



System Integration in the connected consumer domain

Ben Pronk

System Architect BL Digital Video Systems

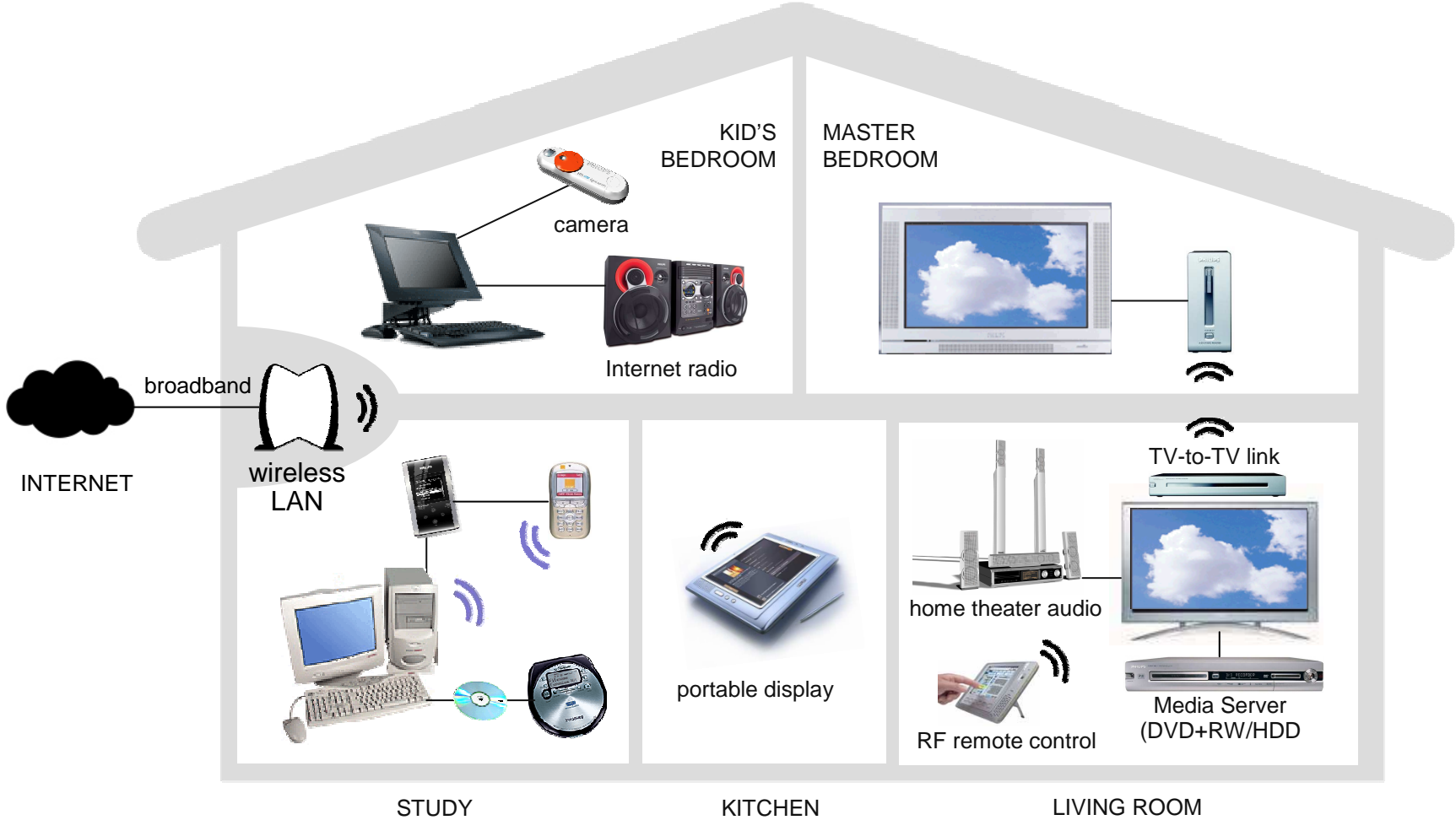
NXP semiconductors



# Outline

- ▶ The context, connected consumer products
- ▶ Major trends influencing system integration
- ▶ A real life case the TV543, connected TV
- ▶ A multi view approach
- ▶ Summary and Conclusions

