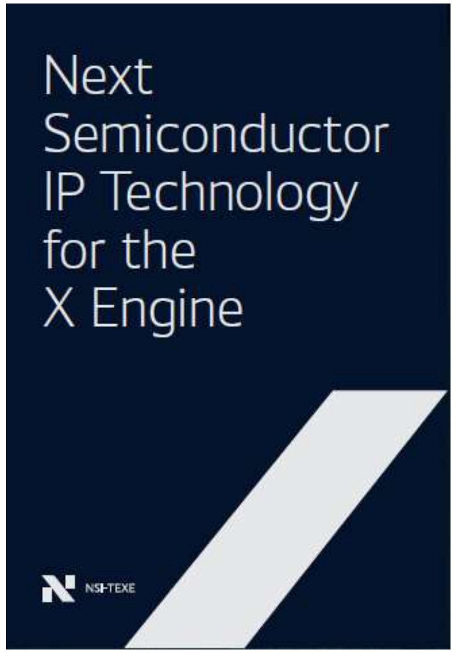
NSI-TEXE

Architecture Challenge for Future Automotive or Embedded Compute SoC

Hideki Sugimoto
CTO NSITEXE

July 8th, 2019



Next
Semiconductor
IP Technology
for the
X Engine

NSI-TEXE

Contents of Today

- Why is ADS*1 difficult?

  *1: Autonomous Driving System

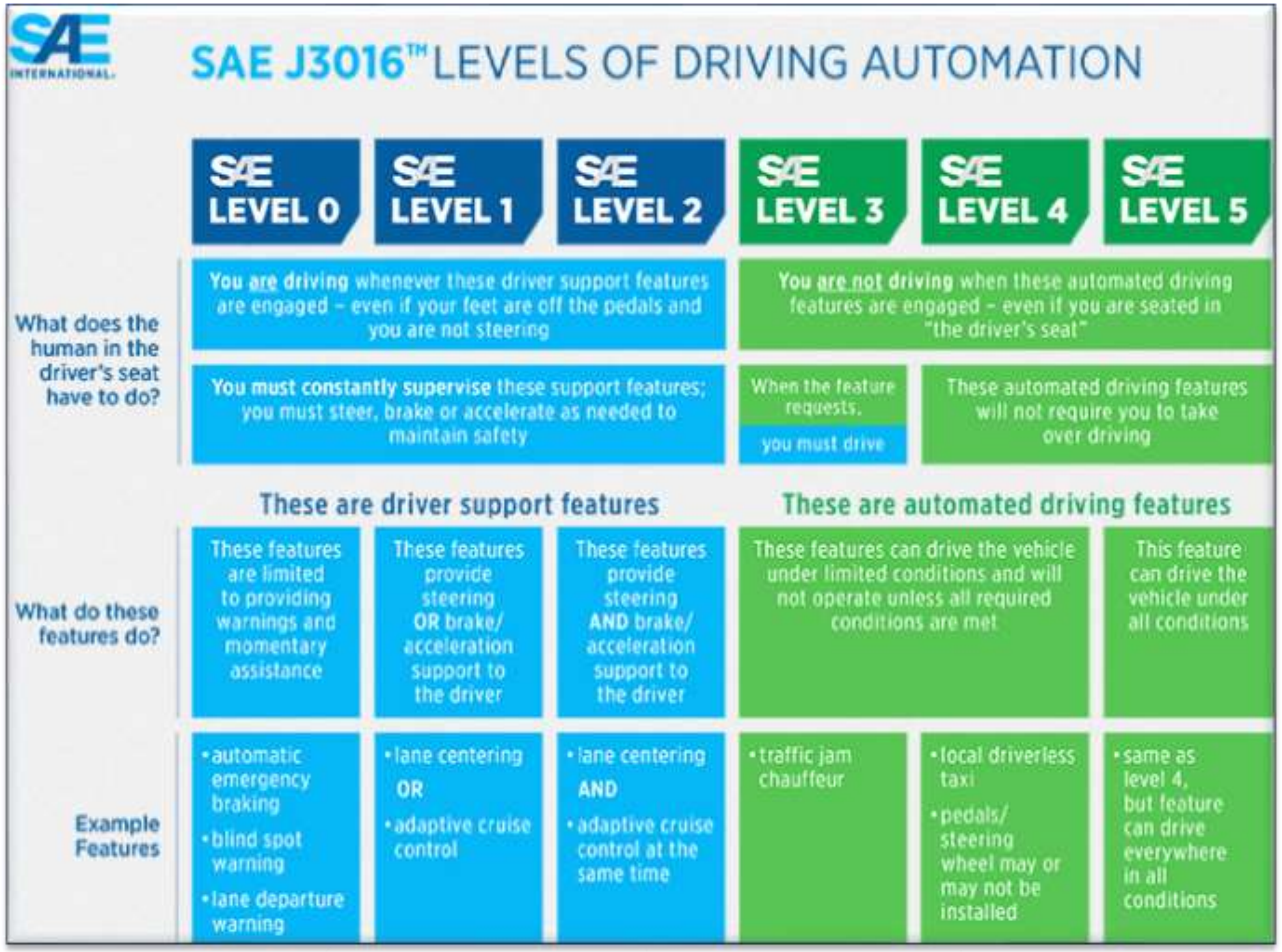
- Specificity of the AD computer
- Specificity of the AD SoC
- Architecture Challenge for Future Embedded Compute SoC

1. Why is ADS*1 difficult?

*1: Autonomous Driving System

ADAS/ADS Lv.1 to Lv.5

- Performance
  - Lv.4 ADS requires over 100TOPS (System Total) of Compute Performance
- Design Methodology
  - Design time should be less than five years (H/W) or three years (S/W)
  - “Automotive” Quality
- Maintenance
  → Update will be needed to measure future (unexpected) risks



SAE INTERNATIONAL

SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/ acceleration support to the driver	These features provide steering AND brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering OR • adaptive cruise control	• lane centering AND • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/ steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

➔ System should support many types of functions and performance

Requirements for ADS

- More comfortable
  - ➢ Can a Dynamic Map replace Human Skills for more comfort drive?
  - ➢ Cloud based, Fog based or In-vehicle compute?

  …

- More secure
  - ➢ What type of information should be provided to passengers or other drives for security?
  - ➢ How do people feel secure? Functions? Performance?...

  …

- More safe

  → Now just stopping is no longer safe.
  (e.g. DoS attack may shutdown all traffics in the citiy)

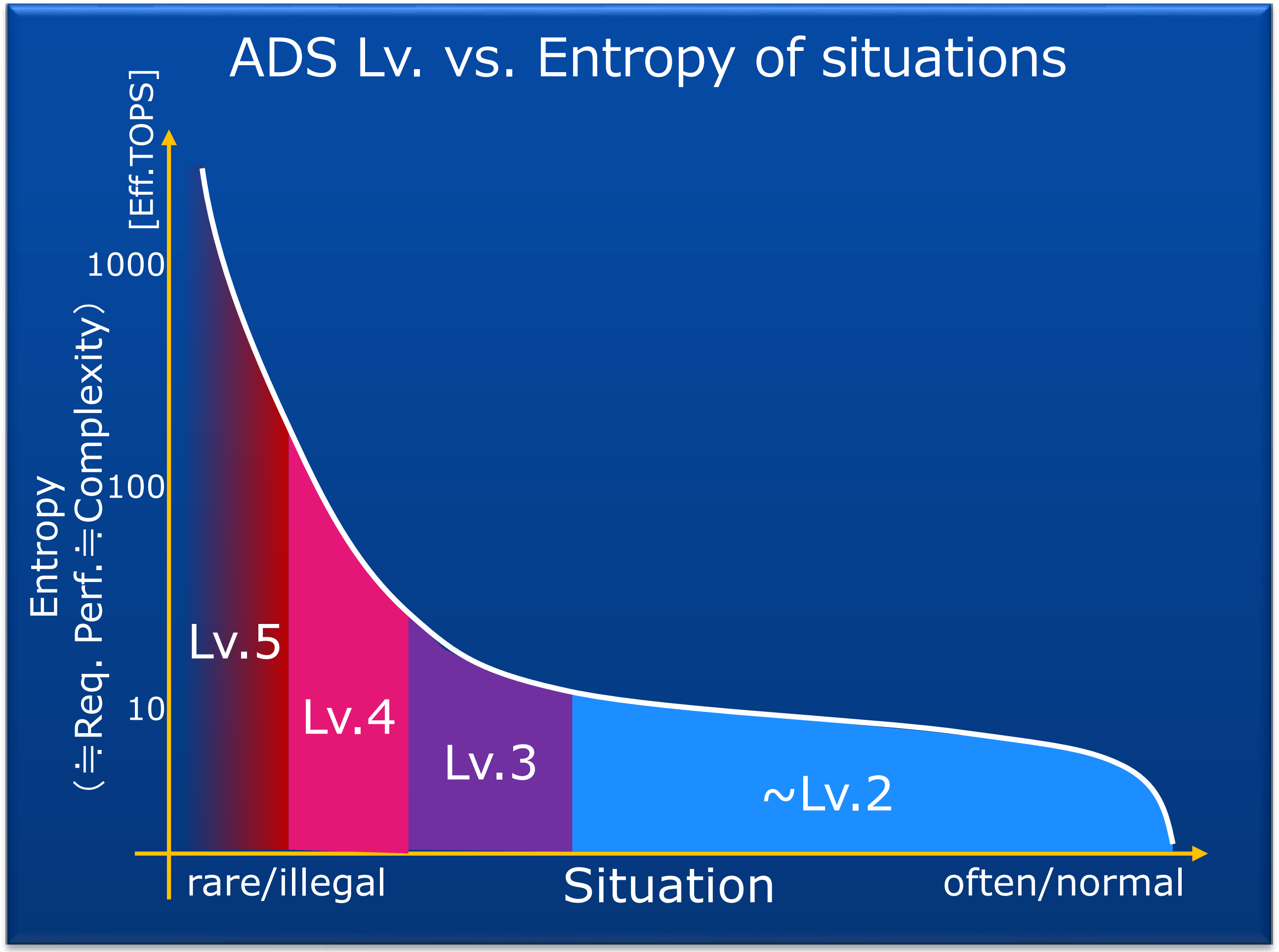
➔ Complex factors such as law, social risk tolerance, and road conditions are intertwining and unifying will be difficult.

Difference of responsibility

ADAS Function		Status/Scene	AD Func.
Driver	**System**		**System**
Monitor ADAS Behavior and Override As Needed	Avoid accidents	Detectable hazard e.g. A object in front of you (All sensors can detect)	Avoid accidents
Monitor risks and prevent or avoid them as needed	Notify Risks	Detectable risk e.g. Other car cut the line (All sensors can detect)	
	Should do nothing	Risk of incomplete detection e.g. Seems a hole but dirty? (Some sensors will detect)	Monitor risks and take mitigation actions as needed
		Risks that are difficult to detect or difficult to handle e.g. Suddenly rushes	Anticipate risks and take preventive measures as needed
		No risks	

Compute Power Requirement for ADS

- Higher level of ADS requires extremely higher performance!
  - Lv.3 ADS will covers major "normal" driving situation
  - Lv.4 ADS should handle more complex and many variation of rare situations
  - Lv.5 ADS should handle basically all situations
  - ➔ Rare cases may have more complex and difficult scenario for safe drive
  - ➔ It may requires variable algorithms which will be difficult to handle by H/W



How difficult it is? – Example

- Processing explodes when trying to do as much as possible
  - ➢ In worst case estimation with 7nm node, several or several tens of kW power will be needed even if we use “good” hardware
  - ➢ If we can prioritize processing (based on necessity), we can slightly decrease total processing, but···
  - → It should be possible, but we cannot know correct answer now.
  - → It means specific hardware is not feasible for this usage.
- How we can say “enough” safe?
  - ➢ Aircraft may crashing in front of you?
  - ➢ Parts of other vehicles will dive to you?
  - …

➔ Sensor information is not enough to handle all these cases.

➔ Driving “Experience” may be included inside decision algorithms.

Example1: What is this object?

