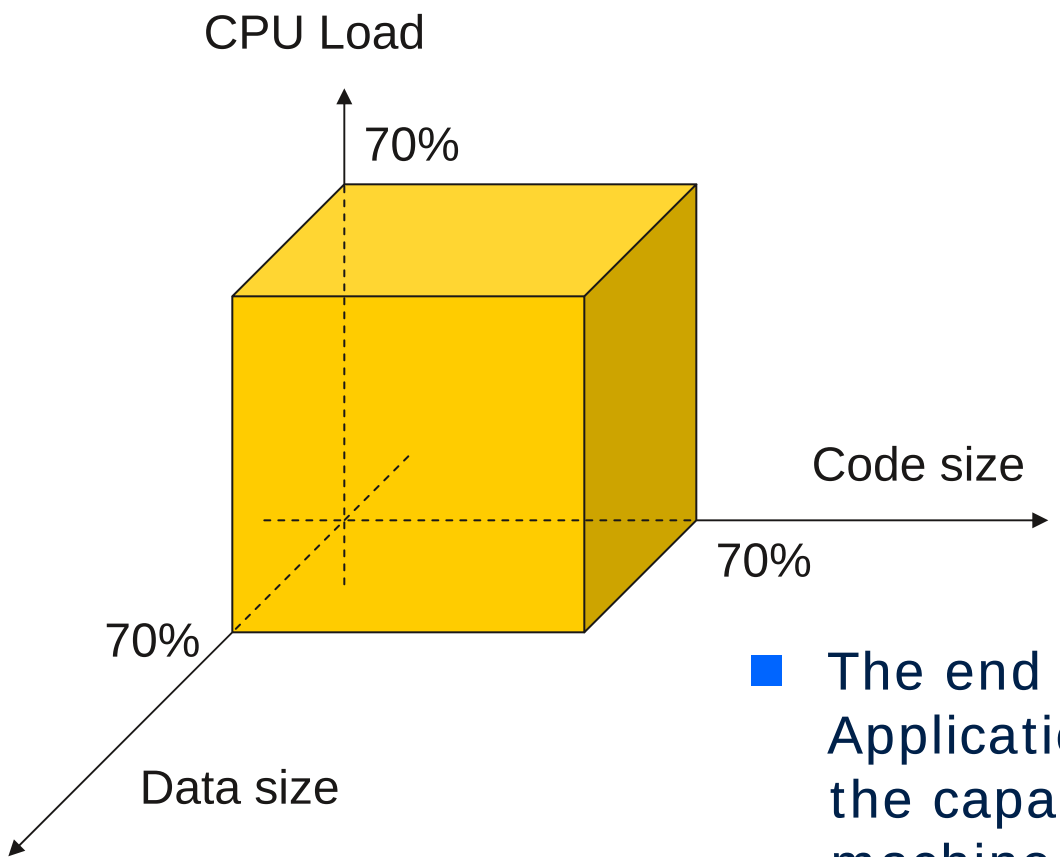
Summary

Proposal EEMBC Automotive Suite V2

- Two big needs
  1. Work load extraction
  2. Application Representative Benchmark generator