# Connected consumer products



# Connected consumer products

- ▶ Example the hybrid TV, expanding functionality
  - Analogue TV reception and connectivity like Scart
  - Digital TV reception and connectivity like HDMI
  - Ever growing audio and video processing capabilities
  - Advanced graphical user interfaces: 2D and 3D graphics capabilities
  - TV-applications: Teletext, Electronic Program Guide, MHP, MHEG
  - PC-monitor functionality: VGA, DVI, Display port
  - PC-connectivity: Ethernet, Wifi, USB
  - In home networking: DLNA, uPnP
  - Internet connectivity: YouTube, Catch-up Tv, Portals, VOD
  - USB media devices: Flash devices, Camera's, Camcorders
  - Connectivity to portable media devices: Ipod, mobile phones
  - Recording applications, PVR, Blue-Ray, transcoding
  - Content protection and Digital Rights Management

# Connected consumer products

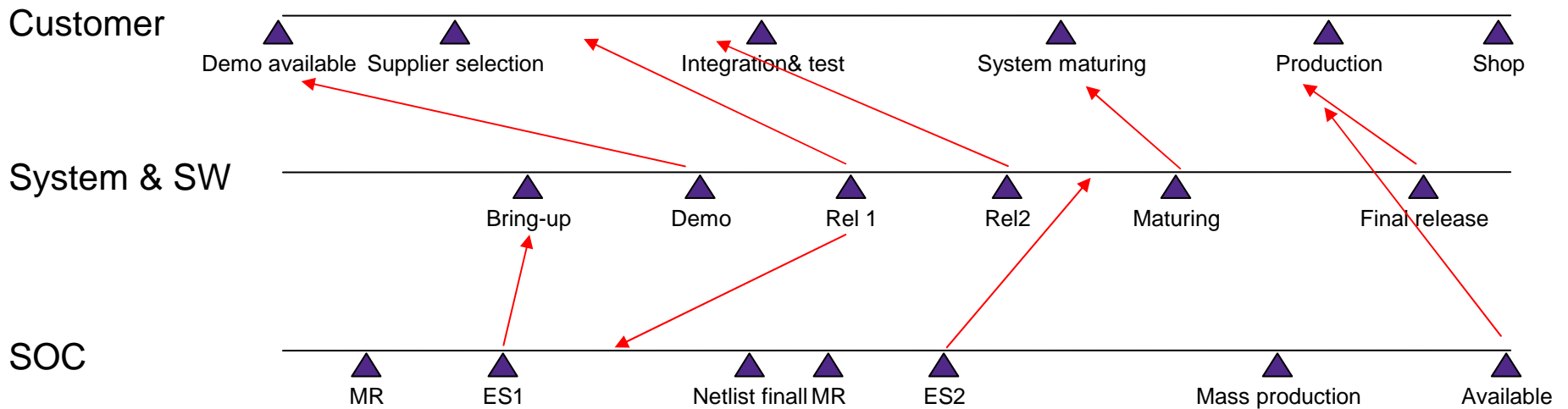
- ▶ Proliferating diversity and change
  - Increasing amount of interconnects
  - With various protocols
  - Exploding amount of potentially connected devices and compatibility
  - Exploding variety in media formats
  - New formats and devices appearing every day
  
- ▶ Yet the basic TV quality of services needs to be maintained
  - Consumer products have a higher expected QOS than a PC
  - In a very competitive and price oriented market

# Major trends influencing system integration

- ▶ Shortened life cycle
  - Consumer product release 6 months after engineering samples
- ▶ Increasing integration of System On Chip
  - > 50M gates
  - Thousands of registers
- ▶ Exponentially growing software content
  - High diversity of interconnects, protocols and standards
  - Major convergence of “PC” type of functionality
- ▶ Extensive use of 3<sup>d</sup> party software and COTS
  - No silicon vendor can provide the requested functionality “on its own”
  - Economy reality and TTM forces the use of COTS