- Obstacles? Just dirt? or fake?
  - ➢ Only mono camera will detect this as Obstacles, but···
  - ➔ Can we ignore this as a noise?
- ADAS
  - ➢ Driver should make judgment if indeterminate
    (System do nothing)
- AD
  - ➢ Handover to driver → no time···
  - ➢ Handle as Obstacles → May too often stops

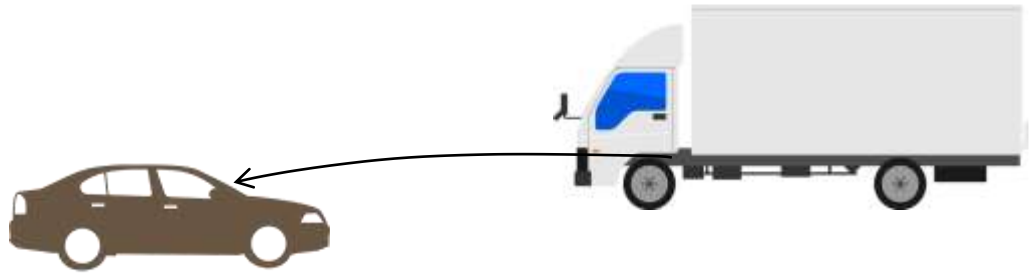
➔ Simple sensor fusion solution will not work for this case

➔ Intelligent (knowledge based) decision will be needed but it may need more compute power!

Example2: Compute Power vs. Latency

An object flying from the opposite lane at a relative speed of 160 km/h

- Sensor @60fps: The amount of movement of one frame is 74cm
- Assuming that it flies at a relative angle of 45 °, it can be recognized in 50 to 100 frames before the collision
- Need to recognize in a few frames considering the evasion judgment and the time required for the evasion action



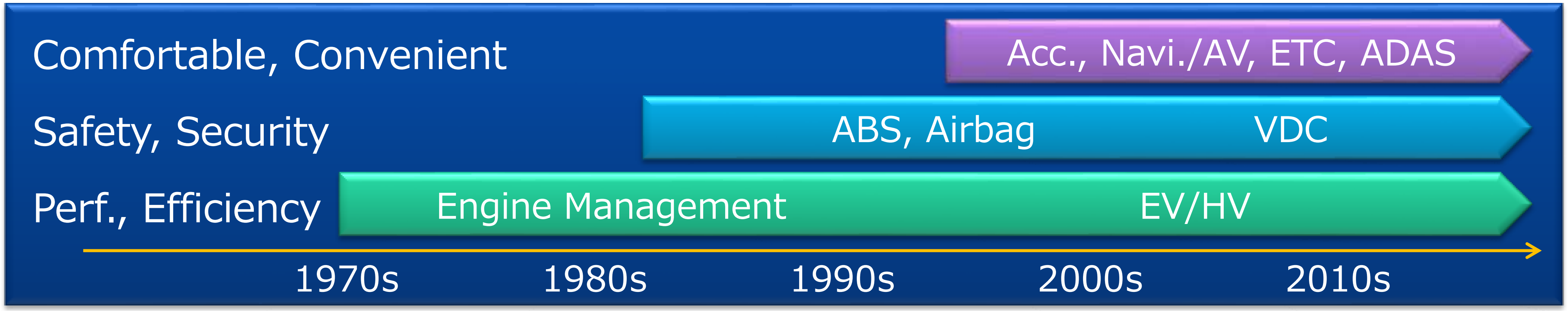
In the example of the fake picture in previous page, it is needed to traces about 20 frames to detect that is fake.

➔ How we can do within latency req.?

2. Specificity of the AD computer

LSIs in Automotive

- Support functions to the main role of the function

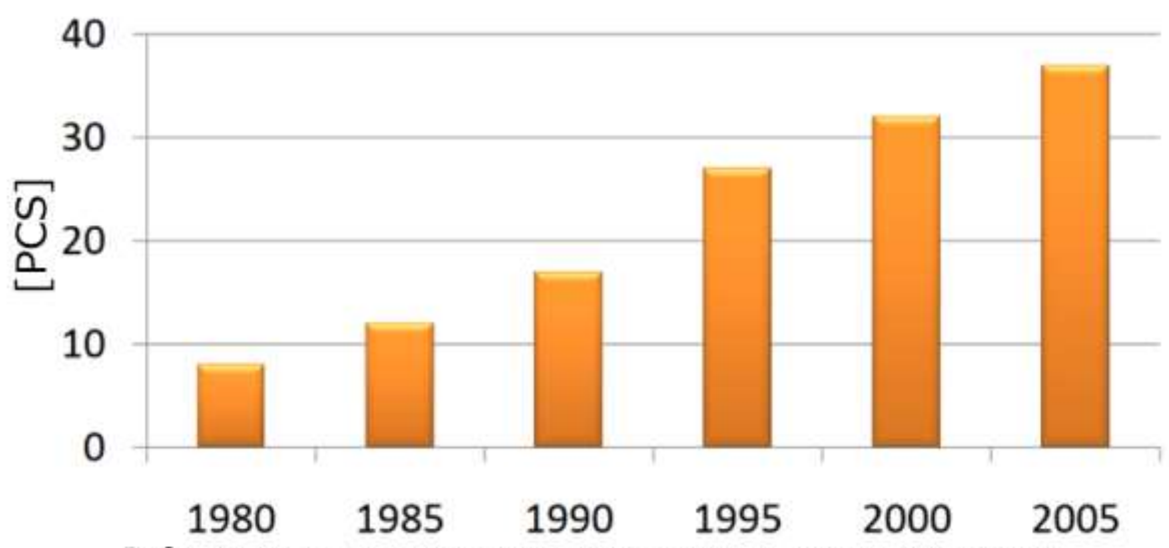


→ The number and type of LSIs has been rapidly increasing

ex.) Transition of the number of "microcomputers" used for one Mid Class vehicle

＊Over 100 for High-end Class

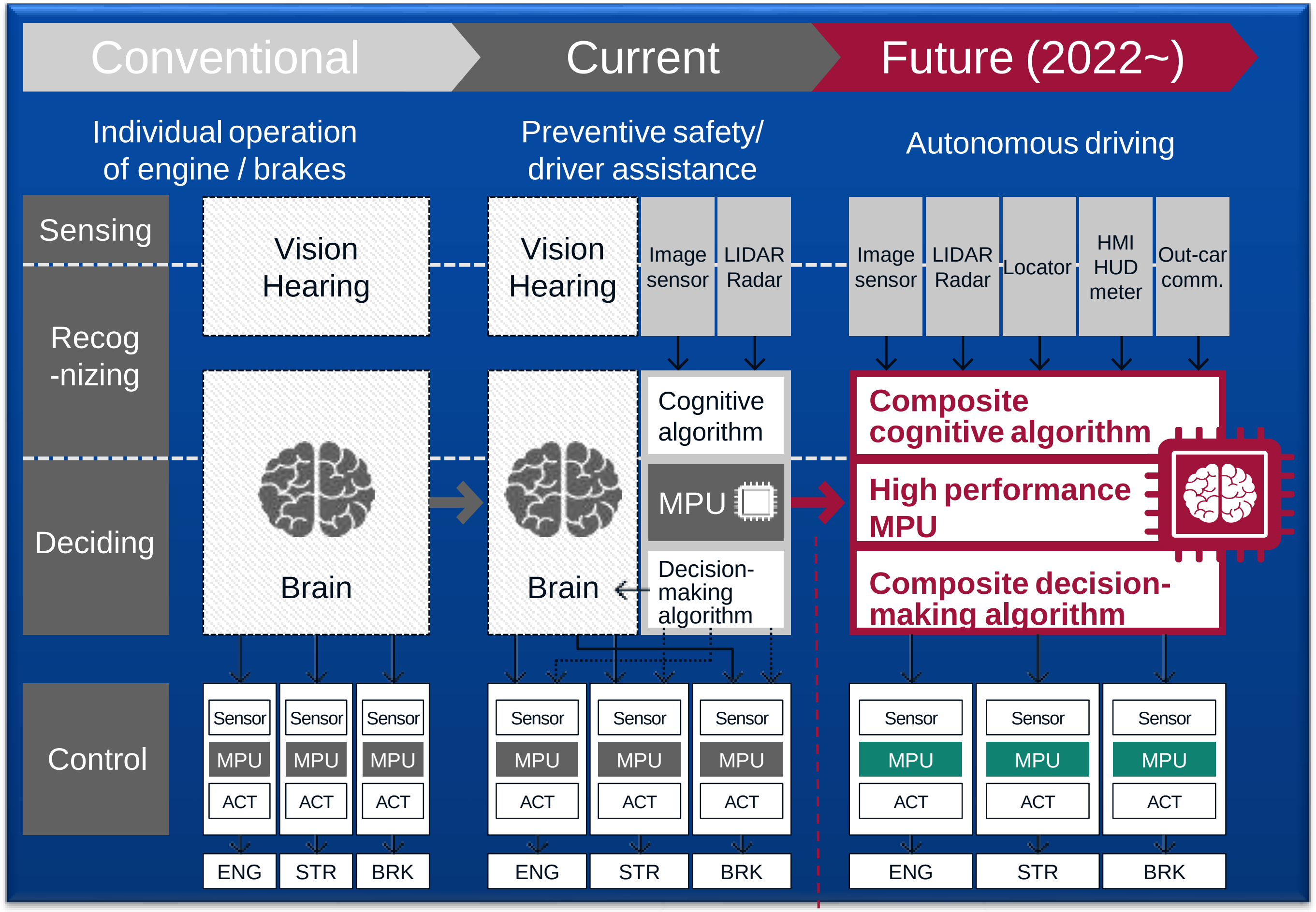
→ Needs dozens of "microcomputers"

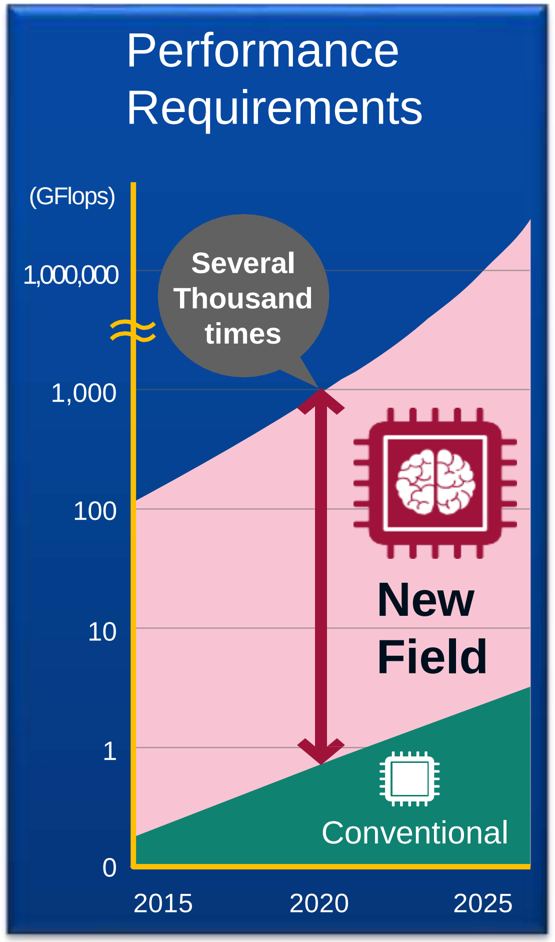


Reference: 矢野経済研究所, "車載用MCUと半導体メーカの自動車戦略2005"