Application Characterization

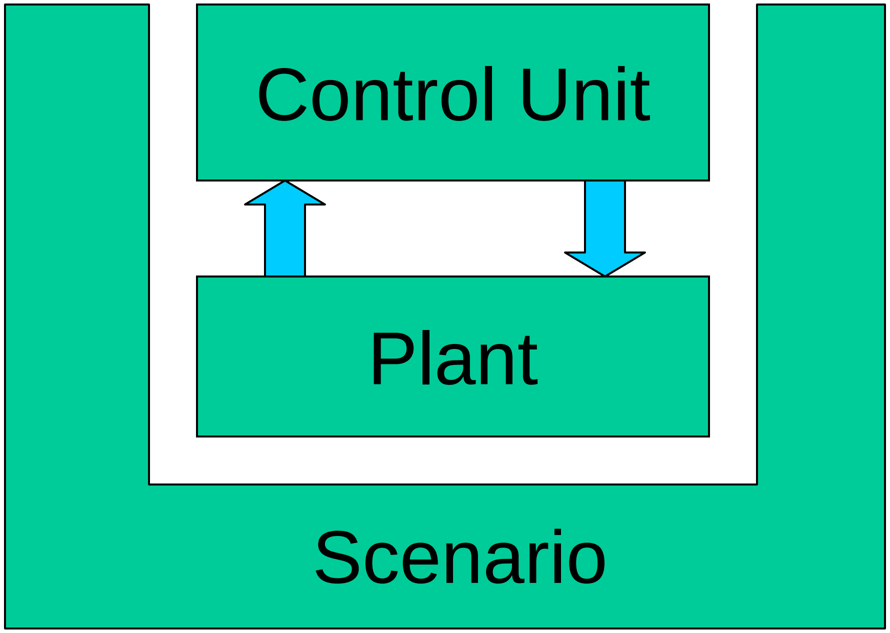


- The end customers like the Application to be at 70% of the capability of the machine at the first SOP of the product / platform

Embedded Software

To assess performance of a µC for an application is important to have a good knowledge of:

1. Control Algorithm
2. Plant Algorithm
3. Scenario



Work load characterization

At Source Code level

1. This analyses the source code
2. Without care of the use cases

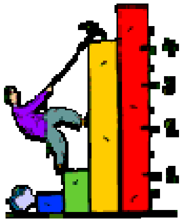
At Trace code level

1. This analyses a given code trace
2. For a give plant and a given scenario
3. E.g. Full load, Idle, Engine Acceleration..

Benchmark Characterization

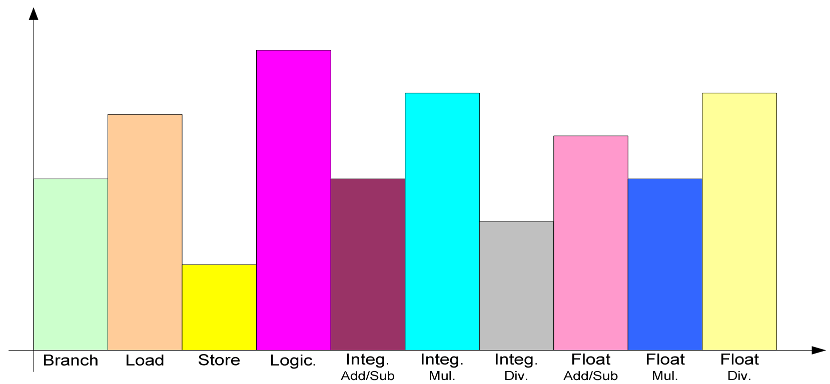
Multiple parameters:

1. Instruction distribution
2. Inherent instruction-level parallelism (ILP)
3. Branch predictability
4. Inherent floating-point (FU) usage
5. Minimum cache size to minimize misses



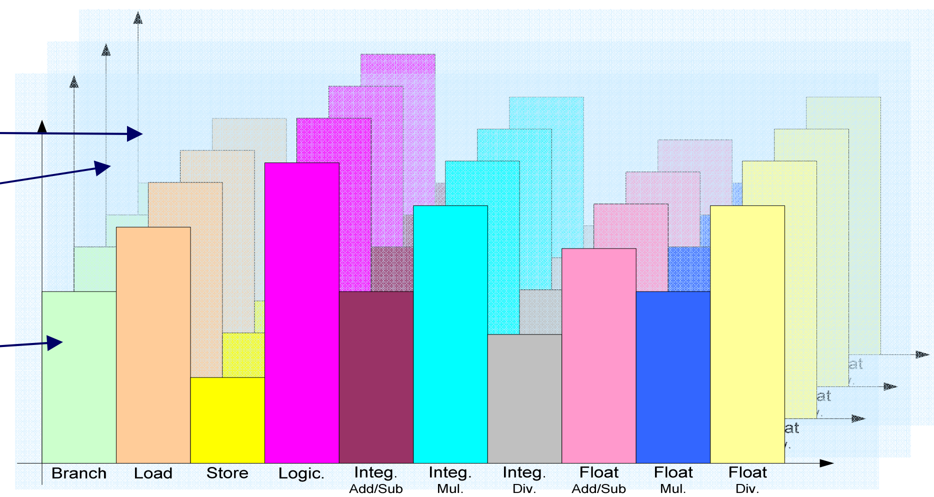
Instruction distribution at Application level

