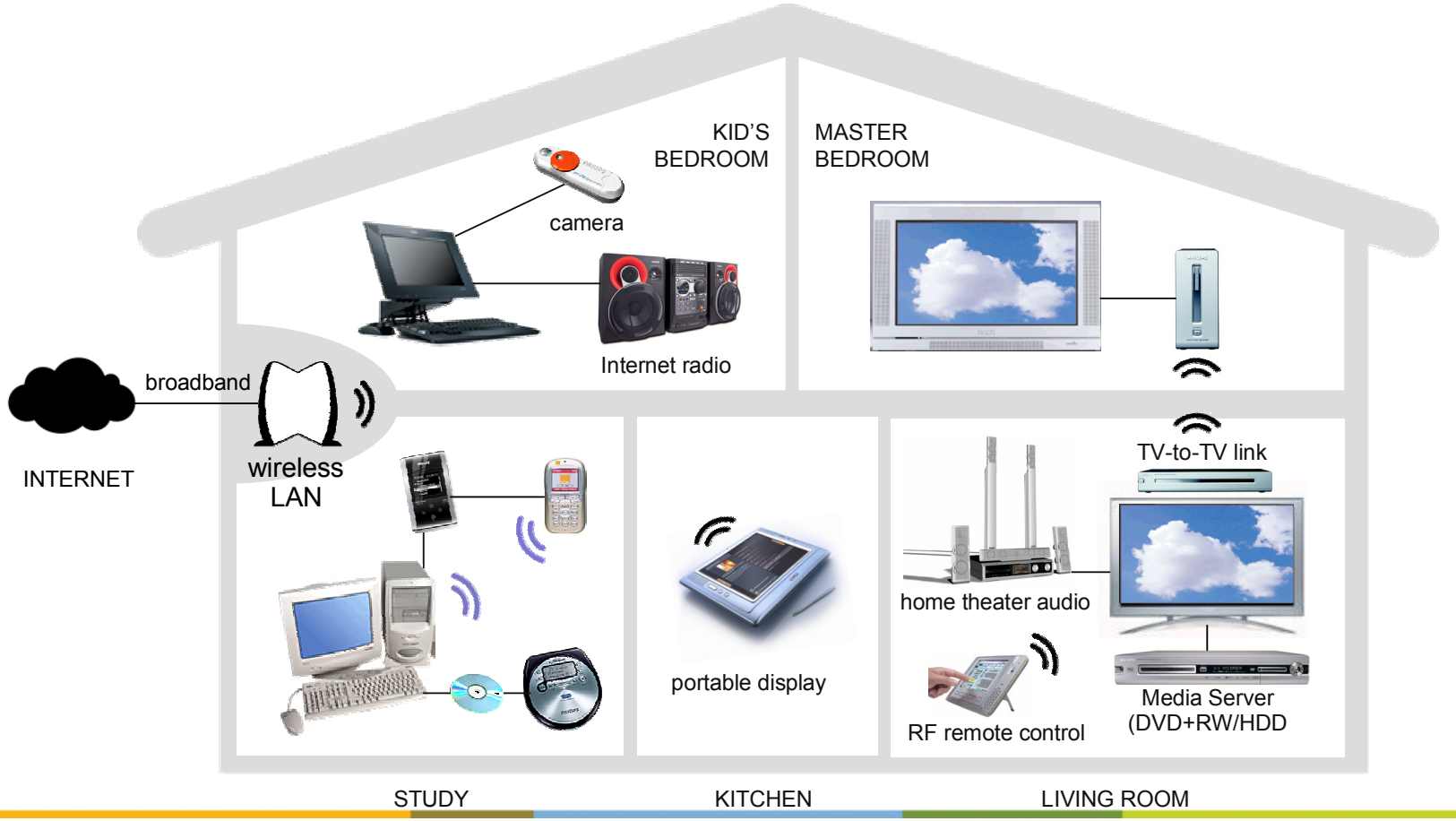
Fault tolerance in consumer products

Ben Pronk

Content

- Consumer electronics, some background
- Reliability and software in consumer products
- Current solutions
- Future outllok