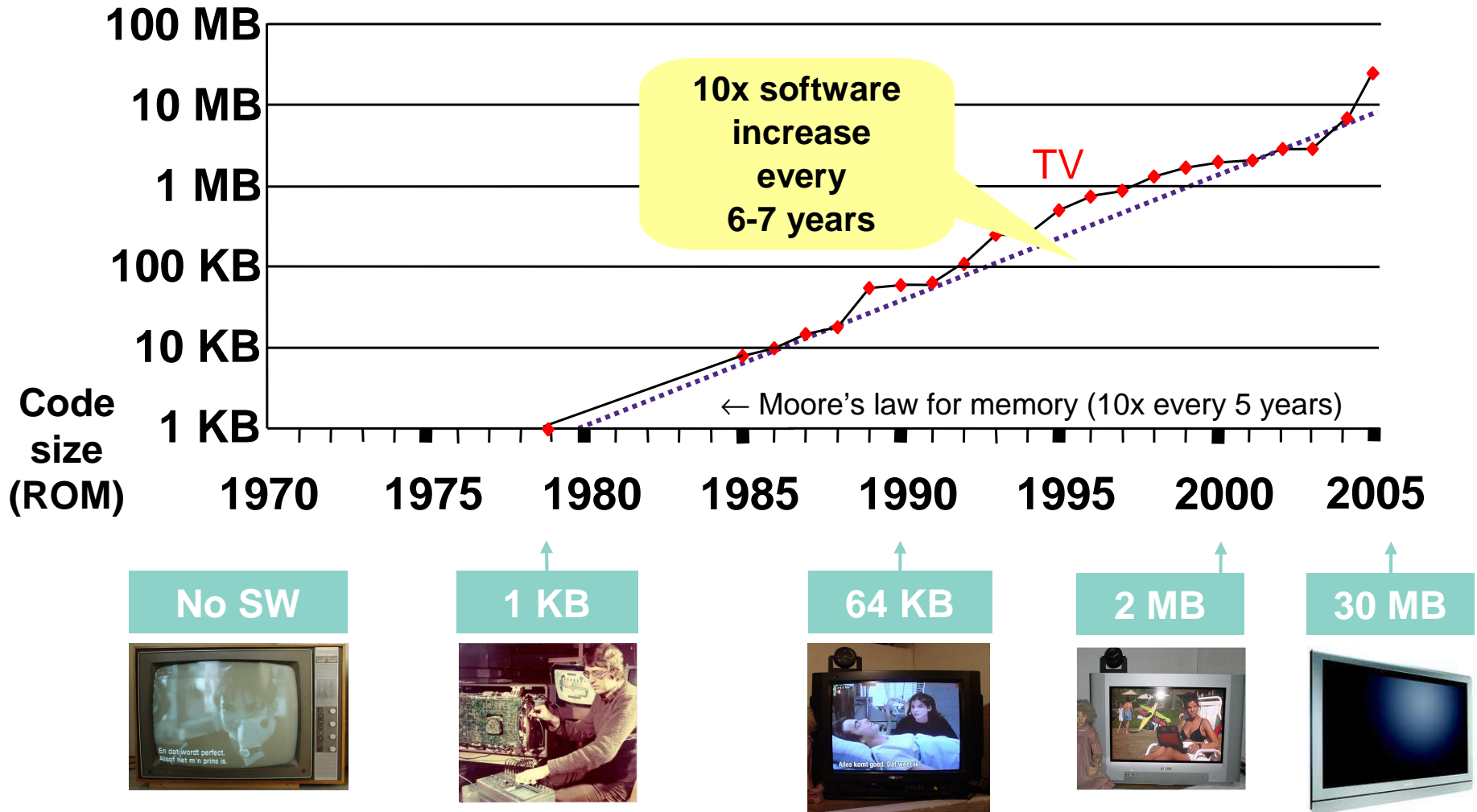
# Major trends influencing system integration