At Source Code level



At Trace Code level

- Trace 1
- Trace 2
- Trace n



Agenda

Automotive System Trends

Market requirements

Benchmark requirements

Workload characterization

New Benchmark

Summary

Algorithm selection

Arithmetic Fundamentals

Bit manipulation

Basic Calculation, Saturation & Rounding

Polynomial equations and limited dev.

Table look-up & Interpolation

Zeros of a 2nd order equation

Trigonometric

Logarithmic, Exponential, X Power Y

Vector and Matrix functions

Filters

De-bounces

Schmidt trigger

Finite Impulse Response (FIR) Filter

Infinite Impulse Response (IIR) Filter

Fast Fourier Transform (FFT)

Inverse Fast Fourier Transform (iFFT)

Discrete Cosine Transform (DCT)

Inverse Discrete Cosine Transform (iDCT)

Control

Finite state machine

PID

Kalman + LQG

Statistics

Autocorrelation

Convolution

Mean Value, Variance, Standard deviation

Linear regression

Random sequence

Instruction distribution

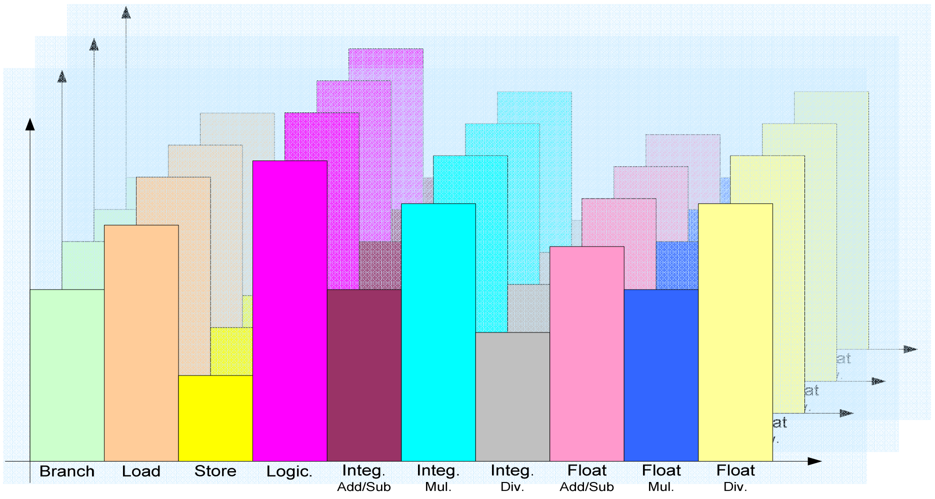
At Reference Algorithm level

Ref. Algorithm 1

Ref. Algorithm 2

Ref. Algorithm 3

Ref. Algorithm n

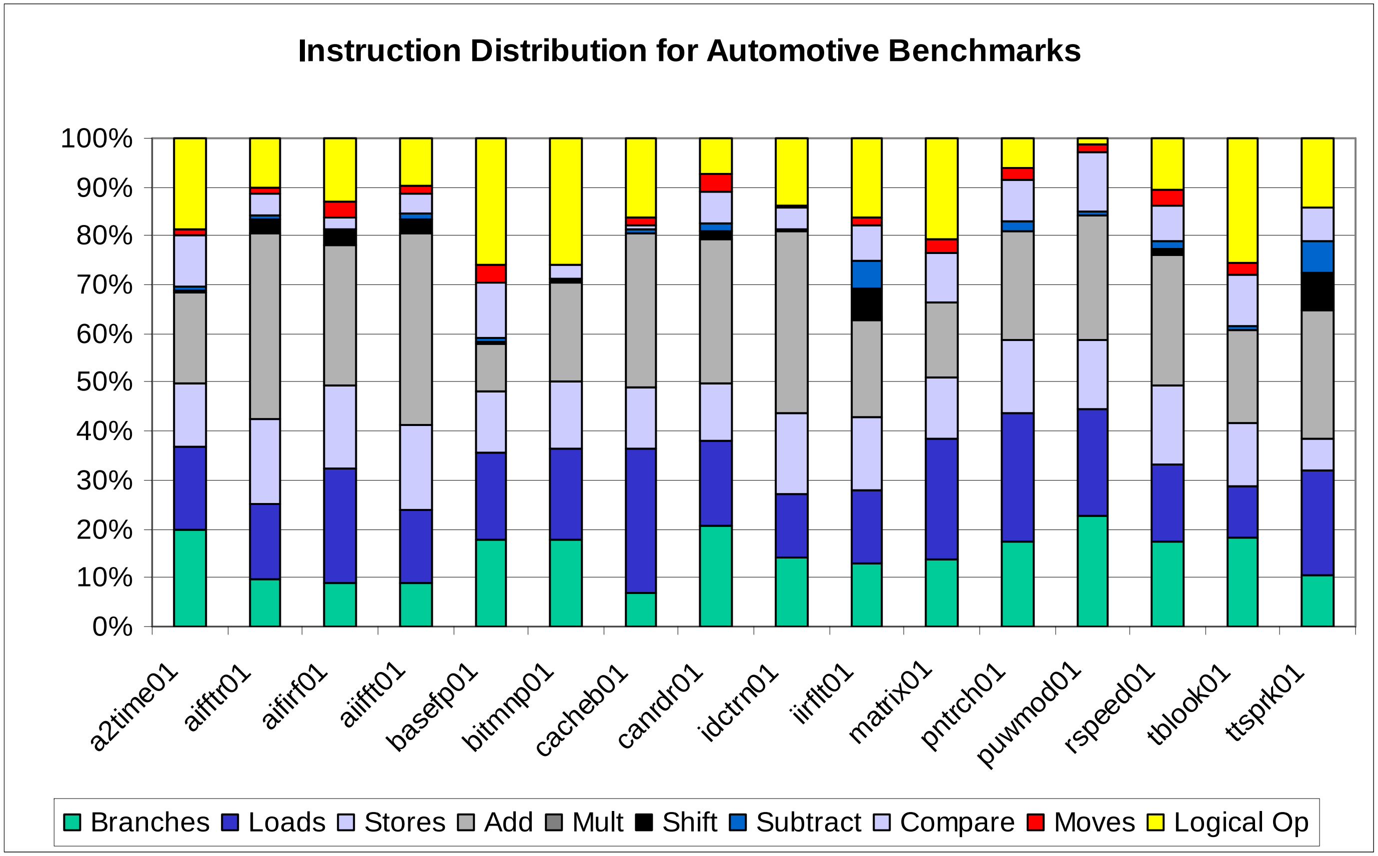


Level of definition

Algorithm parameters (Format, dimension..)