Control vs. Compute in Automotive

- ADS requires huge Performance for itself and also for Control

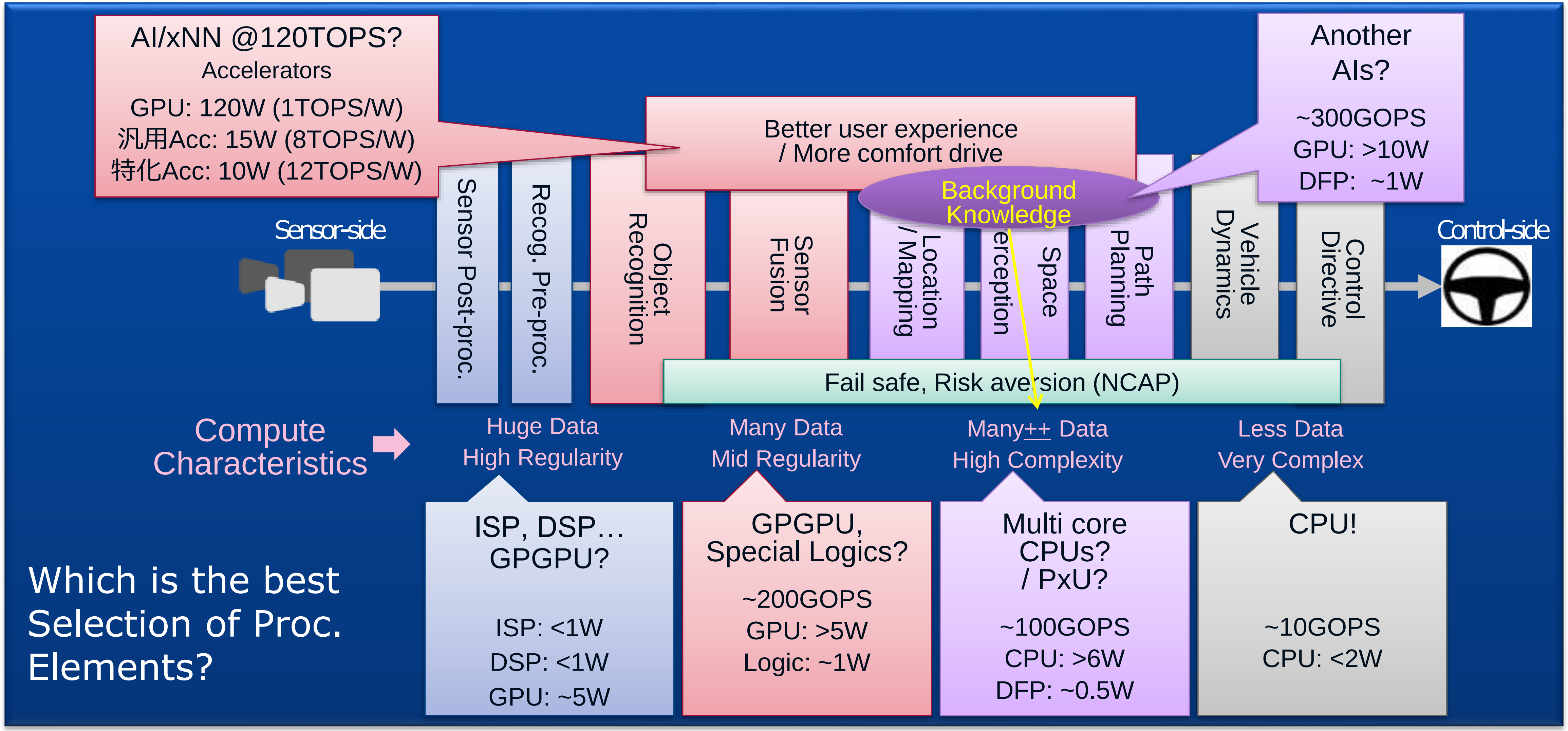




Automotive equipment requires higher-performance processors for both “New Field” and “Conventional” MPUs/MCUs

Situation of Today

- ADS requires multiple semiconductor IPs



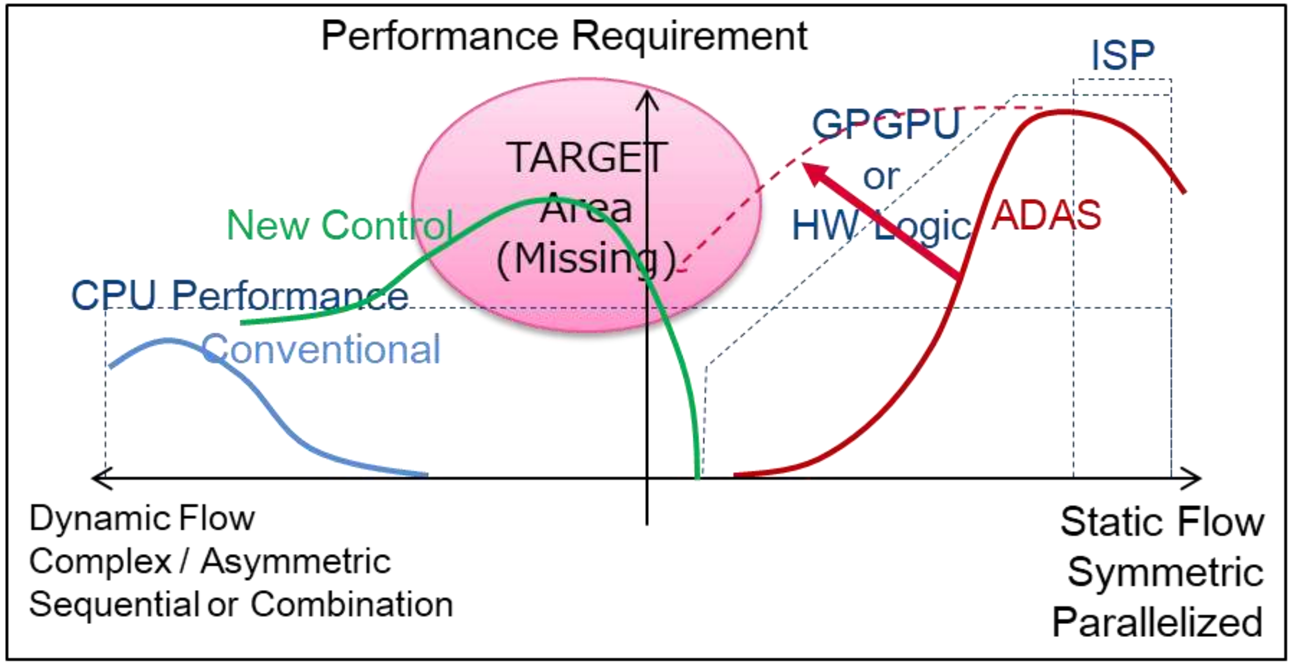
Why need to implement so many IPs?

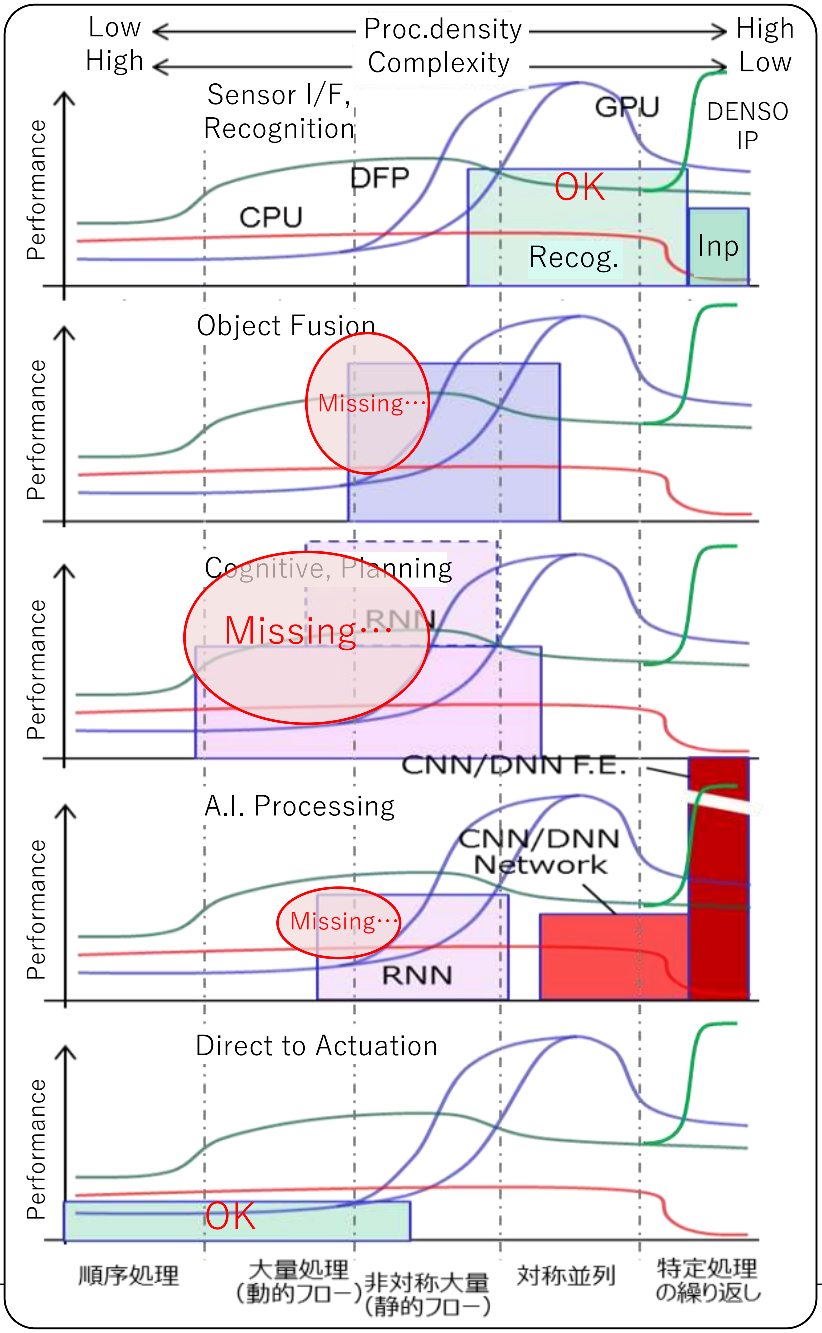
- Compute Characteristics vs. Efficiency

CPUs no longer cover all applications

- Gate/Wire delay trend in DSM technologies
- Parallelism vs. dependency
- Complexity, How to verify…

Requires parallel processing of asymmetric and dynamic processing graphs



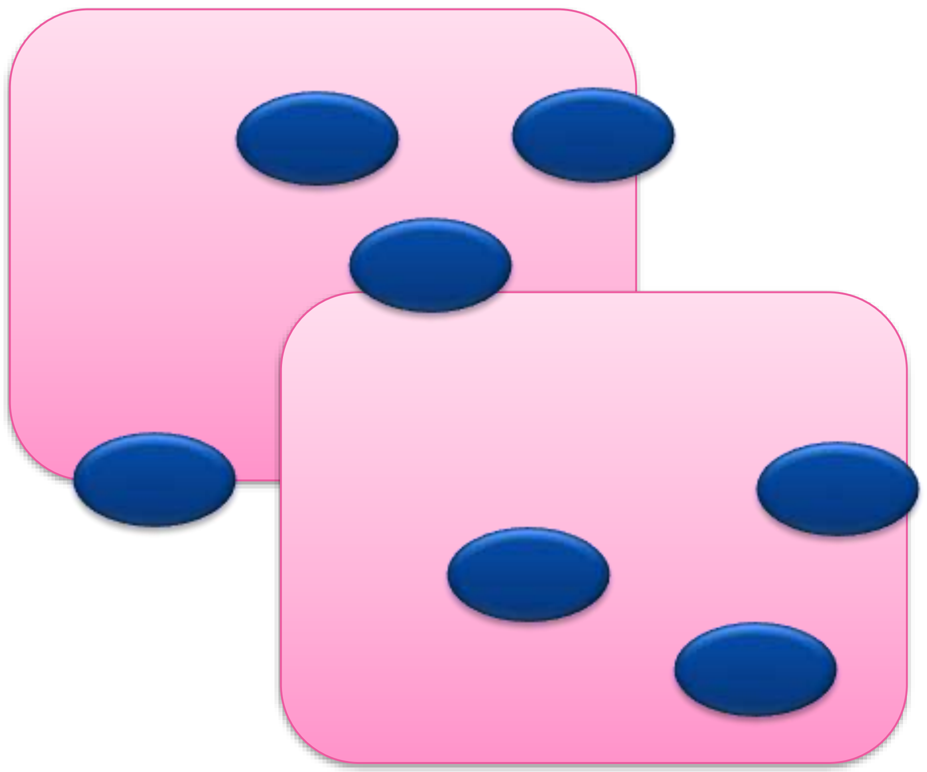


Why need to implement so many IPs? (cont.)