- ▶ Some timelines and their impact on system integration



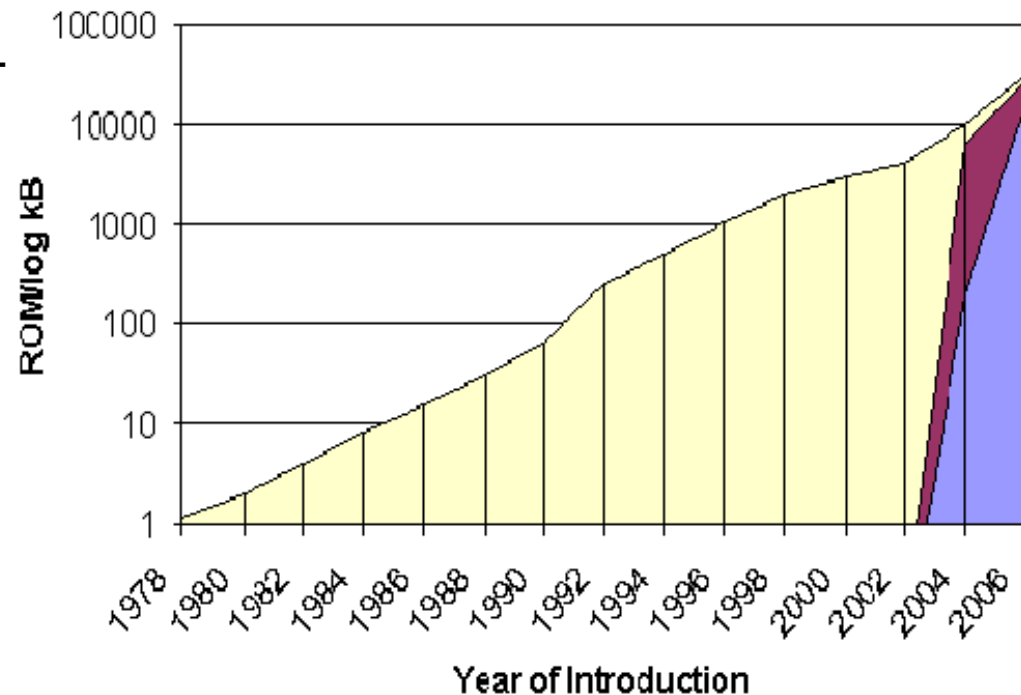
Short timelines and schedule interaction make system integration extremely critical

