

Embedded Networks

Finnish-Russian University Cooperation Program in
Telecommunications (FRUCT) seminar
Turku, Finland, Nov-07

Michel Gillet and Sergey Balandin

michel.gillet@nokia.com, sergey.balandin@nokia.com

CTC Computing Structure, Architecture Solutions,
Nokia Research Center

Outline

- Once upon a time ...
 - Initial problem and requirements
 - Change over time of scope and requirements
 - Current status and vision
- Embedded networks
- Our research
- NoC vs. chip-to-chip
- Future work and research topics
- Conclusions

Once upon a time



NOKIA

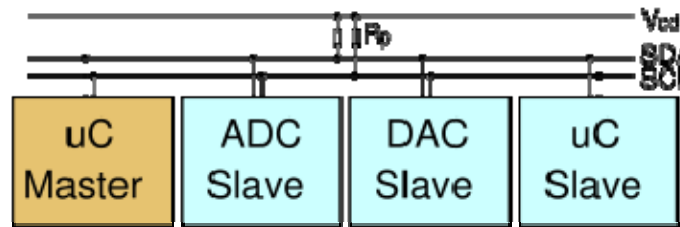
Company Confidential

Once upon a time ...



- Like every good story, this one starts like this ...
- “Once upon a time, there were 2 engineers asking a simple question: how can we add easily functionality to an existing mobile device?”
- This was 7 years ago
- At that time, the only available technologies for an extension bus in the mobile world were:

- I²C
- SPI
- Etc.



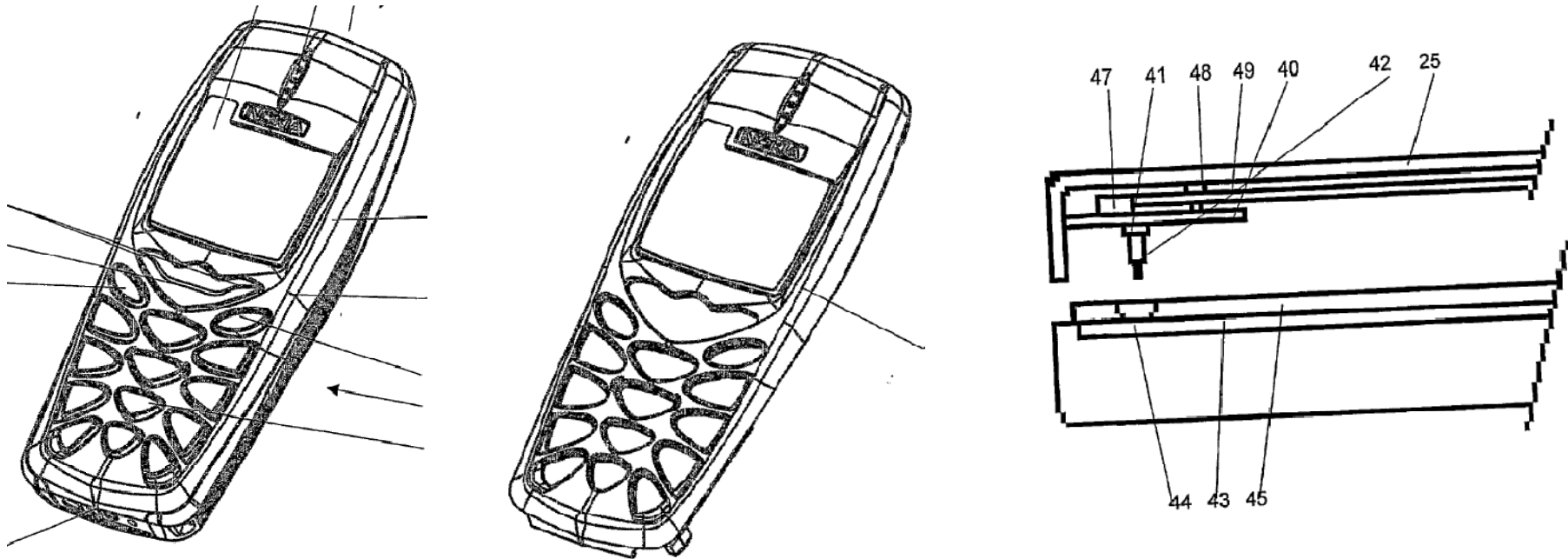
- The main issue then was the very low bandwidth provided

Context

- 7 years ago, in the middle of the “dot-com bubble”, a technical problem was hindering
 - The fast introduction of new technologies in products
 - The R&D, interfacing and prototyping of new ideas
- The issue was made more severe by the absence of so called “extension bus”, like ISA or PCI in the PC industry
- The development cycle of new chipsets allowing the introduction of new IP blocks was pretty rigid and having its own roadmaps and agenda
- All in all, the mobile architecture was very static and monolithic

The concept of “functional cover”

- The idea was to use the concept of “functional covers”
- Since every mobile device has at least one cover, if we could have a galvanic connection between the cover and the terminal, we could add functionality by changing the cover



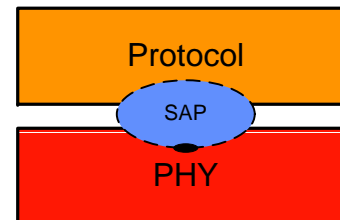
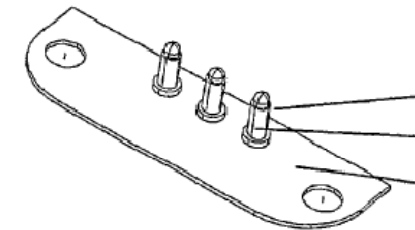
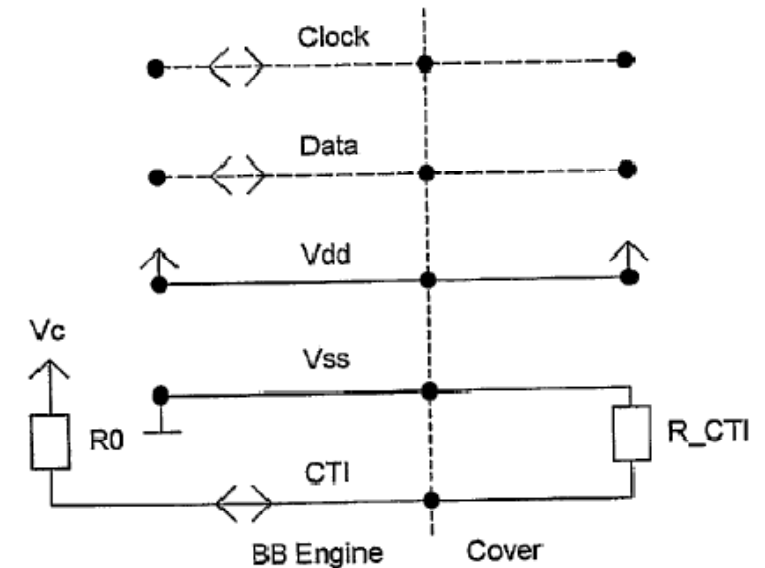
Company Confidential