Algorithm data set

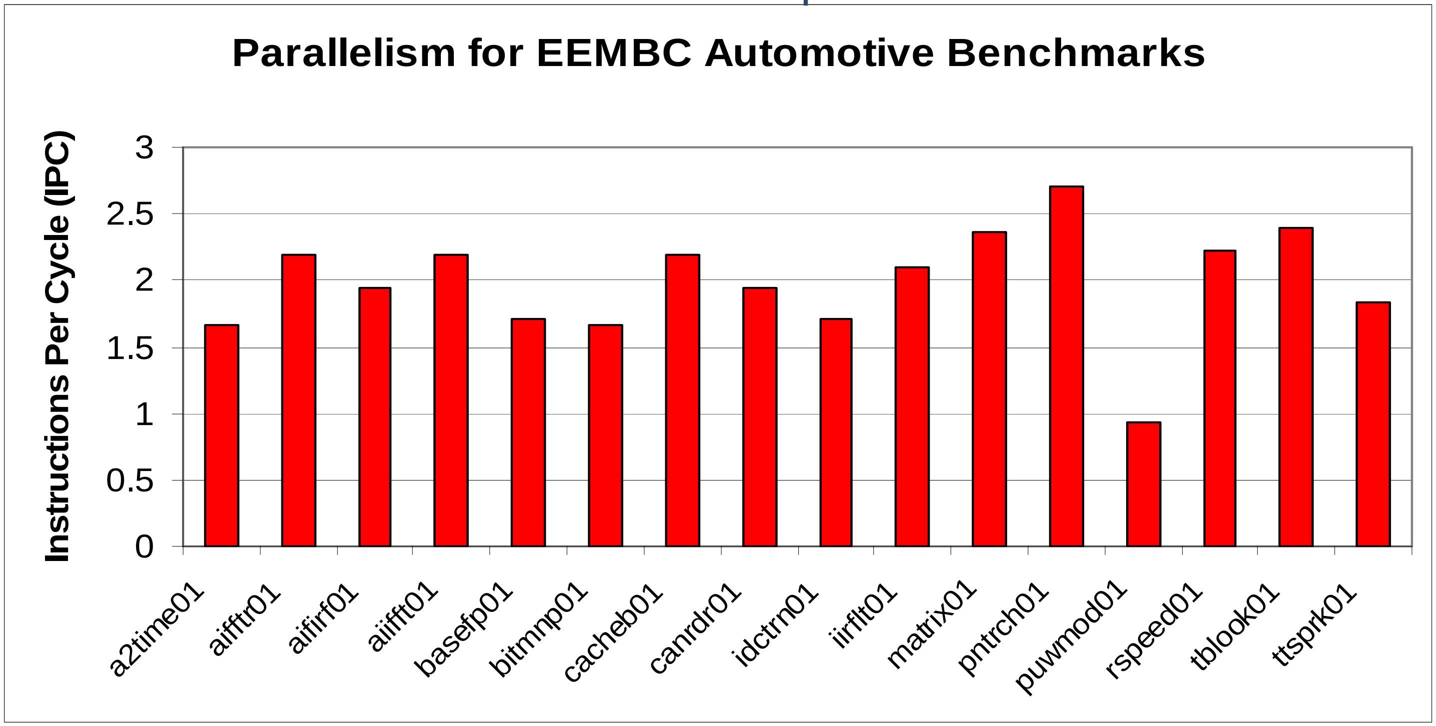
Instruction distribution



EEMBC Automotive V1

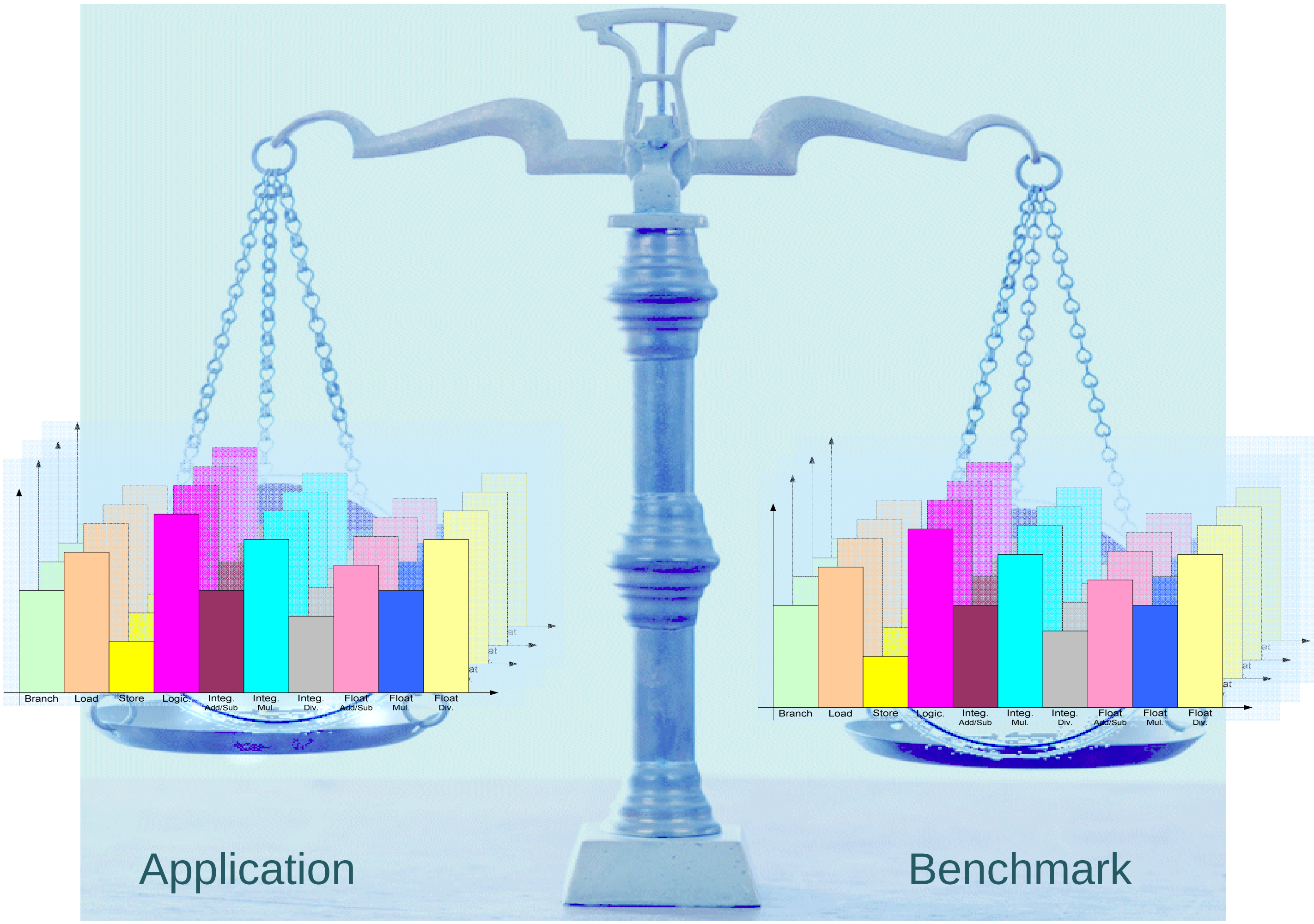
Inherent ILP

Inherent instruction-level parallelism