# Major trends influencing system integration

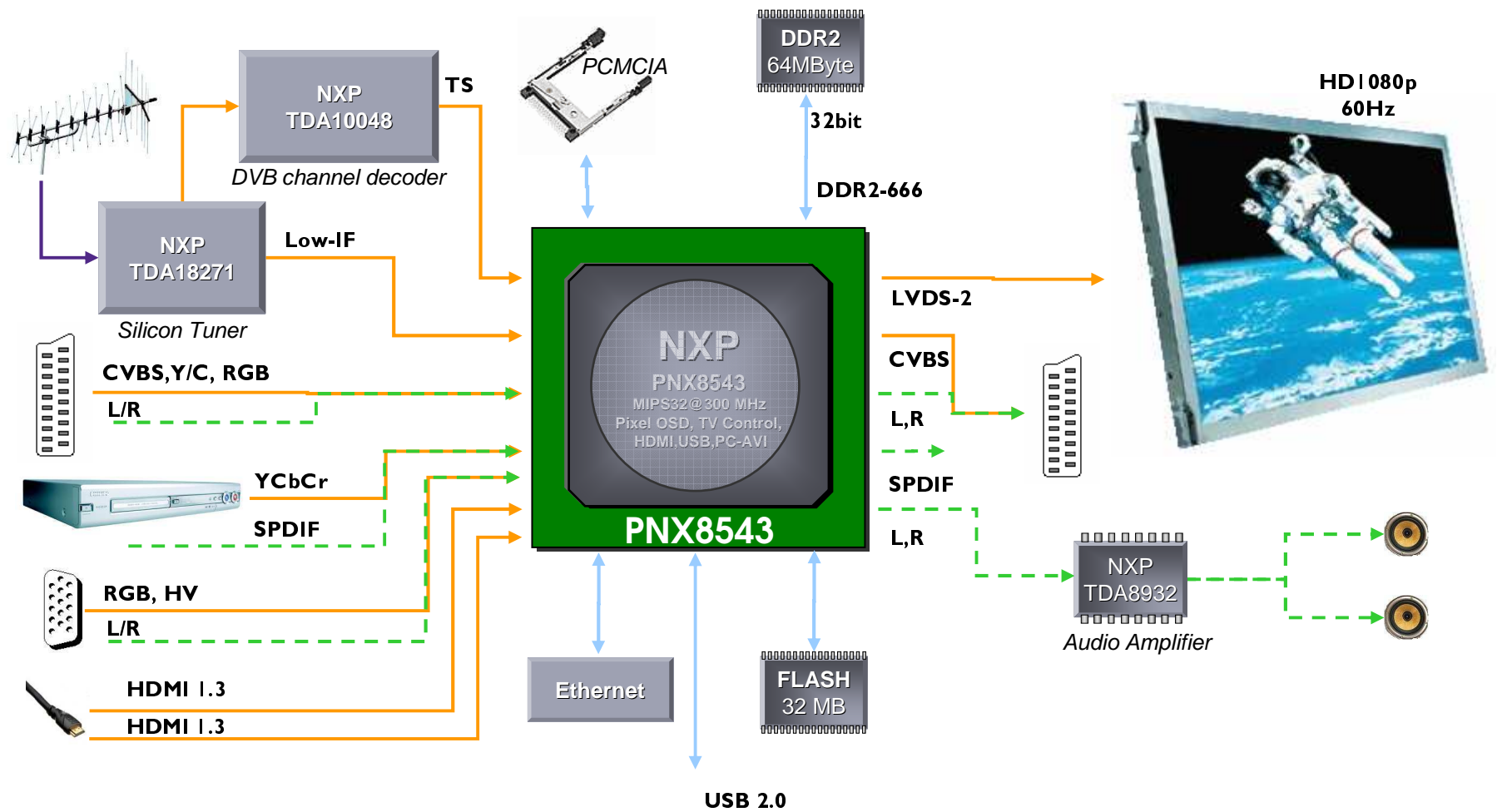


# Major trends influencing system integration

- ▶ Third party and COTS software is required to cope with the growing software content of consumer electronic devices.
  - Economy of scale needed for software.
  - No added value in non differentiating software
- ▶ However
  - Limited standardisation
  - Architectural mismatches
  - Undocumented features