Taken from a patent, Publication Number WO/2003/079653

Original functional cover requirements

Taken from a patent, Publication Number WO/2003/079653

- Because of the cost of the mechanics and connectors, the number of pins acceptable for one cover was to 4 or 5
 - 2 pins for data
 - 2 pins for power: VCC and GND
 - 1 optional pins for different purposes
- I²C and SPI allow few Kbit to Mbit/s
- The goal was to reach few 100 Mbit/s
- The PHY was seen as the biggest issue to solve to reach these speeds
- The system was divided in PHY + protocol



Company Confidential

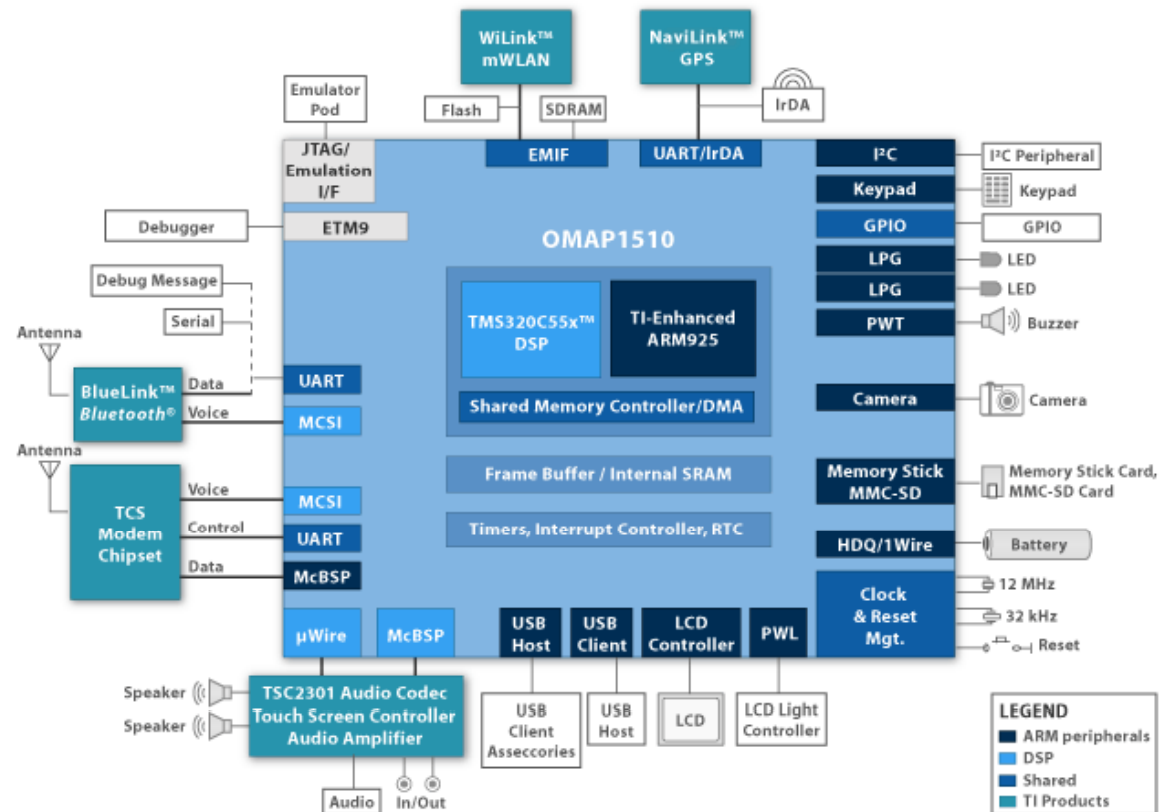
Concept buy-in

- After few months and when more managers were in the loop, a new technical solution to support only 1 functional cover was deemed as not satisfying. Up to 4 covers should be supported
- Few 100 Mbit/s was seen as too conservative and the bar was raised to 1 Gbit/s
- The great idea was then to define a bus technology
 - Being multi-points
 - Supporting Gbit/s speed
 - Maximum 5 pins, better for including power



Burden of interfaces management

- Up to now in the mobile industry, there is no solution equivalent to PCI or PCI-Express
- The consequence is that every new application will have its own interface
- The number of different interfaces is growing continuously and their management costs are very high



Taken from www.ti.com

Monolithic architecture

- After defining the concept of “functional covers”
- After making the parallel with the ISA or PCI bus
- After seeing the burden of supporting too many interfaces
- It was clear that the main issue was more in term of monolithic closed architecture
- Many new use cases where added
 - Display, cameras, mass storage, wireless radio/modem, etc.
- The direction was changed from “functional covers” to a “display camera and options bus” with the goal to support
 - A more open host/peripheral architecture (PCI-like)
 - A more distributed architecture (MPI-like, but much simpler)

Collaboration

- When we started to work on embedded network, the level of secrecy of the work was such that we were not allowed to talk about it to any colleague not directly involved in the work
- But if one compare the radical change which would arise in term of architecture if embedded networks are used in mobile devices, it seems clear that even Nokia is too small on its own to impose such a drastic change
- From an architecture point of view and compared to the PC world, it's like we would jump from the first IBM PCs of the 80's to PCI-Express based PCs, and 2x or 4x faster
- So we had to relax the level of secrecy and one key partner joined the discussion

Standardization

- But at some point, one partner was not enough and it became clear that to be successful in the deployment of such ideas, a broad support in the mobile industry was needed
- There was no other choice than to go for standardization
- The working group of UniPro (UNified PROtocol) was then created in MIPI



Mobile Industry Processor Interface alliance (MIPI)

- MIPI has been established by Nokia, ARM and TI in 2004, www.mipi.org
- Alliance is targeted in creating new standards for the mobile industry with lifetime expectancy of 10+ years
- The original focus was solely on processor to peripheral interfaces internal to the mobile device, but later it has been extended to networking architecture and external interfaces
- At the moment MIPI consists of 150 member companies (Nov-2007)
- The access to produced knowledge is restricted, so based on their financial input the member companies are divided into the following groups:
 - 7 Board Members – driving development of MIPI standards
 - 56 Contributors – participating in the specifications development
 - 87 Adopters – get access to the specifications only after board approval
- MIPI consists of the following Working Groups:
 - UniPro WG, Camera WG, Display WG, Test&Debug WG, Software WG, etc.
- Due to its focus area, UniPro is the key and most critical WG in MIPI