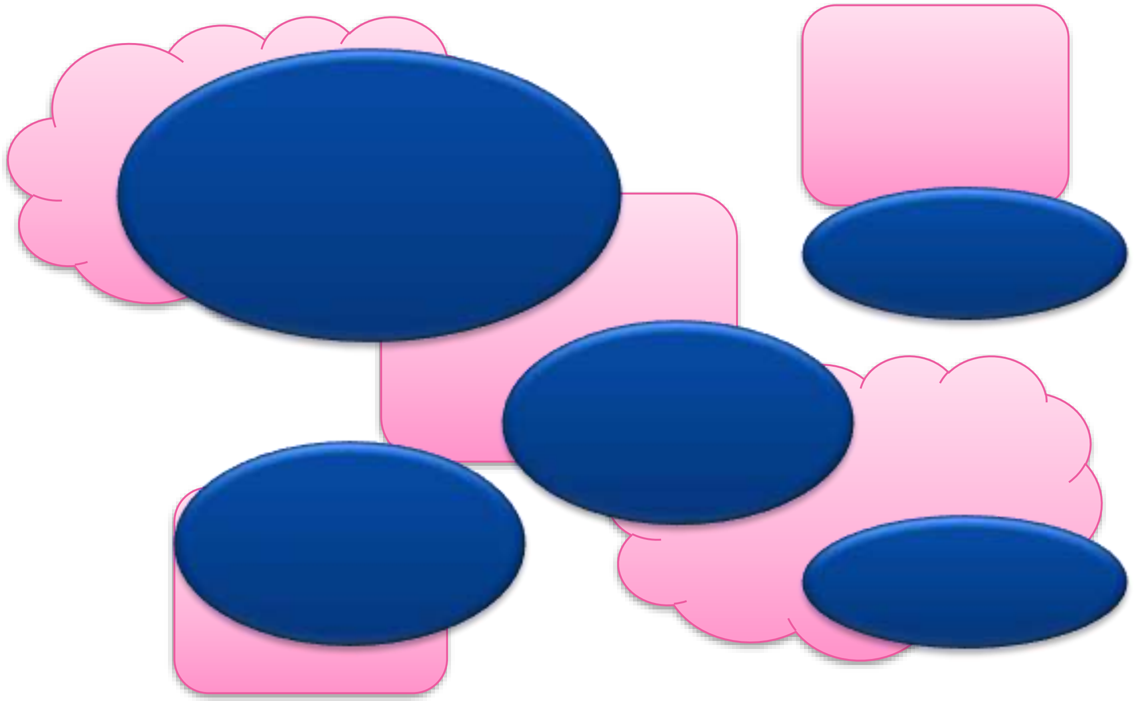
Because of tech. trend, H/W coverage is going smaller and deviates from real requirement

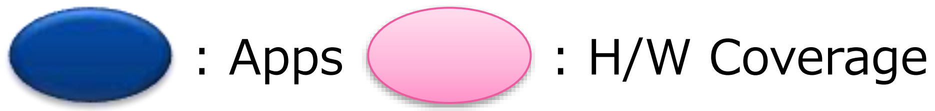
- What are major reasons
  → Difference from the past

H/W and Apps in past – Generic



H/W and Apps Today – Domain Specific?

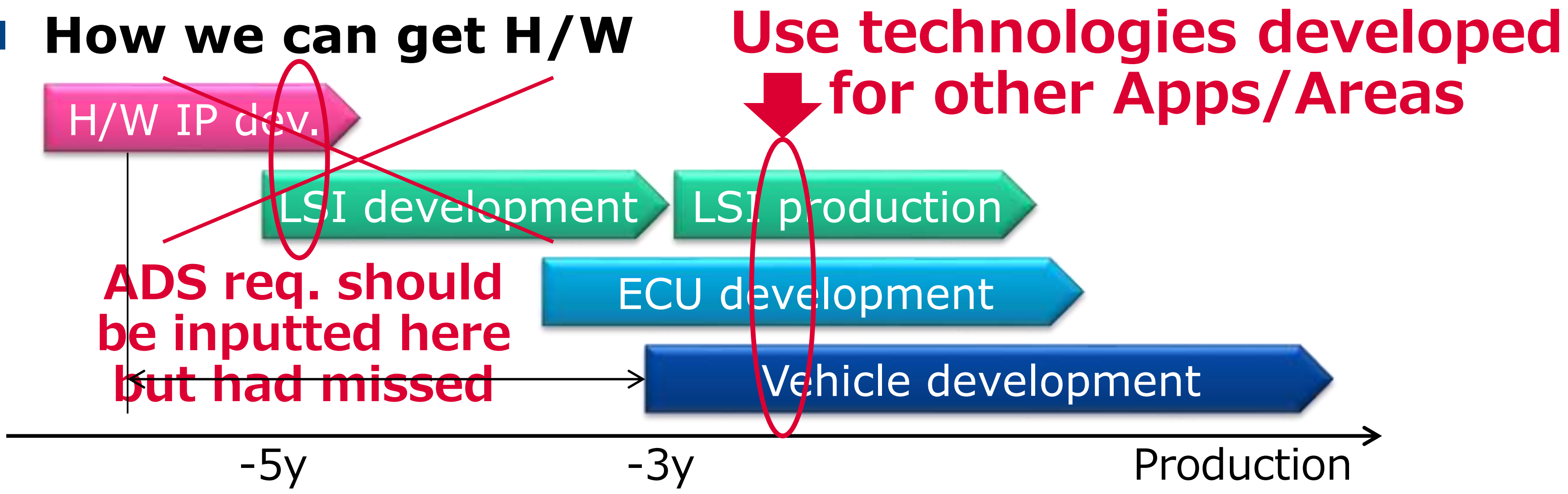




3. Specificity of the AD SoC

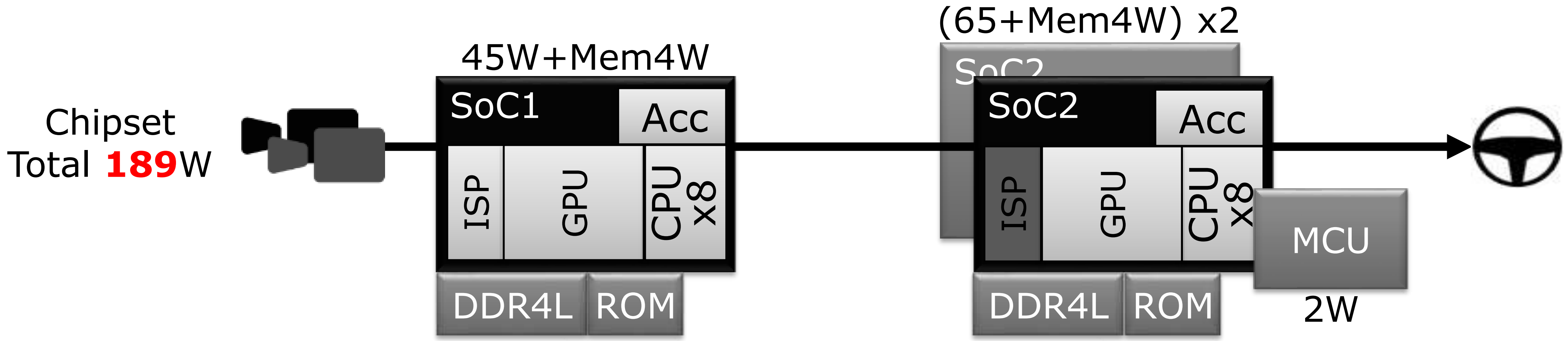
Design time ? LSI vs. System

- **How we can get H/W**



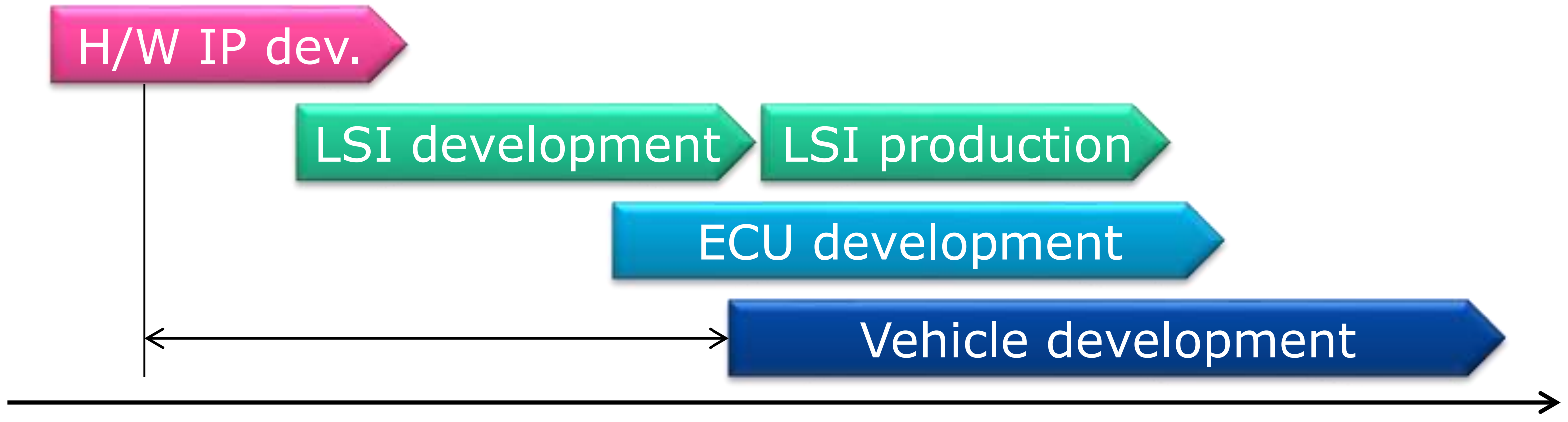
Use technologies developed for other Apps/Areas
H/W IP dev.
LSI development
LSI production
ECU development
Vehicle development
ADS req. should be inputted here but had missed
-5y
-3y
Production

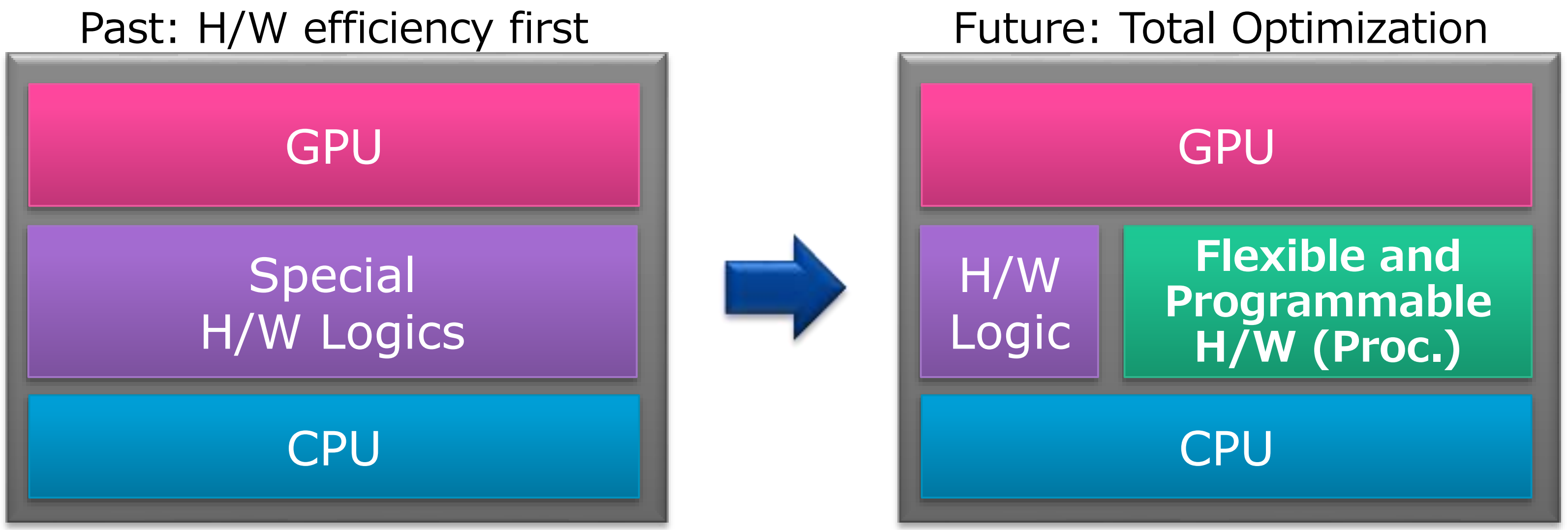
- **Example of ADS chipset using non-Auto based tech.**



What we want?