# The TV543, connected TV

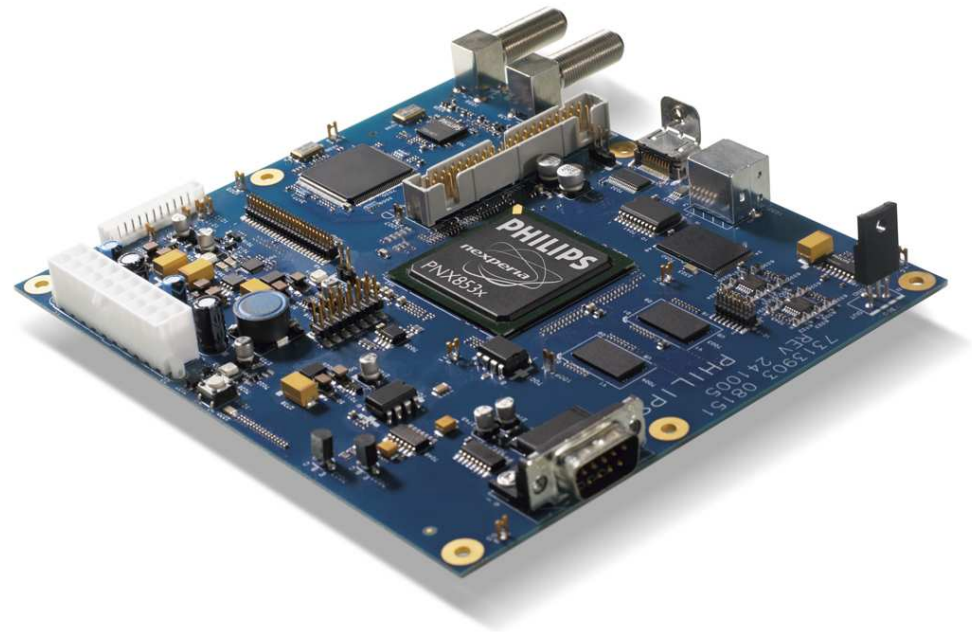


# The TV543, connected TV

- ▶ Broadcast, decoding
  - Digital video decoding: H264, MPEG2, VC1, MPEG4, DivX all for HD
  - Digital audio decoding AC3+, AC3, WMA, MP3,...
  - Analogue video decoding Pal, NTSC, 3D-comb processing
  - Analogue audio decoding
- ▶ Connectivity
  - HDMI1.3, VGA, DVI, Scart,
  - USB2, Ethernet
- ▶ Audio processing
  - 10 band equalizer, bass, volume, loudness, treble
  - Virtual Dolby Digital, Virtual Dolby Surround
  - SRS 3D, True Surround XT, WOW, HD

# The TV543, connected TV

- ▶ Video processing
  - Spatial and temporal noise reduction
  - Contrast and sharpness enhancement
  - High Quality De-interlacing
  - Scaling
  - Colour features
  - Display adaptation
- ▶ Further key features
  - 300 MHz Mips CPU
  - 2D-GFX acceleration
  - Low standby power



# The TV543, connected TV

- ▶ Example video processing, spatial noise reduction
  - Working on Chrominance and Luminance:
  - Keep sharpness in details
  - Prevent from contouring

Noise artifacts  
reduced



# A multi view approach

- ▶ System integration has become a major bottleneck
  - Theoretically of course a complete design should allow for zero integration
  - Reality shows that nowadays sometimes 50% of the project is spend in integration and test.
  - A topic of extensive research within NXP-semiconductors
- ▶ What do we need?
  - Only weeks for integrating the entire software stack on a new SOC
  - Significantly lower cost/throughput time in integration
  - In short “ease of integration”
- ▶ What are the means
  - All HW-SW integration has to be prepared pre-silicon
  - All Software development and integration needs to be ready at samples available
  - The interactions between the subsystems needs to be minimized

# A multi view approach

- ▶ There is no simple single solution
  - A variety of methods and techniques will be needed
  - Both product and process oriented
- ▶ What are “standard” solutions
  - Apply standard solutions from the desktop domain, where-ever possible
  - Abstraction, component based working
  - Philips developed the Koala component model for consumer product families
  - Linux, apply processes to isolate software
  - Allocating separate CPU's to subsystems
  - Standardization of API's, interfaces (UHAPI, OpenMAX, CELF, Autosar, NOC..)