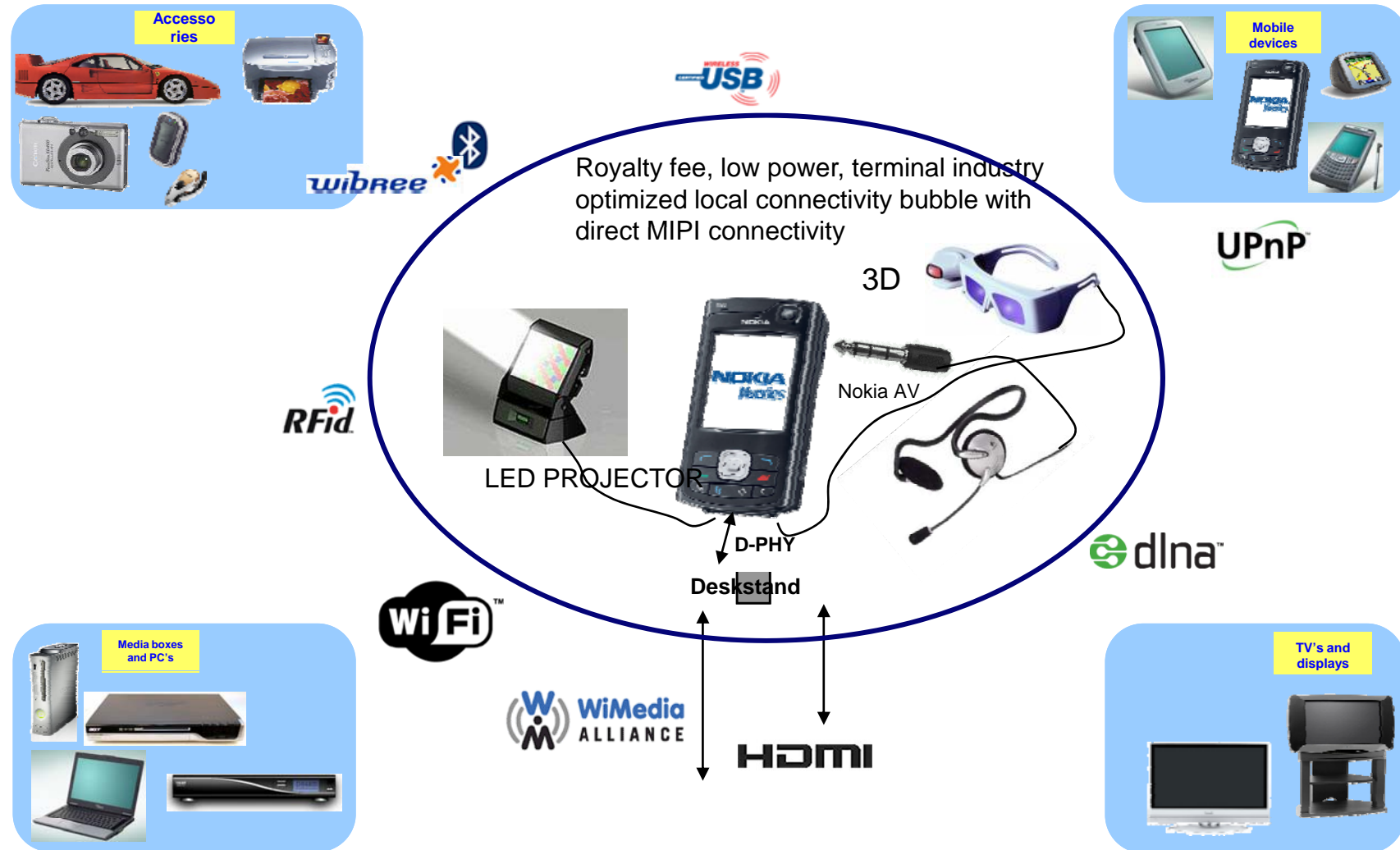
Local Connectivity landscape, Today

Multitude of standards to be made simple-to-use and to provide interoperability



Local Connectivity landscape, Tomorrow?