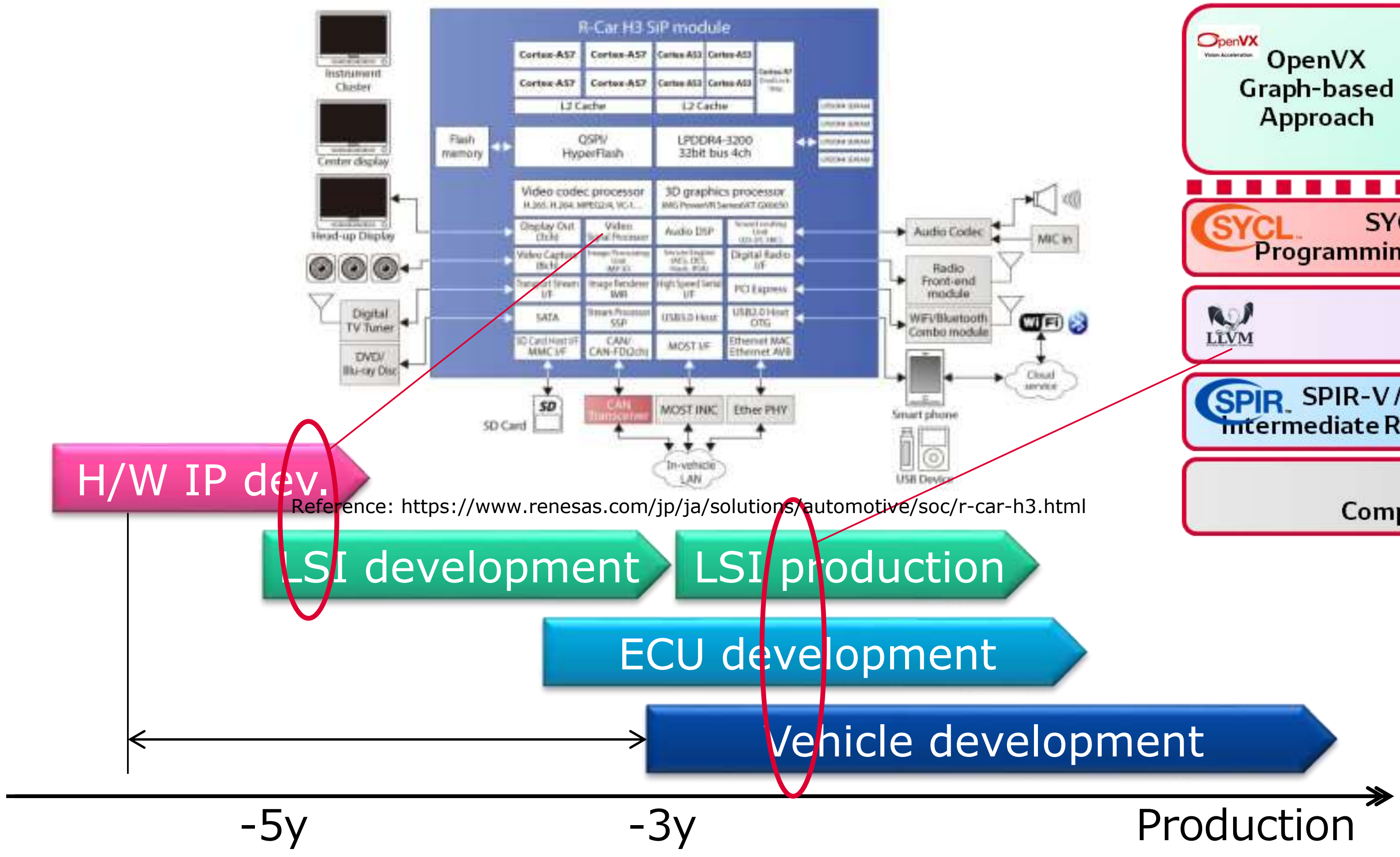
- Time to market
  - LSI dev. should be
    - Start earlier
    - More simpler
  - ADS dev. should be
    - Concentrated investment in strategic func.
      → Define early
    - Use Generic /Flexible H/W for other parts
      → S/W defined



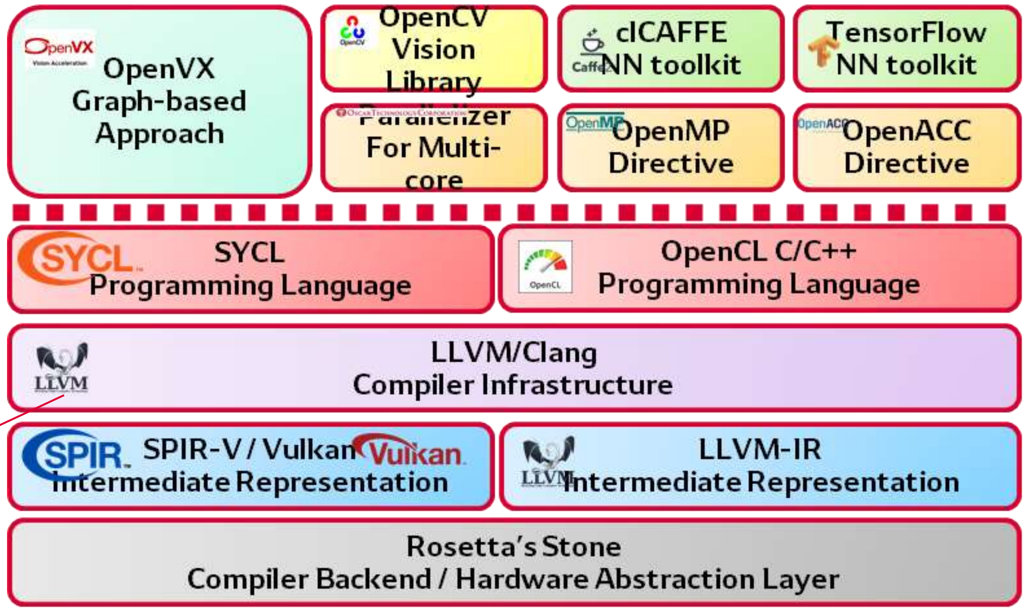

Past: H/W efficiency first
GPU
Special H/W Logics
CPU
Future: Total Optimization
GPU
H/W Logic
Flexible and Programmable H/W (Proc.)
CPU

What we want? (cont.)

- H/W specifications are determined long before start of S/W dev.
  → S/W framework should be defined before H/W design



Reference: https://www.renesas.com/jp/ja/solutions/automotive/soc/r-car-h3.html


OpenVX Graph-based Approach
OpenCV Vision Library
clCAFFE NN toolkit
TensorFlow NN toolkit
Parallelizer For Multi-core
OpenMP Directive
OpenACC Directive
SYCL Programming Language
OpenCL C/C++ Programming Language
LLVM/Clang Compiler Infrastructure
SPIR-V / Vulkan Intermediate Representation
LLVM-IR Intermediate Representation
Rosetta's Stone Compiler Backend / Hardware Abstraction Layer

Issues in AD SoC

- Difficult to find good balance point of trade-offs
  - Higher Performance is always needed
    → but Power consumption / Efficiency is also important
  - Efficiency is still important
    → but Flexibility of H/W is also needed to maximize System or S/W value by applying latest market requirements

    → Fixed function logics are (sometimes) not fit to SoC design time or less efficient to implement many parameters to support variation of Apps./Algorithms
  - Transistors become cheaper
    → but complexity from so many #of Transistors exceeds limit of Design/Implementation/Verification capability

➔ We need to challenge to get reasonable answer for these issues!

4. Architecture Challenge for Future Embedded Compute SoC

Requirements, Limitations and Architecture Challenges

- Conventional way will not work for future Embedded Compute

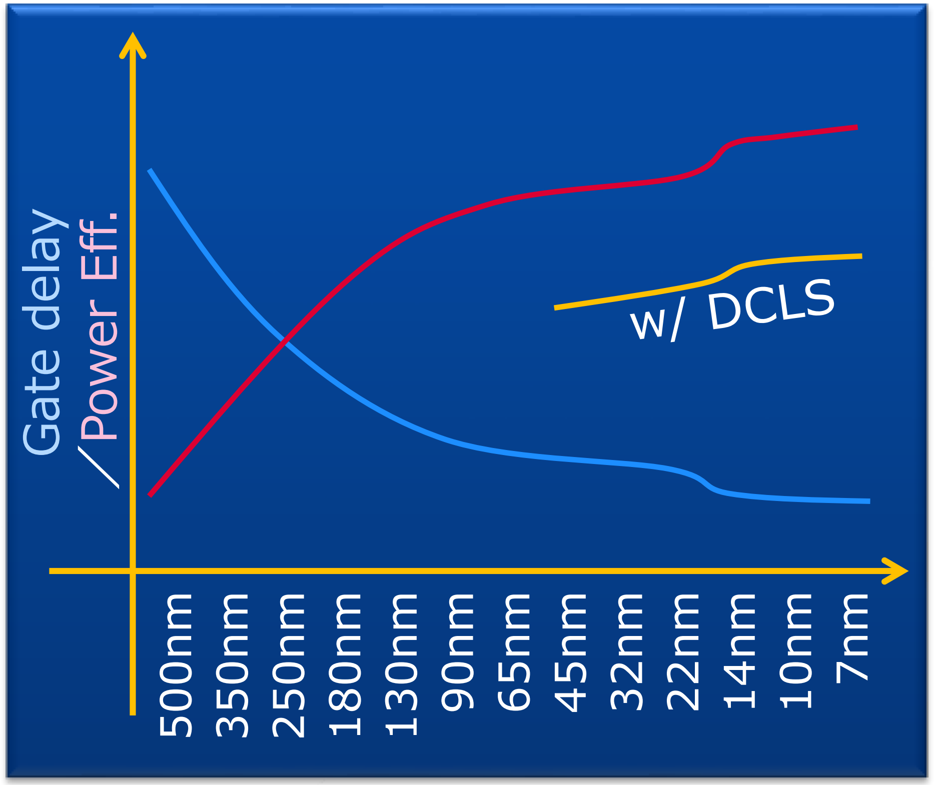
Requirements	Limitations	Solution	Challenge
Performance	Transistor / wire delay is no longer reduced	Parallel processing	How to cover dispersed characteristics of compute
Low Power / Efficiency	Hard-wired logics will not fit to design time	New generation processors	High-dense and flexible compute
Productivity / Design Quality	Circuit complexity is too high	Repeated design	Simple and flexible compute

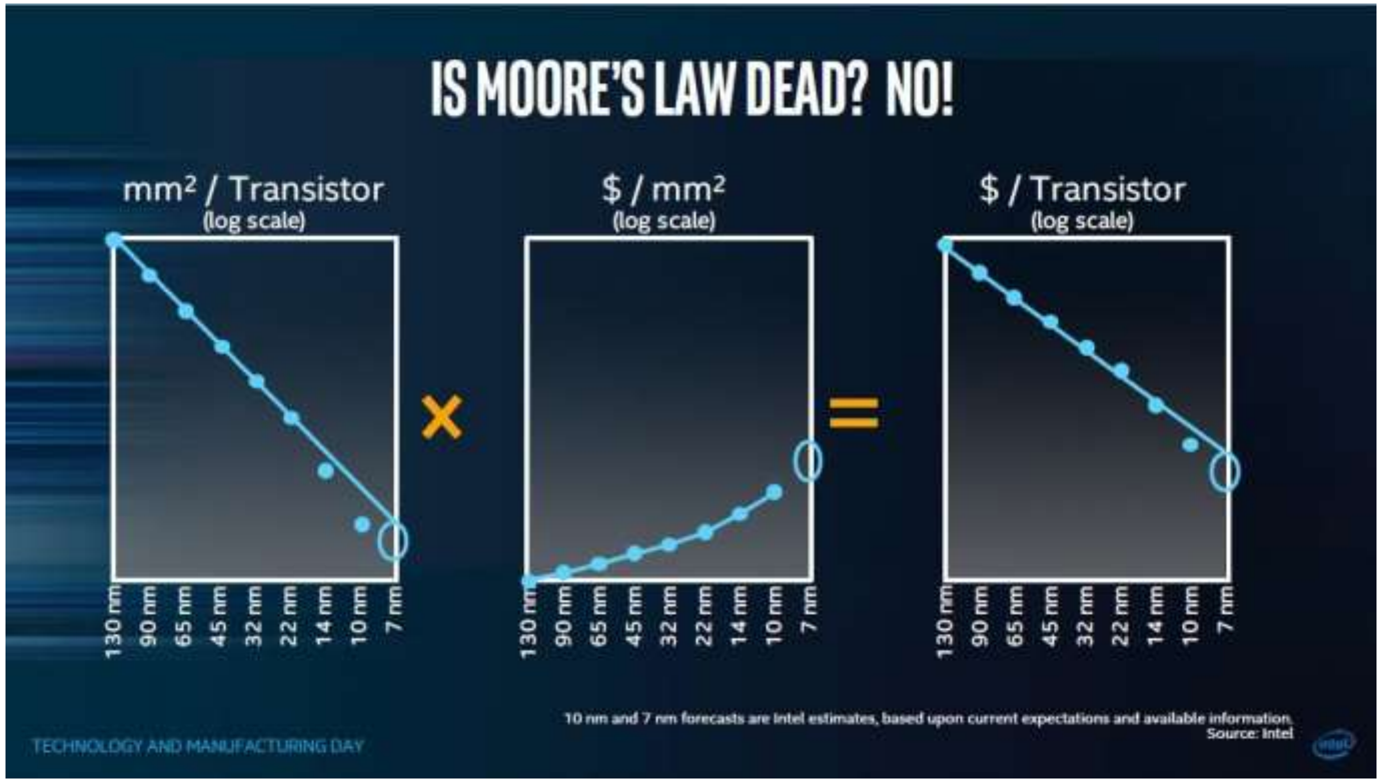
Conventional way will not work for future Embedded Compute!