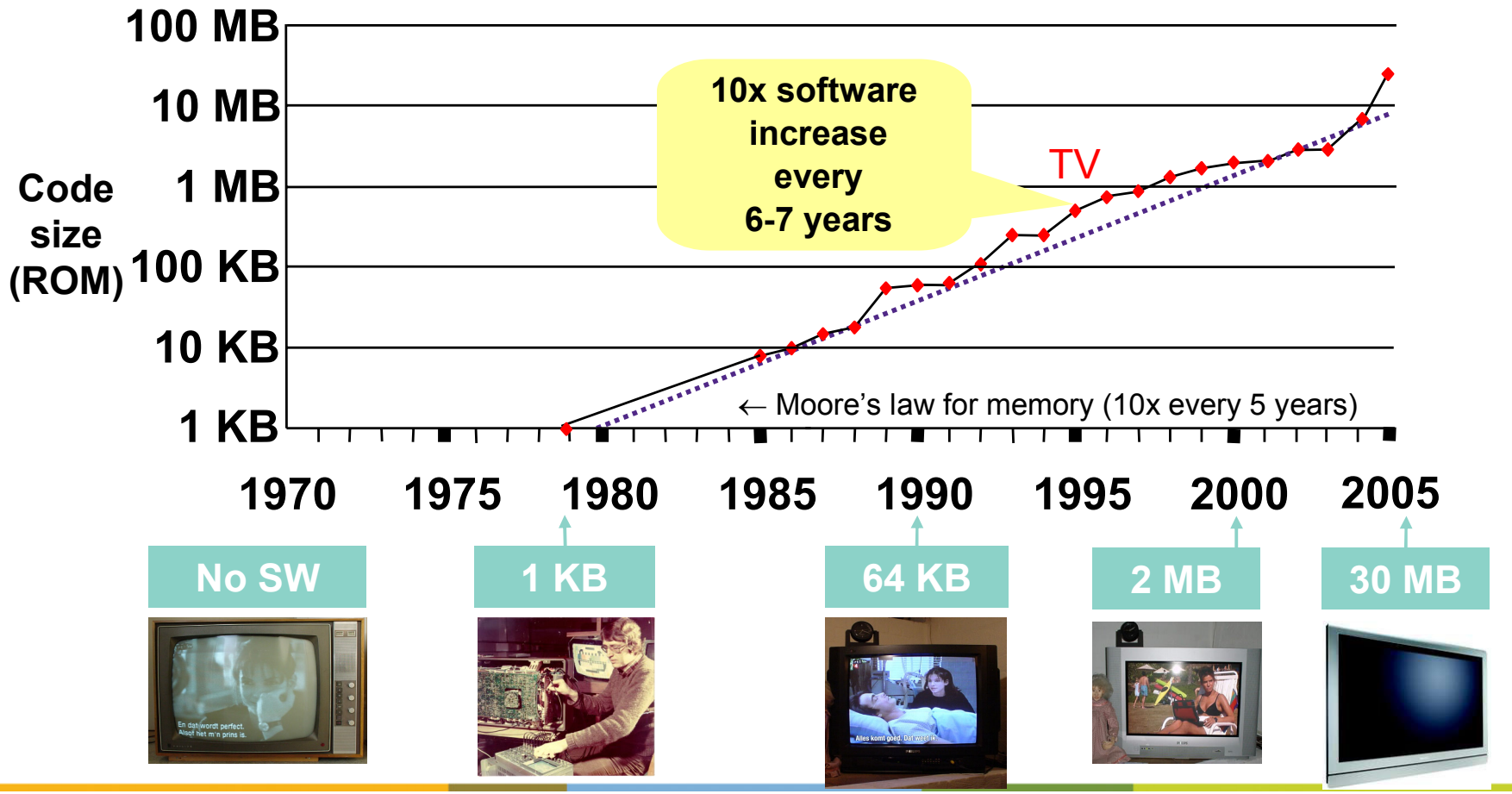
Consumer electronics, some background



Consumer electronics, some background

- In consumer devices we find increasing
  - Digital connectivity and interaction
    - Ethernet, Wireless, USB, NFC, Bluetooth, HDMI, 1394
  - Cross-functionality
    - TV as monitor, PC as TV-receiver, MP3 player in TV
  - Steep rise in number of supported formats
    - Mpeg2, Mpeg4, H264, AVS, DivX, JPEG, MP3,……..

Consumer electronics, some background



Source: Philips Consumer Electronics

Consumer electronics, some background

- No or only limited remote service capabilities
  - No large log files or trace buffers
  - No extensive service reports
  - No remote monitoring
  - If it fails in the eye of the customer it will be returned to the shop
- The business aspects
  - Consumer electronics is a low margin business
  - Just a few percent of returns will eat away the margins
  - Will undermine your position in the market (internet logs)
  - Will cost you shelf space in retail chains
  - Even an update in the manufactoring/logistic chain is very costly.
- So functionality, connectivity and software content go up
  - While maintinaing the same low “call rate”

Reliability and software in CE products

- Historically a CE product is some electronics and a cage
  - Reliability was determined by PCB/electronics reliability
  - This is reasonably well understood and well predictable

- Nowadays, with a million lines of code and full interconnectivity
  - Software and interoperability issues become a major issue