Multitude of standards to be made simple-to-use and to provide interoperability



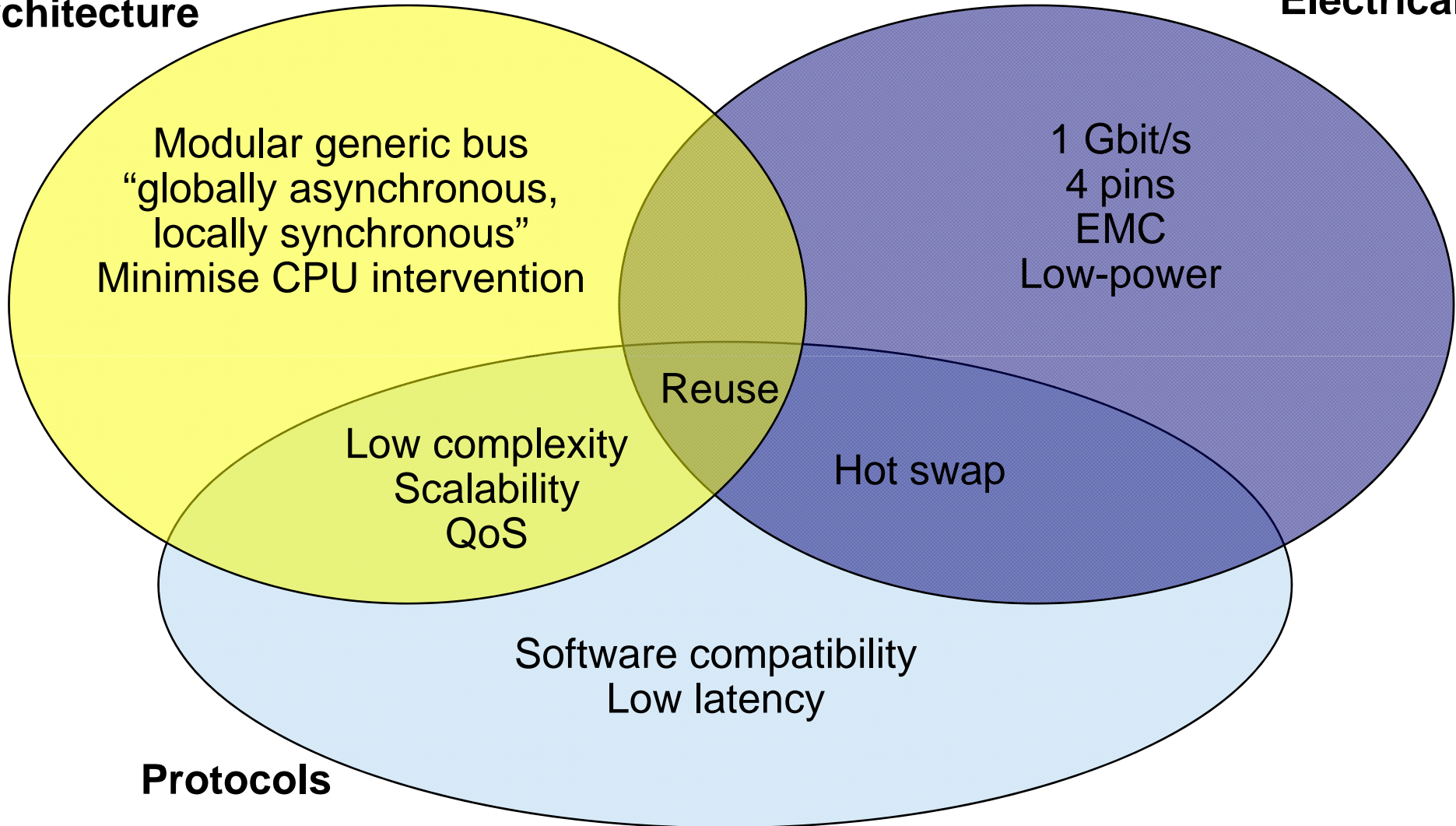
Company Confidential



Requirements

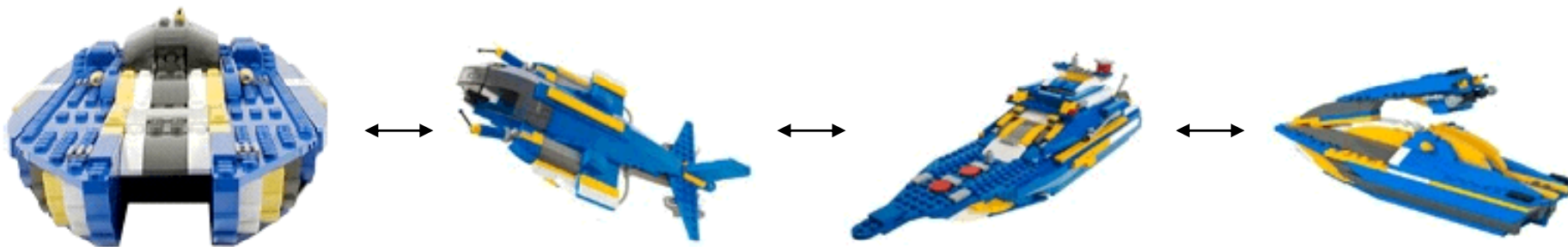
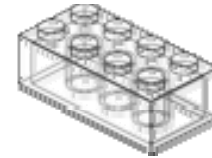
Architecture

Electrical



Our approach: Let's play LEGO

- What are the LEGO blocks?
 - Engine, modem, camera, display, etc.
- Each block has a standard attachment mechanism
 - Engine has few *sockets* where *anything* can be connected
- With the same basic blocks, many configurations or topologies can be made easily
- Needs a skeleton block to attach all others
 - Network of blocks