Performance – Process technology trend

- H/W Trends
  - No more perf. by process shrink
  - Power consumption is also
  - → What is good trade-off factor?

Transistors become cheaper
→ Implement individual functions?
→ Or built-up by regular programmable functions (e.g. processors)

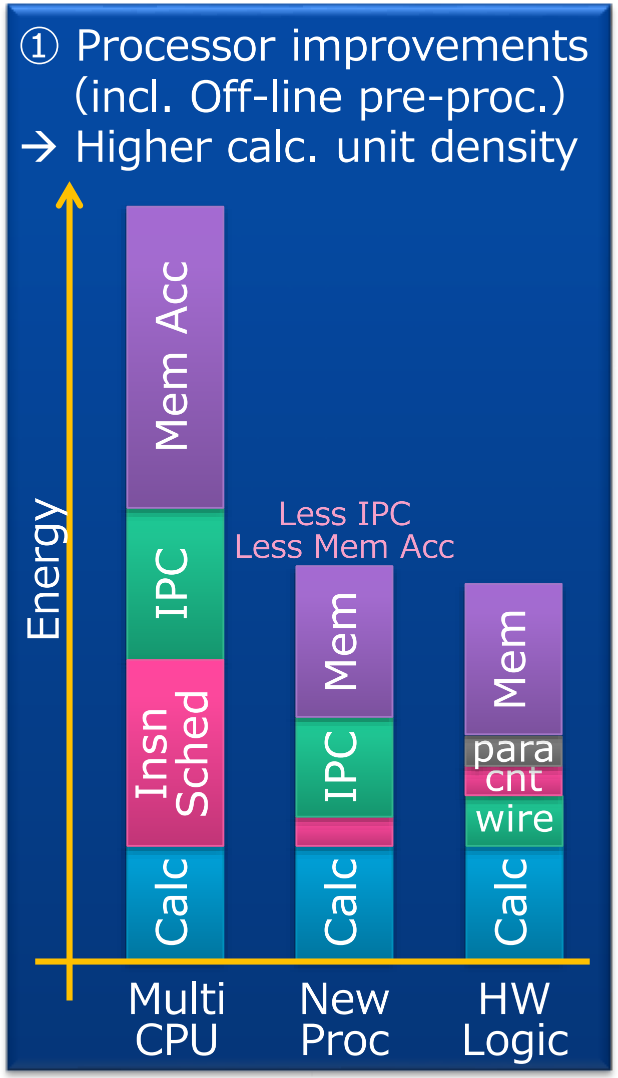


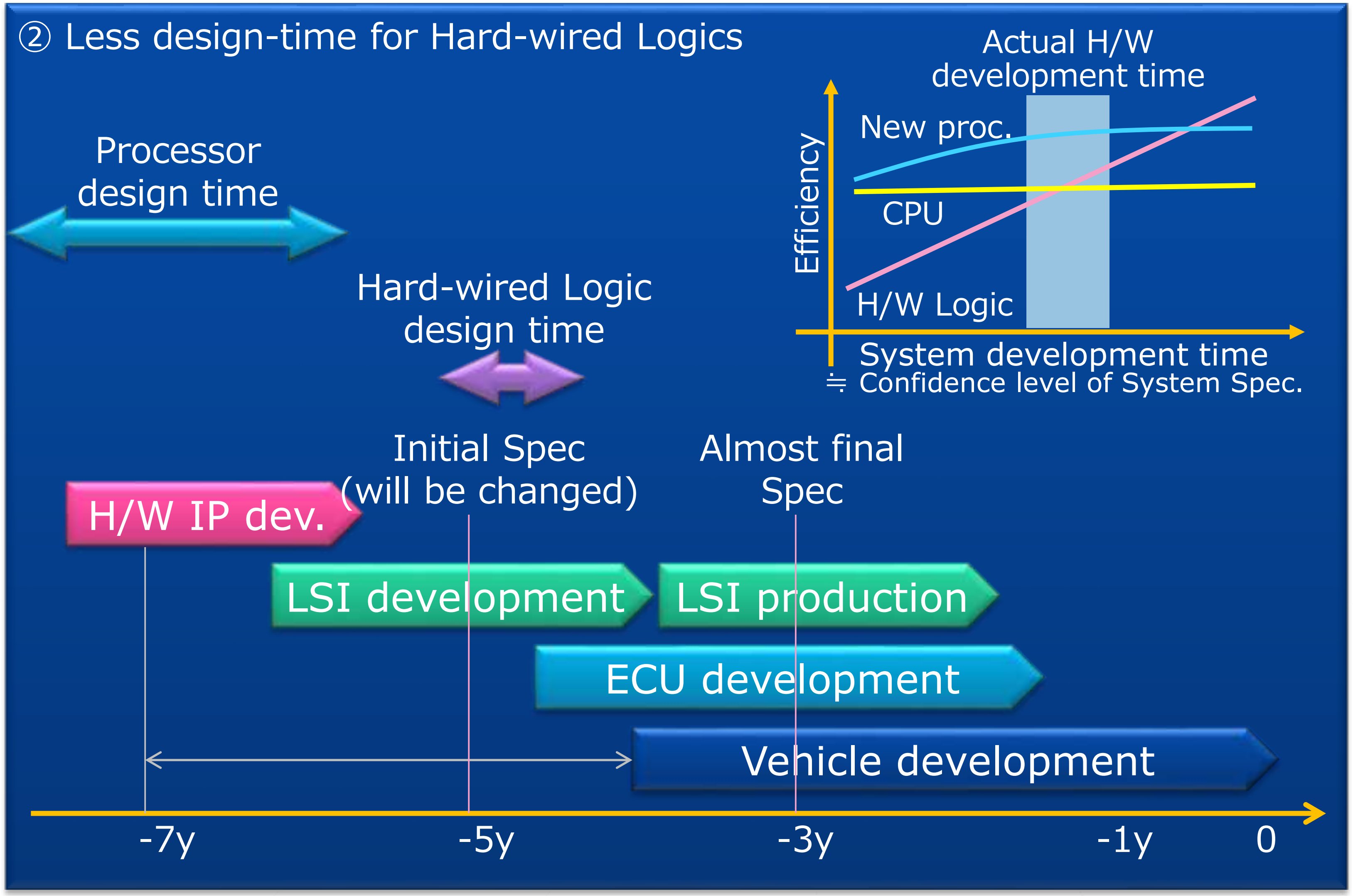


Reference: Intel Technology and Manufacturing Day

Efficiency – Hard wired Logic vs. Processor

- End of Hard-wired Logics?



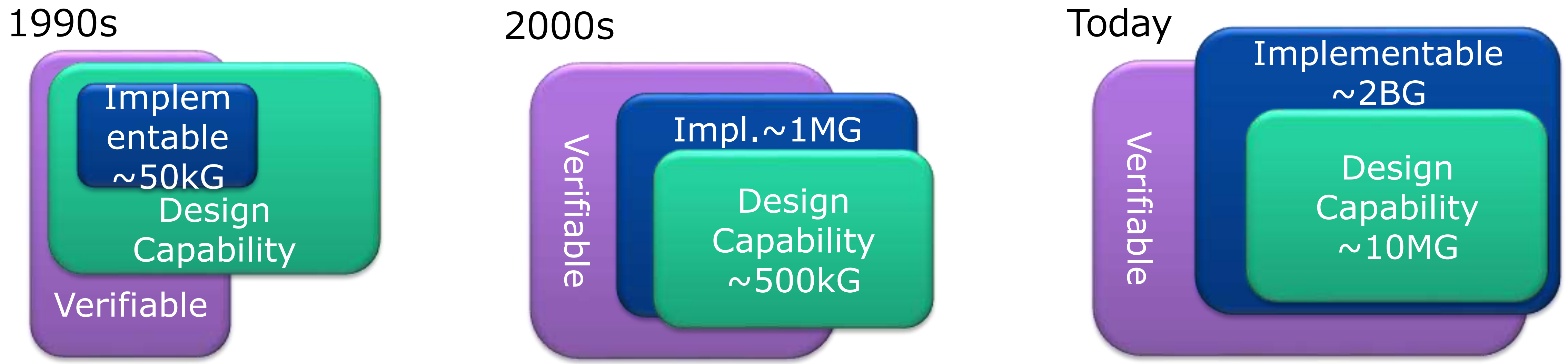


Efficiency – Hard wired Logic vs. Processor (cont.)

- New generation processors can now over perform Hard-wired logics!

Productivity – Design bottleneck

- Limit of design and verification space (within limited time/resources).



- In ADS, the correct answer changes with the times
  - Industry standard (ISO26262 etc..)
  - Emergence of new dangers (DoS, Zero-day Attack···)
  - Common sense (Acceptable risks···)
  - Individual difference (Customization, Personalization···)

  ···

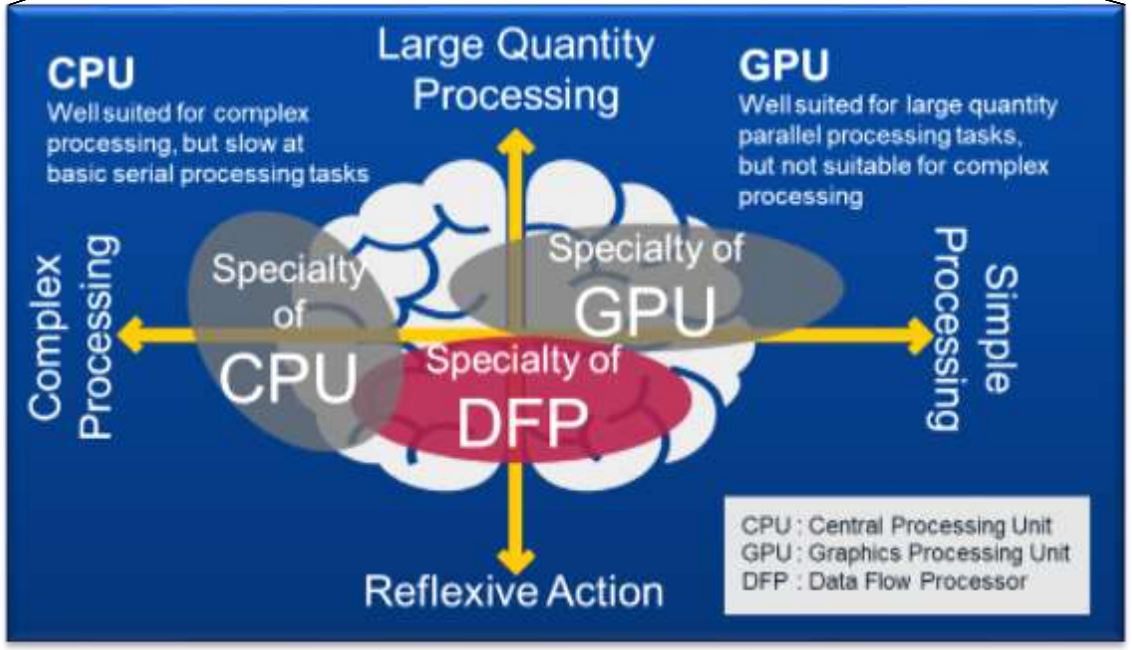
➔ Flexibility of SoC Arch. for Updates is also needed

Computer vs. Embedded

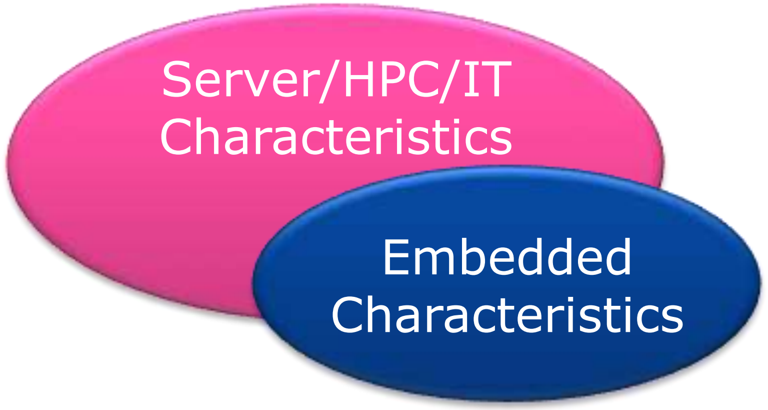
- Why Server/HPC processors are sometimes not fit to Embedded?