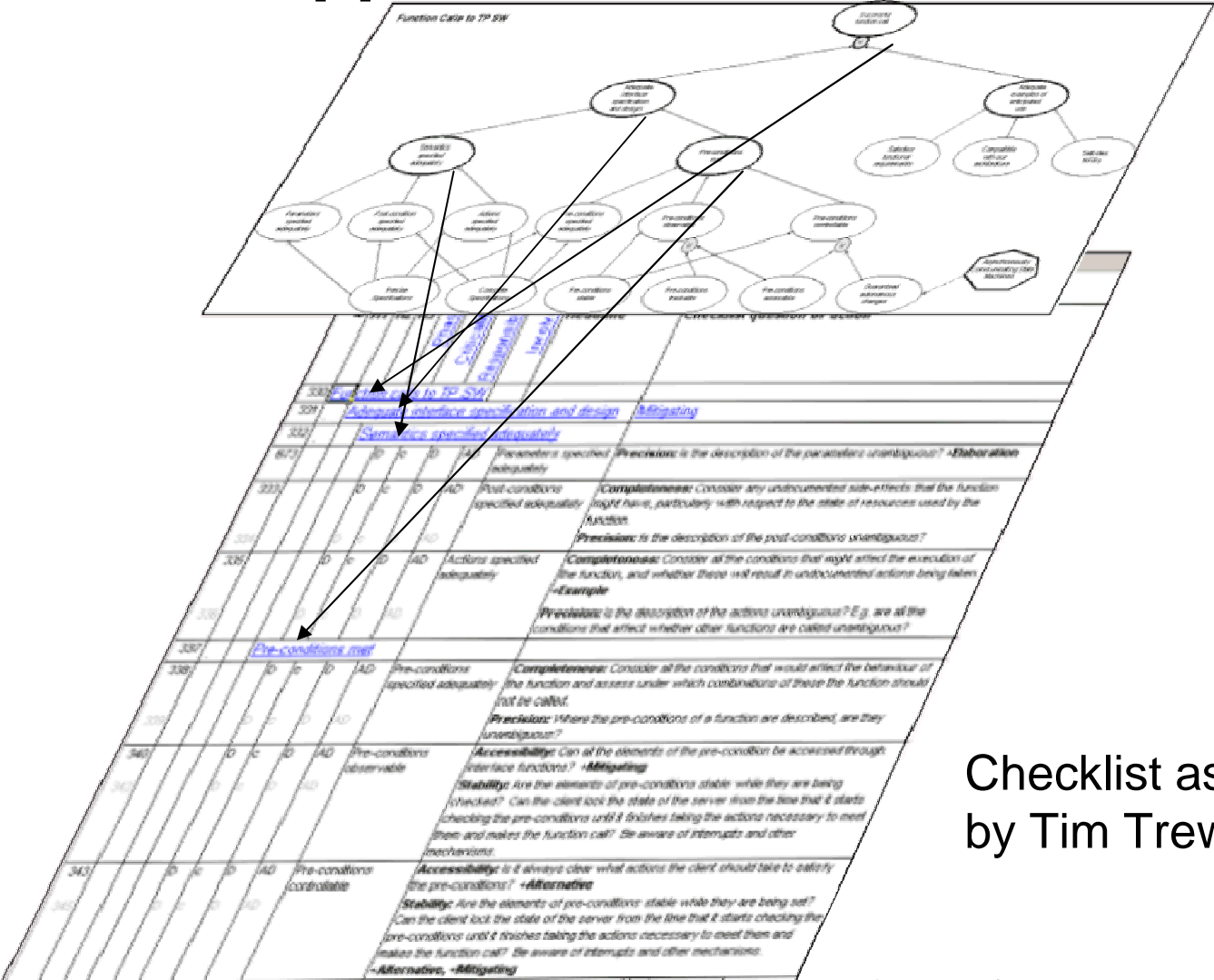
# A multi view approach

- ▶ Early HW/SW integration, so more co-design
  - Simulation in PC-environment
  - Emulation
  - FPGA's
- ▶ Virtual prototyping environment
  - Full system-C model of all Ip's
  - Includes models of all CPU's
  - Allows to run full software packages on prototyping environment at far higher speeds than emulator
  - Allows for real integration and debugging of the software

# A multi view approach

- ▶ 3<sup>d</sup> party software integration, potential issues
  - Insufficient information, lack of documentation
  - Execution architecture,
    - Cycle, memory consumption and strategies
    - Synchronizations
    - Priorities and thread model
  - Context assumptions
    - Operating systems, OSAL
    - Logging, file systems, configuration
  - Other architectural mismatches,
    - interface technologies, thread safeness, notiications, events
    - Decomposition paradigm
- ▶ Within NXP an extensive procedure has been developed to analyse the impact of 3<sup>d</sup> party (Tim Trew) and potential incompatibilities

# A multi view approach



Checklist as developed by Tim Trew

# A multi view approach

- ▶ Service oriented architectures
  - Functionality accessed and offered through services
  - Independence of location
  - Dynamic discovery and binding
- ▶ Network centric physical architecture
  - Systems composed from functional subsystems
  - Fully self contained
  - Interfaced through a network architecture
  - Supported by hardware for low interaction resource sharing

# Summary and conclusions

- ▶ Consumer electronics is confronted with:
  - Shortening of product timelines well below one year
  - Exponentially growing software content
  - Increasing heterogeneity in its software stack
  - Growing investments and throughput time in system integration
  
- ▶ There is no one single solution, problem is approached by
  - Deploying standard PC domain techniques, adapted to comply with the requirements of the embedded domain
  - Processes supporting the analysis of integration of “external” software
  - Service oriented architectures with hardware support to further reduce the interaction.