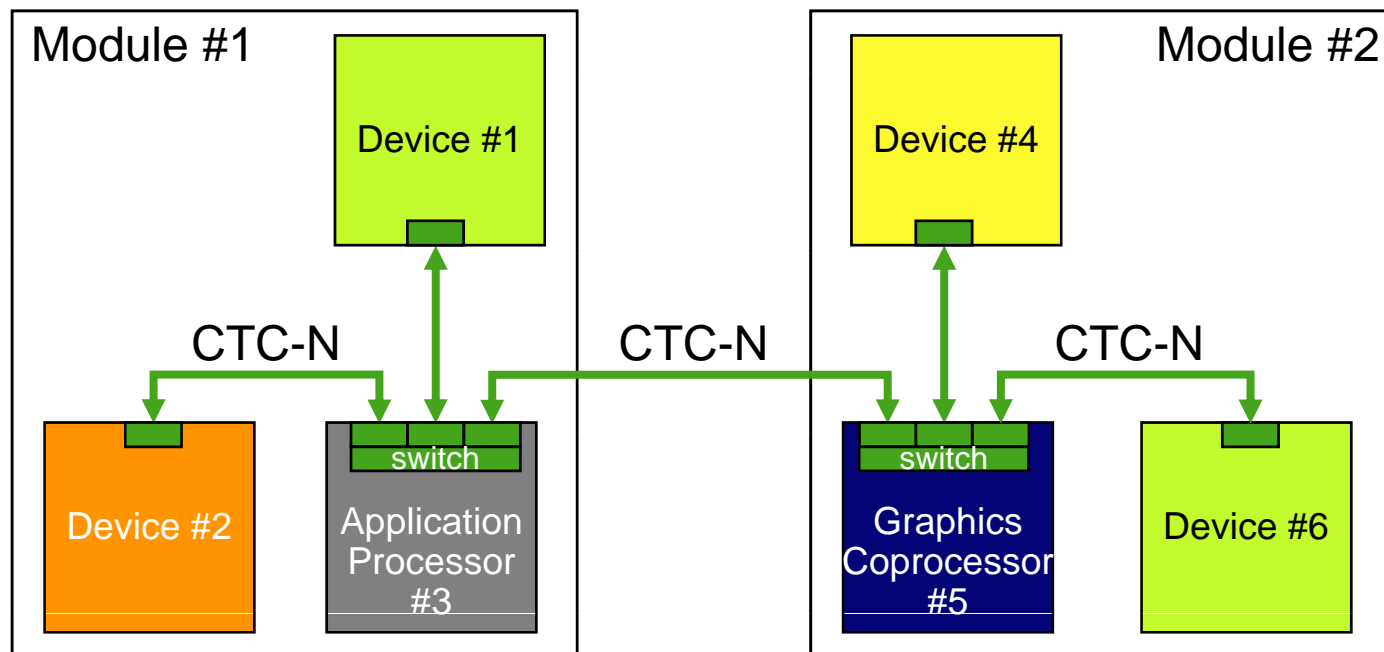
Embedded networks

NOKIA

Company Confidential

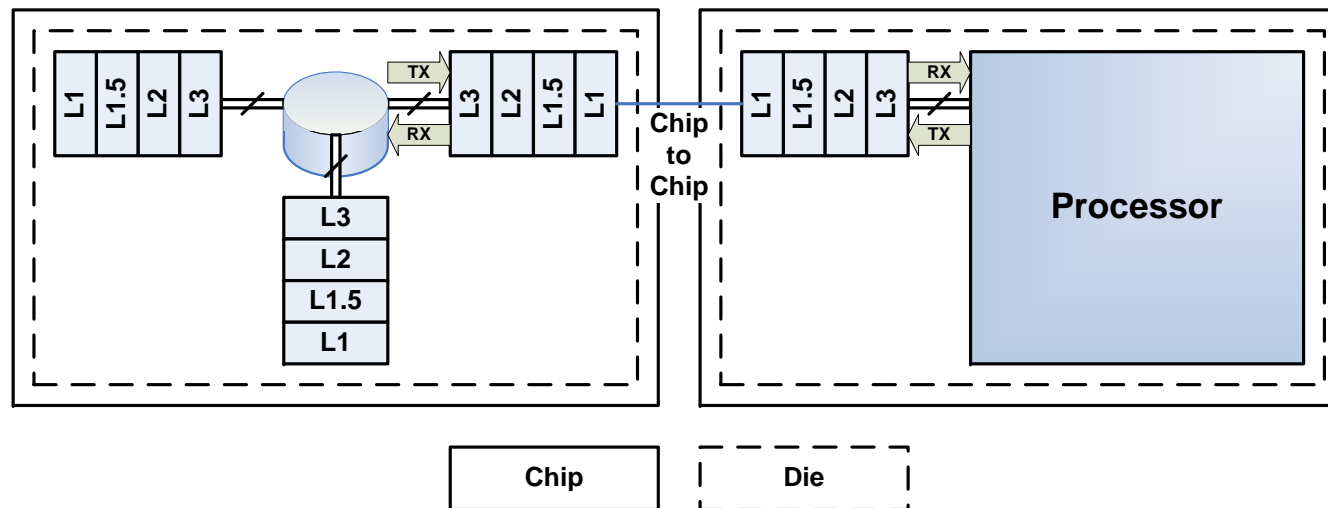
Embedded networks

- From the simple “Let’s play Lego” motto and the mobile device context, emphasis was mostly put on chip-to-chip networks
- On possible use case is shown below



Chip-to-chip embedded networks

- This is the first use case to look at and the most fundamental, since it allows to break the monolithic architecture of mobile devices
- It's the first step or milestone towards modular, extensible and distributed architecture



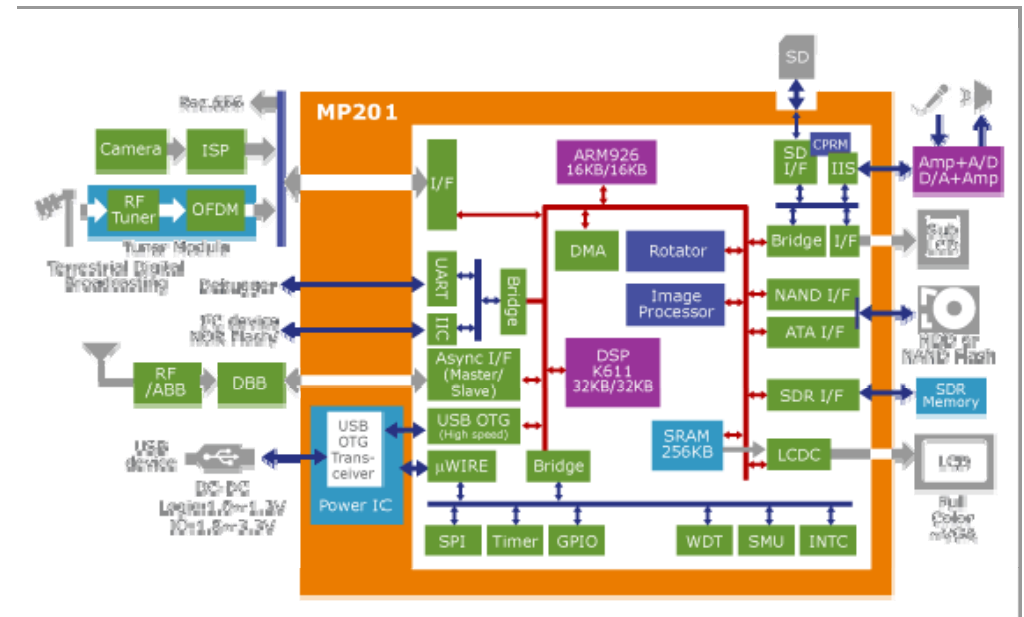
Chip-to-chip: cost factors

- The driving factors for a successful industrialization of such technology in the mobile device domain are cost and power consumption obviously
- What are the main cost factors?
 - Number of pins needed per link
 - Mechanical and electrical implications (connectors, cabling, flexes, shielding, etc.)
 - Overall integration costs (including R&D time spent to test new technologies, etc.)
 - Number of links
- Not so critical factors
 - Silicon area of the PHY



Chip-to-chip: pin count

- Why is the pin count so critical?
- As an example, let's take the MP201 from NEC, which is packaged in a 529-pin plastic FBGA, 14 mm, 0.5 mm pitch
- High costs are induced by
 - Packaging and the package itself
 - Mounting and assembly
 - PWB manufacturing
 - Many metal layers needed
 - Soldering reliability and durability
 - Resistance to drop tests
 - Rigidity of the assembly
 - Etc.



Chip-to-chip: high-speed serial links

- To succeed in reducing the pin count and lowering the power consumption, high-speed serial links seem very attractive
- High-speed serial links are defined by their differential signalling and the fact that data is send a bit at a time
- Roughly, there are 2 main class of high-speed serial links using
 - Source synchronous clocking
 - Embedded clocking

Chip-to-chip: source synchronous clocking

- Advantages
 - is simple and thus has a low silicon area
 - power efficient compared to CMOS (pJ/bit)
 - no need of special synchronization mechanism in the RX side
 - fast transition between off and on state
 - simple to increase throughput by adding more data lanes
- Disadvantages
 - not optimum in term of number of pins (minimum 4 per direction)
 - Not as power efficient as embedded clocking in terms of pJ/bit
 - Doesn't scale well in term of bandwidth of a single lane (getting harder above 1 Gbit/s)
 - Harsh constrains on wiring to avoid skew between clock and data lanes

Chip-to-chip: embedded clocking

- Advantages

- optimum in term of number of pins (2 per direction)
- power efficient all other solutions (pJ/bit)
- scales well in term of bandwidth of a single lane (up to 5 or 6 Gbit/s)
- no need to add more data lanes in the foreseeable future