Architecture	Performance	Efficiency	Quality	Design	Arch. Life
Server/HPC…	Throughput	Perf → Power	Uptime	TAT → Opt.	1 to 3years w/ Updates
Embedded	Latency	Power → Perf	Defect rate	Opt. → TAT	3 to 9years w/ Ltd. OTA

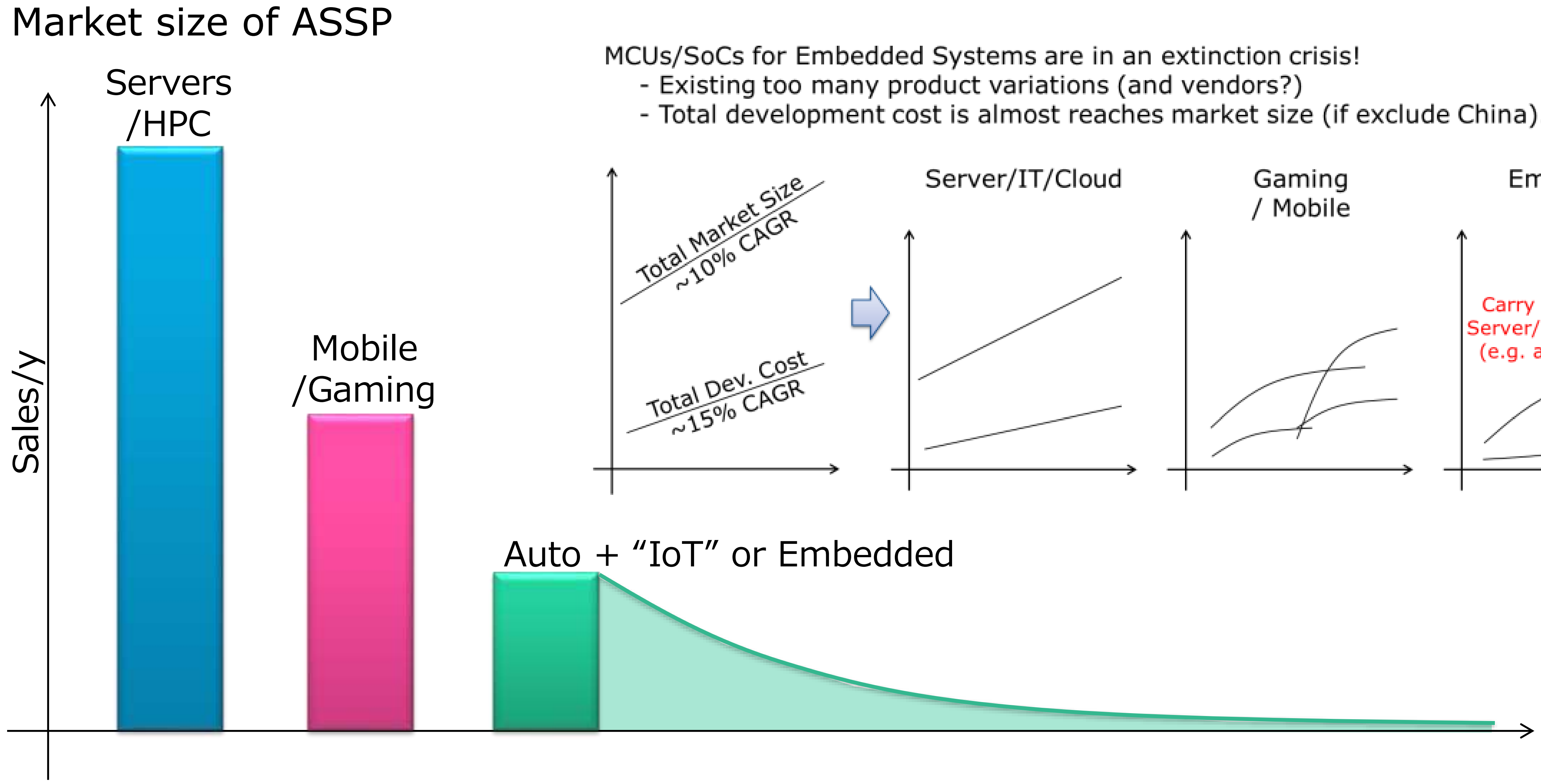
Processer Characteristics Example (from NSITEXE DFP concept)


CPU
Well suited for complex processing, but slow at basic serial processing tasks
Large Quantity Processing
GPU
Well suited for large quantity parallel processing tasks, but not suitable for complex processing
Complex Processing
Simple Processing
Specialty of CPU
Specialty of GPU
Specialty of DFP
Reflexive Action
CPU : Central Processing Unit
GPU : Graphics Processing Unit
DFP : Data Flow Processor

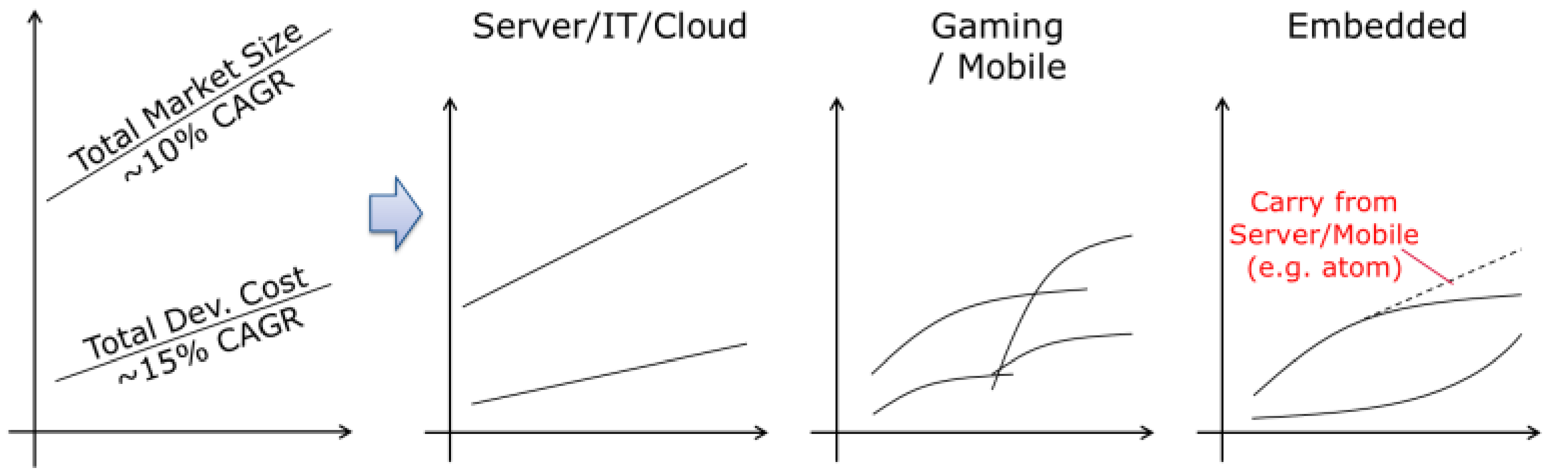
→ Covering complete Embedded area is too wide for Today's technologies/processors.



Can Embedded ASSP survive?

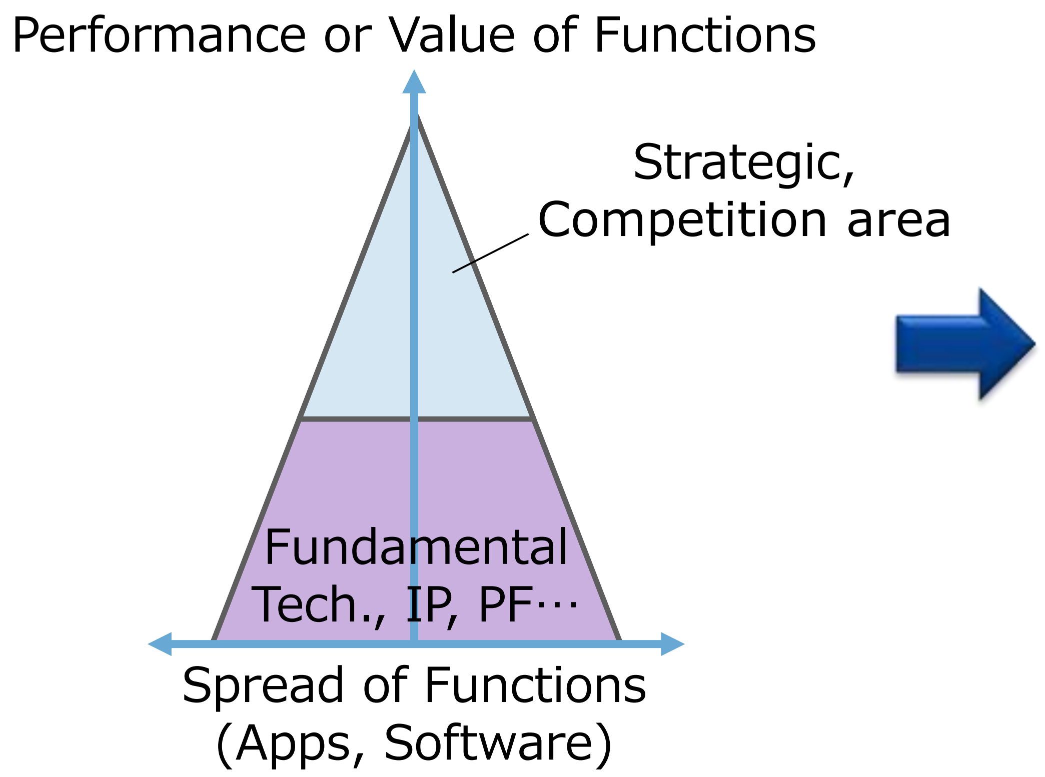


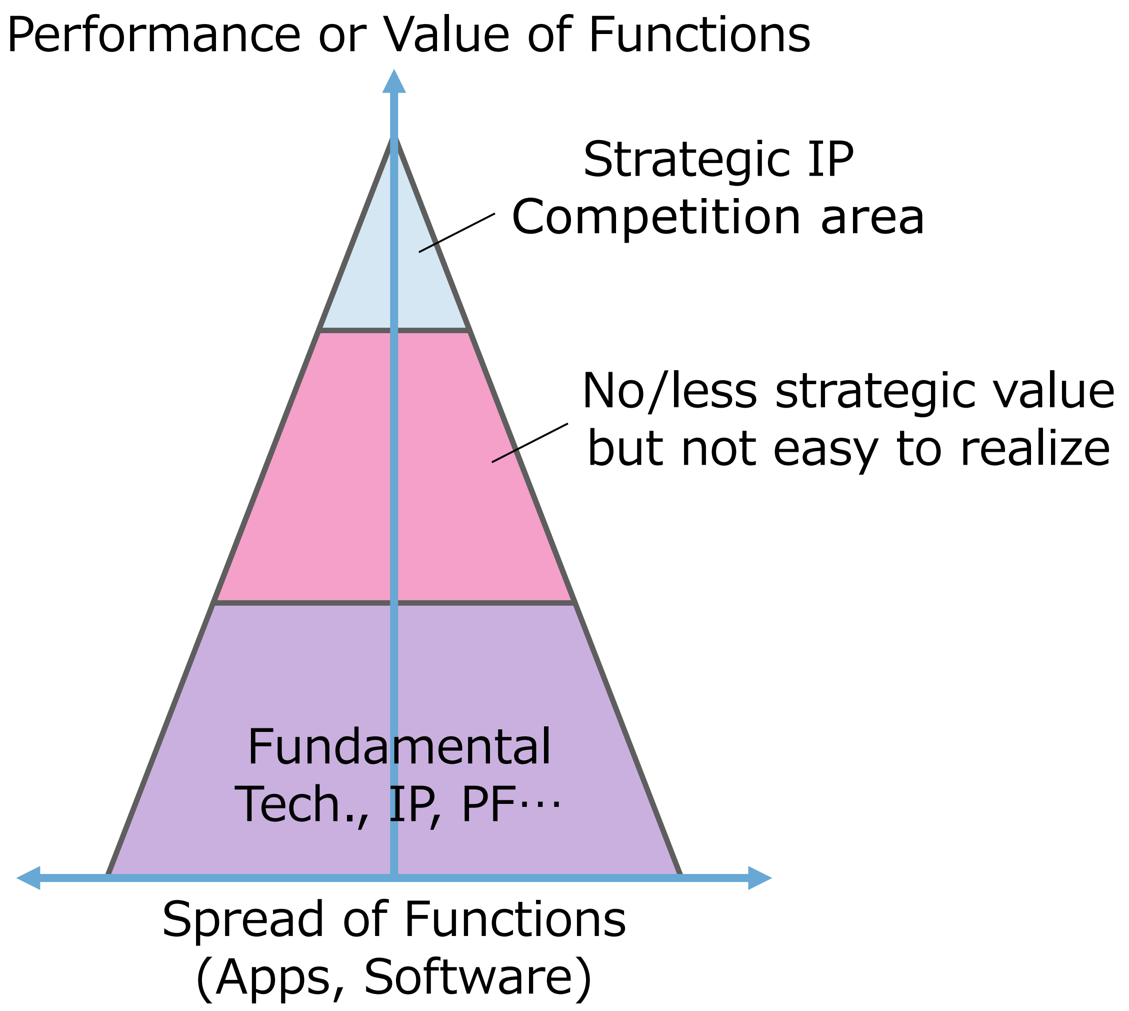
MCUs/SoCs for Embedded Systems are in an extinction crisis!
- Existing too many product variations (and vendors?)
- Total development cost is almost reaches market size (if exclude China).