  - And already determine a significant part of the “call rate”

Reliability and software in CE products

- TV, historically
  - “normal” electronics wear
  - Bad signal quality
  - Limited content dependence only example teletext

- TV-now
  - Many many content dependencies, digital TV, all sorts of codecs
  - Many unexpected connections
  - Software crashes
  - And yes also still “normal” electronics

Reliability and software in CE products

- Note that also the user interaction becomes increasingly complex
  - Much more complex installation
  - Connection/network with all kinds of devices (wired/wireless)
  - Increasingly complex “PC-type” of functions

- Also this has lead to a large growth in “call rate”
  - CE device manuals and UI’s are not known to be intuitive
  - People don’t read the manual anyway
  - React unexpectedly on errrors (and draw different conclusions than you)

Reliability and software in CE products

- On the internals
  - CE devices are still very much resource constrained systems
  - So limited ram/flash normally no disk
  - Make use of embedded (sometimes home build) OS's
    - Flat memory space, no protection against overwrites
    - With limited capabilities on protection and recovery
  - Do have to recover always autonomously,
    - There is no repair outside the power switch
  - Yet:
    - Have to implement a steep rising set of requirements
    - Integrate a lot of "third" party components
    - Become increasingly more multi-cpu (3-4 CPU's is no exception)
    - Have to work around failures in the IC's/HW