- Disadvantages

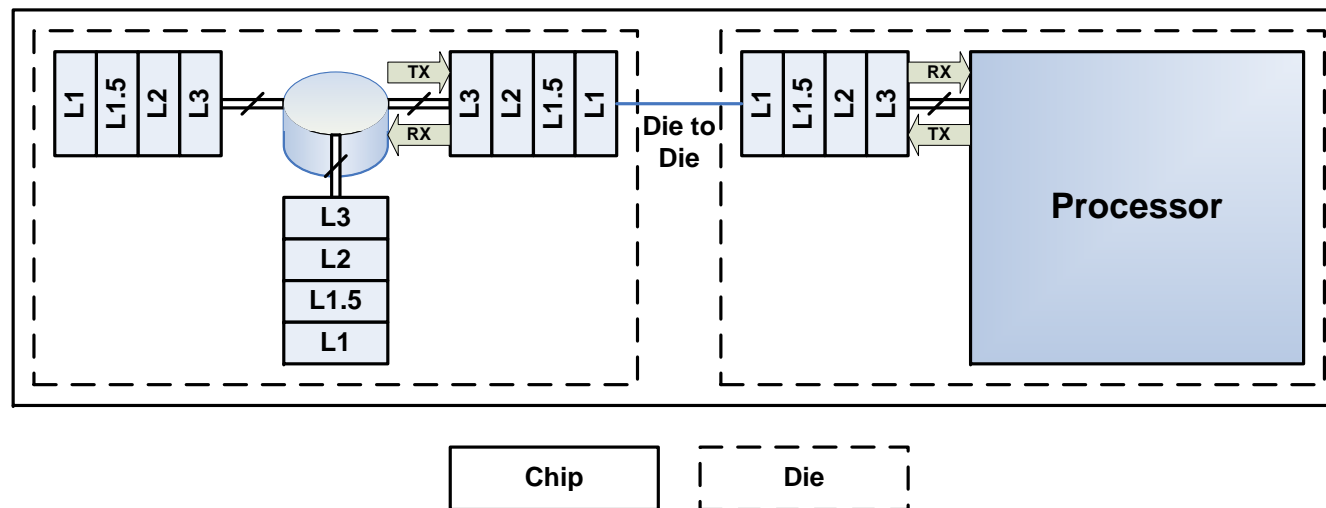
- More complex and thus has a high silicon area
- need of special circuitry to recover the clock and synchronized to it in the RX side
- fast transition between off and on state, because of the TX-RX synchronization
- constraints on the electrical design due to very high-speed differential signaling

Chip-to-chip: PHY types

- There are 3 main categories of PHY to consider:
 - Electrical PHYs
 - Optical PHYs (wired and wireless)
 - Wireless (Radio based)
- Each with its own advantages and disadvantages
 - EMC issue for electrical PHYs
 - Power consumption for optical PHYs
 - High BER and different error model for wireless

Die-to-die embedded networks

- From a cost point of view, higher level of integration always means cost reduction if the volume are very high
- The logical next step to chip-to-chip is then die-to-die use cases
- Of course, existing applications designed for chip-to-chip shouldn't need any redesign to be truly cost effective



Die-to-die: cost factors



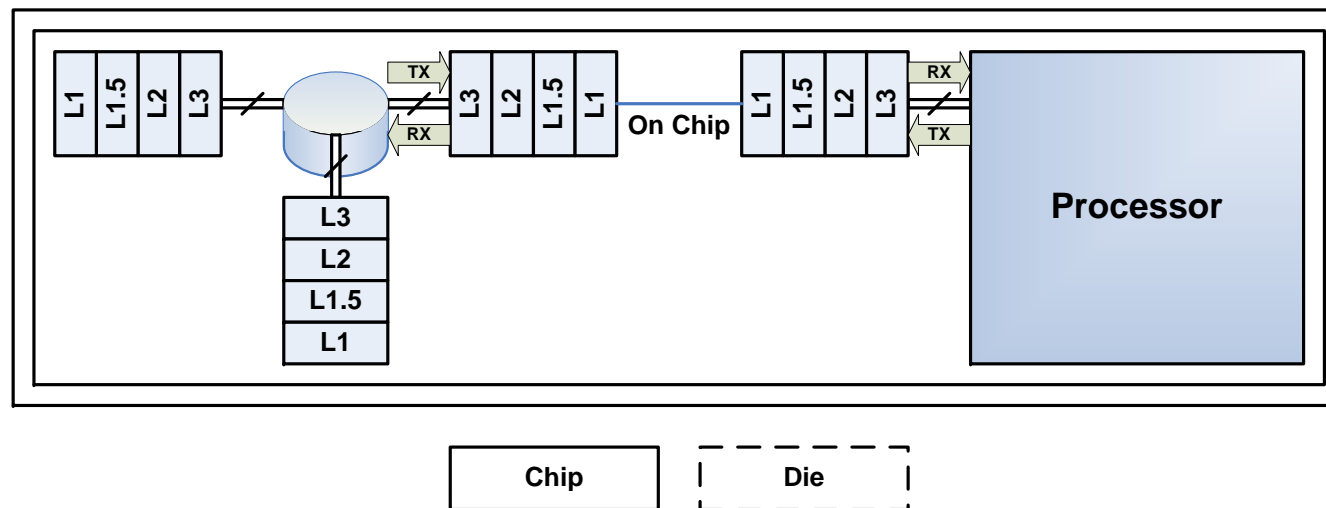
- What are the main cost factors?
 - Assembly
 - Number of pins (but less critical than in chip-to-chip case)
 - Die needs extra specialized processing
 - Thinning, bumping, etc.
- Not so critical factors
 - Number of links
 - EMC are less of an issue (short distance)

Die-to-die: PHYs

- In this case, there is no clear consensus
- Simple parallel busses can do the job
 - but with a trade-off between number of signal wires, clock speed, length of the electrical connection, design and material cost
- Clock synchronous PHYs using differential signaling could be used as well
 - Less pins needed, higher clock speed needed, longer electrical connection possible, simpler implementation compared to chip-to-chip can be used
- There is a consensus around electrical PHYs, but optical PHYs may be an option too in the future

On-chip embedded networks

- Once again, cost reduction induced by high volumes means that the logic next step after die-to-die solutions is to have single chip solution
- Note that the design time and cost of single chips solution is getting higher and higher, which may make this approach more costly in near future compare to die-to-die or even chip-to-chip



On-chip: cost factors



- What are the main cost factors?
 - Buffering
 - Power consumption of the metallic wiring
 - Up to 50% of the dissipated energy comes from metallic wiring and clocks distribution in a chip
- Not so critical factors
 - “pin count” which translates to number of wires in one link
 - Number of links

Chip-to-chip vs. die-to-die vs. on-chip

	"Pin count"	Latency	Bandwidth	BER
Chip-to-chip (wired)	Critical, order 1x	High	Low	10^{-12} , 10^{-9}
Chip-to-chip (wireless)	Antennas are the issue	High	Very low to low	10^{-14} , 10^{-6} , 10^{-3}
Die-to-die	Medium, order 10x	Medium	Medium	Considered below 10^{-16}
On-chip	High, order 100x	Low	High	Considered null in most cases