EEMBC Automotive V1

Benchmark Fitting / Tuning

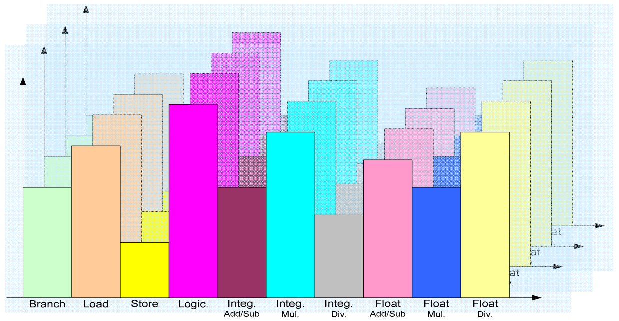


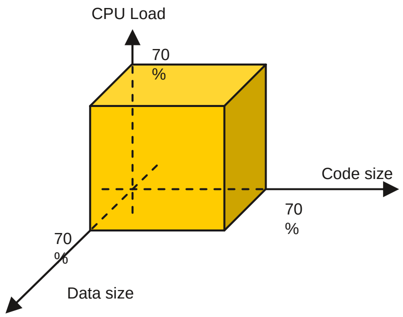
Benchmark Fitting / Tuning

Customer Code

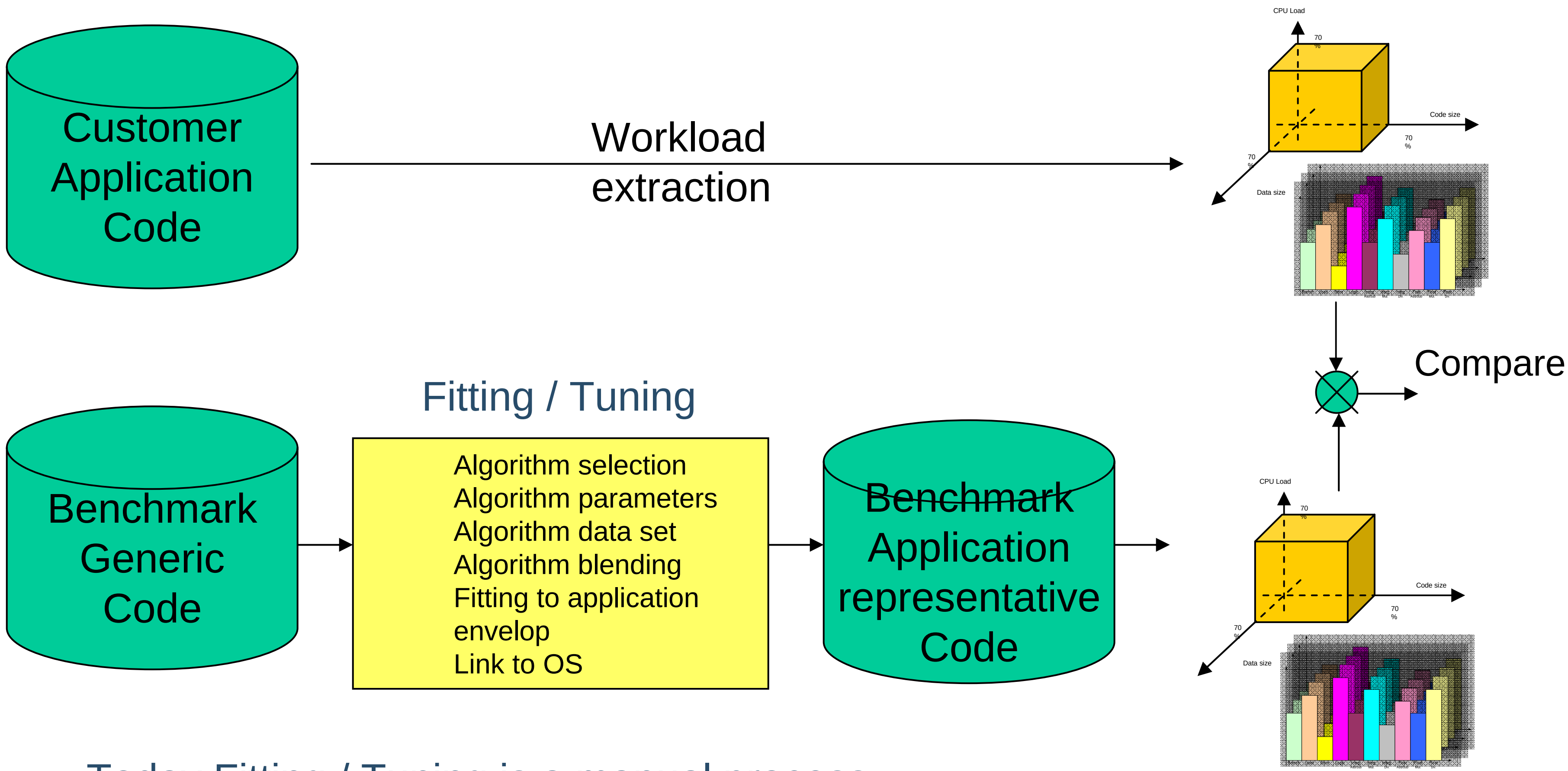
Benchmark Code

- Algorithm selection
- Algorithm parameters
- Algorithm data set
- Algorithm blending
- Fitting to application envelop
- Link to OS