Total Market Size ~10% CAGR
Total Dev. Cost ~15% CAGR
Server/IT/Cloud
Gaming / Mobile
Embedded
Carry from Server/Mobile (e.g. atom)

Why old ASSPs were gone?

Single "peaky" value is no longer exist
→ Complex and Huge scale design
= "Middle" area is not easy to implement
even if it does not have high value

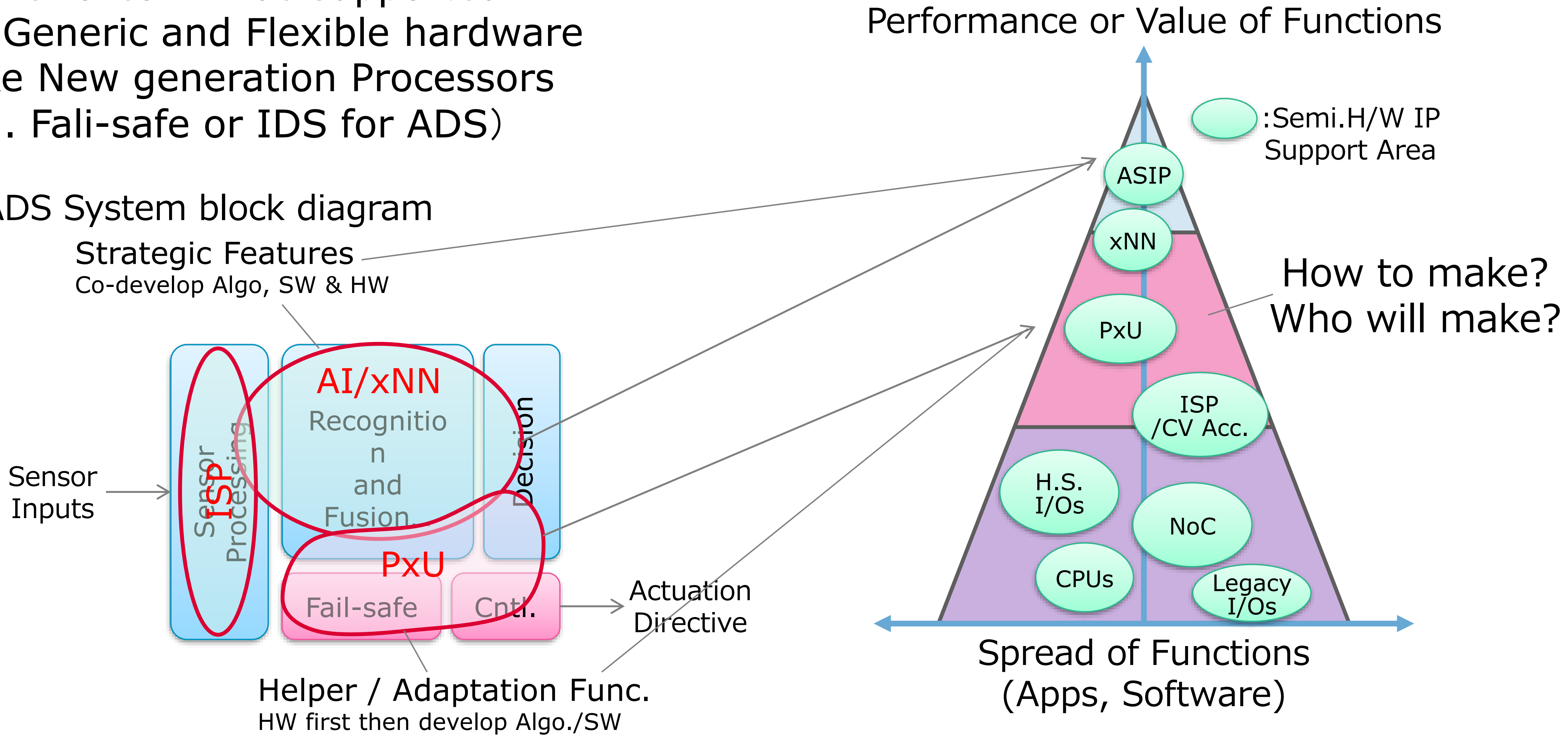




Example: IP map in ADS

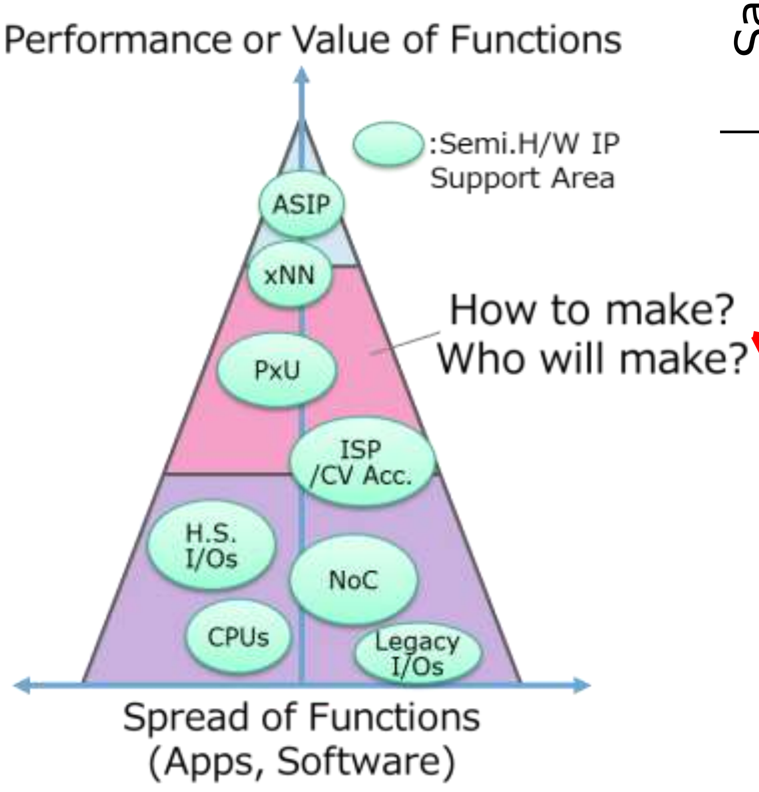
“Middle” area will be supported
by Generic and Flexible hardware
like New generation Processors
（e.g. Fali-safe or IDS for ADS）

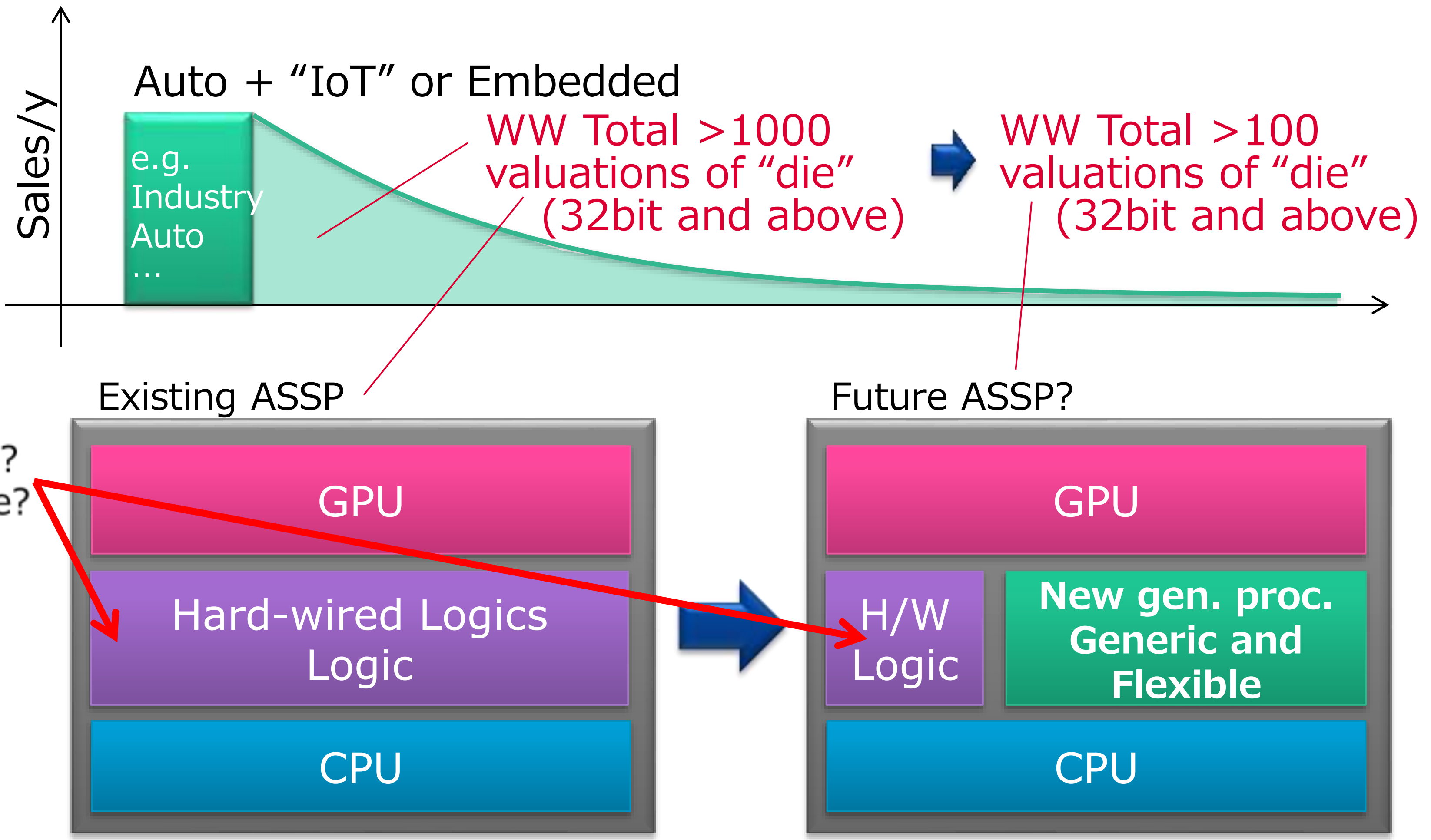
ADS System block diagram



Architecture Challenge for Embedded SoCs

Need "embedded-generic" SoCs
→ Flexible, Scalable and Generic to cover major "Middle" part Apps.


Performance or Value of Functions
:Semi.H/W IP Support Area
ASIP
xNN
PxU
ISP /CV Acc.
H.S. I/Os
NoC
CPUs
Legacy I/Os
How to make?
Who will make?
Spread of Functions (Apps, Software)


Sales/y
Auto + "IoT" or Embedded
e.g. Industry Auto …
WW Total >1000 valuations of "die" (32bit and above)
WW Total >100 valuations of "die" (32bit and above)
Existing ASSP
GPU
Hard-wired Logics Logic
CPU
Future ASSP?
GPU
H/W Logic
New gen. proc. Generic and Flexible
CPU

Conclusion

Conclusion

- ADS: Still no clear common answer for optimum system
  → ADS is very complex and very difficult to describe "correct" functionality of System, so at least these several years, many people will design and implement it very different way. It requires different types of SoCs.

- SoC, ASSP: More generic, more flexible
  → Because of increasing of variations of Embedded systems, SoCs or ASSPs should have more flexibility and wide coverage of Apps to align required time of System design.

- LSI Design: What is critical constraints for design?
  → Need to reduce complexity to use #of Transistors exist

Thanks!

NSI-TEXE