Embedded network vs. sensor network

- Often sensor networks are seen as one special case of embedded networks
- In our research work, there are treated separately and viewed from different angle
 - Sensor networks are seen more as networks of very low power embedded system than as embedded networks
- Practical sensor networks for mobile applications must be extremely low power and have low bandwidth (few kbit/s at most)
- While embedded network are more comparable to PCI-Express or RapidIO
- So essentially in our work, sensor networks are left aside ... most of the time

Our research

NOKIA

Company Confidential

PHY exploration and utilization

- In embedded networks, every mW or penny counts and a design of such systems need a global approach. It's not enough to study and optimize only on aspect, but the whole system need to be optimized
- PHY exploration and research is then of course very important, but a proper protocol usage of the PHY is even more important
- Example:
 - A clock embedded PHY has the lowest pJ/bit figure. If a link is left active continuously but is 80% of the time sending idle symbols, the raw pJ/bit is still unchanged, but the actual pJ/usefull bit is multiplied by 5
- Lots of effort have been and are put in research and PHY specification
 - Low power optical, multi-level logic, fast lock time of embedded clocking, various type of line coding, etc.

Protocol perspective

- After following a bottom-up approach, discussing requirements and implication coming from the low level hardware side essentially, it is time to look at the problem from a top-down perspective
- Since there was and still is a strong emphasis on ease of integration, system composability is of great importance
- Having a network technology supporting only Best Effort traffic is a major drawback for composability, since the overall behavior of the system is not strictly guaranteed, but only statistically guaranteed
- The conclusion was then and still hold today that to be future proof a embedded networked technology for mobile devices should support QoS

Protocol exploration

- Since re-inventing the wheel is not very economical, a huge effort was put in studying existing technologies to see if one would match our needs
- An non exhaustive list of solutions explored include SpaceWire, USB, Fiberchannel, precursor of PCI-Express, Ethernet, RapidIO, IEEE-1394 (FireWire), CAN bus, etc.
- SpaceWire was the best candidate found, but
 - The PHY used DS coding, which doesn't scale well in term bandwidth of a single link
 - The support for L3 was minimal
 - There was no known L4 defined
 - It was really unclear what would be the future if any of the technology
 - No support for QoS

Creation of a proprietary solution

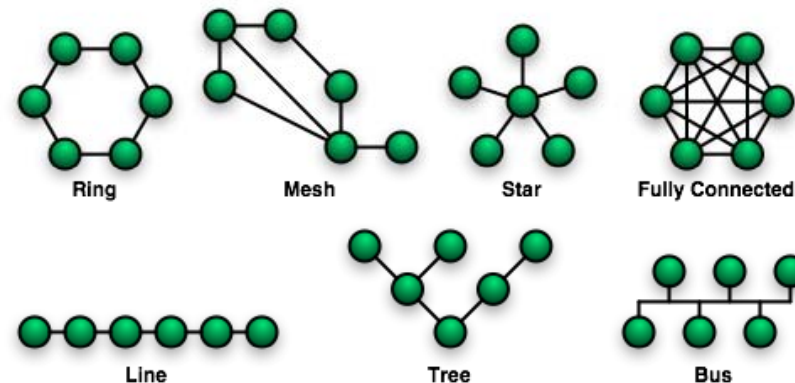
- Our solutions was based on:
 - A L2 having BE and reserved channels following a loosely coupled TDMA scheme using cut-through switching
 - L2 was using a credit based flow control to avoid buffer overflow
 - L3 was using 2 routing schemes following based on wormhole routing
 - Logical routing
 - A special version of source path routing
 - L4 was providing the end-to-end reliability, by replaying packets lost or received containing errors
 - QoS reservation was done by a dedicate protocol traveling the network and making reservations hop-by-hop

Design assumptions: E2E reliability

- The first major design choice was to ensure E2E reliability by replaying corrupted data E2E
- The main reason was a trade-off between buffering, latency and off-chip vs. on-chip usage
 - On-chip BER is very low, it would be a waste to over-design the needed buffers based on the off-chip constraints
 - On the other hand, intrinsic network latencies are much higher off-chip and would lead to poorer response time and increased E2E latency in delivering reliable data
 - But at that time, BER of chip-to-chip was assumed to be around 10^{-14} (so 1 error every 30 hours at 1 Gbit/s)
- This decision forced us to design network to have a small E2E latency as possible

Design assumptions: topologies

- To achieve low latency E2E, wormhole routing and cut-through switching was used.
- Since wormhole routing is well known for its deadlock cases in cyclic topologies and since we were at the time mostly focused on chip-to-chip solutions, we decided to support only acyclic topologies.
 - We essentially postponed the issue for later work when going back to on-chip use cases