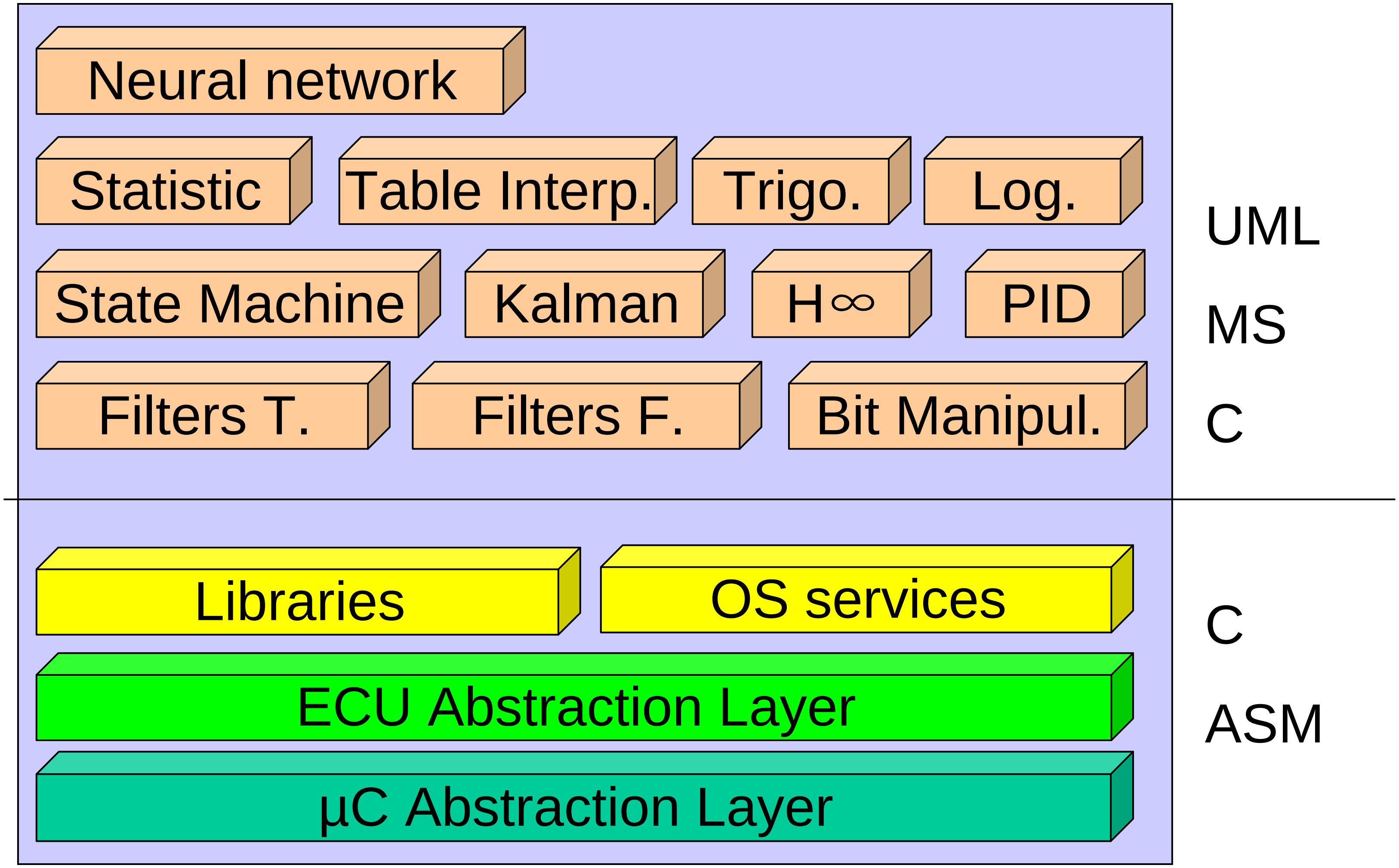


Benchmark Fitting / Tuning



Today Fitting / Tuning is a manual process
in the future this could be semi-automatic or fully-automatic

Software architecture (Static view)



Agenda

Automotive System Trends

Market requirements

Benchmark requirements

Workload characterization

New Benchmark

Summary

Summary

- ➔ It is possible to create a benchmark to substitute the application code and be representative.
- ➔ This benchmark is a tool share performance information between OEMs, Tier1s, Silicon vendors, SW vendors and Tool suppliers...
- ➔ This benchmark is a tool to better architect HW and SW for next generation Automotive Electronics.

Acknowledgement

EEMBC: The Embedded Microcontroller Benchmark Consortium

http://www.eembc.org

Markus Levy
EEMBC President

Shay Gal-On
EEMBC Director of Software Engineering

The challenges of devising next generation automotive benchmarks

Thank you for your attention

Q&A