NoC vs. chip-to-chip

NOKIA

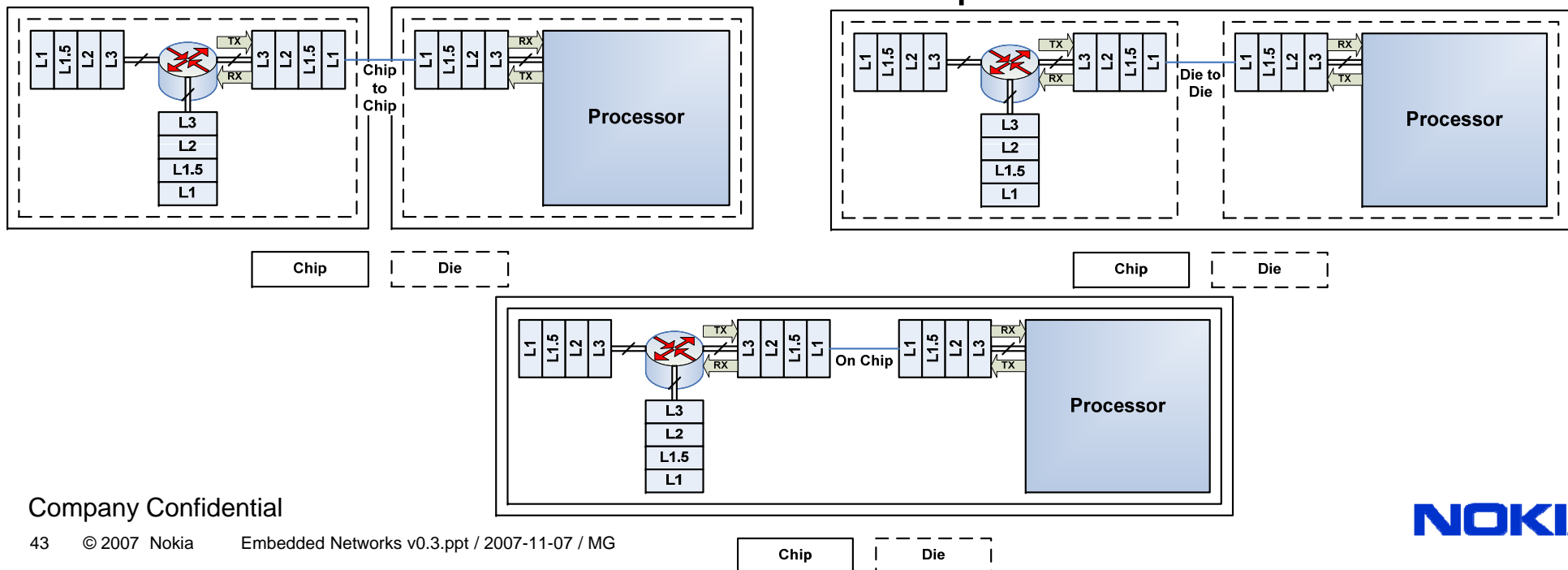
Company Confidential

NoC vs. chip-to-chip

- As discussed earlier, the environment impacts significantly the PHY used in each domain, being on-chip or off-chip/chip-to-chip
- The PHY used in a given domain in turn influence greatly the few key design constraints and goals of embedded networks
- A very good example of this is the how NoC and chip-to-chip research have been almost completely separated and are conducted pretty independently
- It also means that the technical solutions existing in both domain are different and no easily compatible.

The split between NoC and chip-to-chip

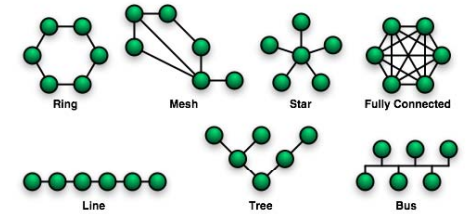
- When looking from a pure technological angle, the constraints being different between both domain logically translates to 2 separated domain of research. So from a technical point of view, the split makes sense
- From a system design or an industrialization point of view, the strict distinction is seen as a bit artificial and problematic



NoC vs. chip-to-chip: packet vs. flit

- In networks in general, often we refer to packet as the atomic unit upon which routing decisions are made
- A pretty fundamental difference between NoC and chip-to-chip is the granularity of this “atomic routing unit”
- For off-chip/chip-to-chip embedded network, this unit is roughly in the order of 100 bytes and is indeed called a packet
- While for NoC/on-chip embedded network, it’s usually 32 or 64 bits and is usually called a “flit”

NoC vs. chip-to-chip: topologies



- We discussed earlier cost factors for on-chip and off-chip solutions
- Off-chip, the number of links has a critical impact on the overall cost and the network topology is usually acyclic
- SpaceWire being a very good exception to this rule
 - Because of its main domain of application, space, redundancy in the network is much more critical
- On-chip links are cheap, so topologies are cyclic basically to improve the performance of the solution and often arranged in a a mesh
 - 2D meshes are very common
 - Torus or N-dimensional meshes are also studied, but not much used so far in practice

NoC vs. chip-to-chip: reliability

- In NoC, the BER are considered insignificant
 - Often not error detection is provided
 - Error recovery is never present
- In the Chip-to-chip case, the BER is seldom ignored
 - Error detection is always provided
 - Error recovery is very often provided
- An interesting point to note is that it may change soon in NoC
- Silicon technology are shrinking constantly, but this applies mostly to the gate size: the metal wiring doesn't scale so well
- So one can expect that more aggressive usage of metal layers will be needed to reduce cost and it will increase the BER on-chip, making error recovery much more interesting

NoC vs. chip-to-chip: system composability (1/2)

- Here many technical and non-technical factors are intertwined
- The mobile industry is at a crossroad, where many key players are, need to redefine their role
 - More and more IC manufacturer will become fabless, because of enormous cost in creating new silicon technologies
 - The IP business and game will likely change drastically
 - New players are coming to try to take advantage of this situation
 - Many aspects of phone manufacturing are becoming commonalities
- System composability and easy integration will become for many companies a key aspect to their success or failure, which will likely force them to deploy technologies applicable off-chip, on-chip, die-to-die, for 3D stacking, etc.

NoC vs. chip-to-chip: system composability (2/2)

- It's also clear that managing the ever increasing complexity of mobile devices will force more rational system design techniques, with as backbone clear standardized interfaces
- But as said earlier, specially when putting into the picture the mobile convergence, QoS will likely have a major impact in system design to
 - Ensure composability
 - But also reduce power consumption

Need for unified embedded network solution

- From a technical point of view, trying to find a embedded network technology applicable for many domains like on-chip, chip-to-chip, die-to-die, using optical and wireless links, etc. is very challenging ... and it makes it very interesting ;-)
- From a mobile industry point of view, it's also clear that something must be done to handle better the system design of complex devices, reduce the time to market, etc.
- The only real question is what company or group of companies will win the battle of defining the next dominant architecture in the mobile world.

Future work and research topics

NOKIA

Company Confidential

Future work and research topics

- Currently, we are mostly finalizing the standardization the embedded network in the MIPI UniPro WG
- But even in the standardization people have spoken already of on-chip extension
- So the next hot topics for a unified approach for on- and off-chip embedded networks are (no order of importance is meant in this list):
 - QoS
 - Security (see Elena Reshetova's presentation)
 - Reliability (optimization of buffers and cost)
 - Power management

Conclusions

NOKIA

Company Confidential

Conclusion: embedded networks

- Essentially, embedded networks are seen by many as a question of survival for Nokia, but also for the mobile industry as a whole
 - Business value chain is changing to a vertical model
 - Time to market is ever more critical
 - The level of complexity has reached a level, which becomes very expensive to maintain without significant changes and radical improvement
- Added to the actual turmoil in the mobile industry (see Google announcement yesterday, changes made in Nokia, ST, TI, etc.), the place of driver of the next dominant architecture is still to be taken

Conclusion: from research to industry standard

- We have been **very** fortunate to have the chance to see the start of an idea based on a vision of what the future should/could be, then drive that idea up to standardization
- There is no magical recipe on how to manage to bring research ideas or visions to an industry standard or to the industry in general. Luck is also part of the equation
- But some few tips
 - Always seek feedback, **never** take it as personal; the more feedback you get, the crisper/better the idea becomes
 - Be **very** persistent, but not stubborn
 - Always keep your idea/vision in sight: the path is not important, the goal is (opposite to Buddhism)

Ultimate Conclusion

Team work, team work and team work
Network of people