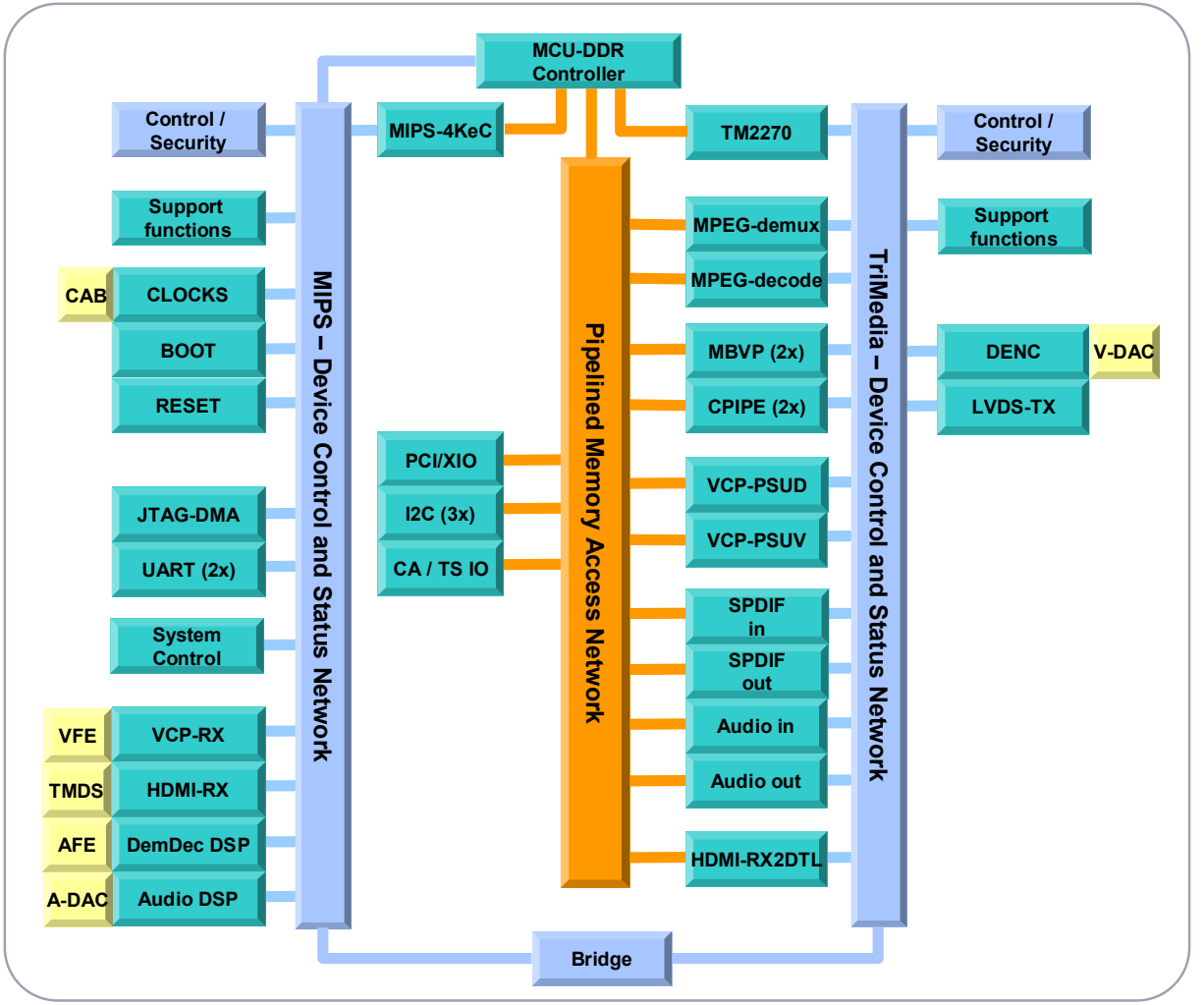
Current solutions

- How do we cope, quite simple measures
  - No assumptions on HW state, reprogram full settings repeatedly
  - Simply reset model
    - No partial resets due to OS memory model and simple process structure
    - Full reset of a processor in case of error/crash
    - Liveness checks between threads/processors
  - Consistency checks (in background tasks)
    - Data structures
    - Stack overflow
    - Code consistency

Current solutions

- How do we cope, maintaining basic functionality
  - Most of the time a TV plays video/audio
  - This function has to be maintained
    - If ths is the case users won't even note resets
  - Approach:
    - Maintain A/V streaming path when host CPU resets
    - Keep hardware state unmodified after a "warm" reset
    - As long as A/V path is only HW (or a separate CPU) this works
  - Does not work for
    - Streaming data coming in from a network in the host CPU
    - Any further data that needs host intervention (UI, teletext)

Current solutions



13

Current solutions

- “Hot boot”
  - Watchdog timing in MIPS and Trimedia,
  - Warm reset, which means restart of the MIPS under special conditions
  - Shared memory and Trimedia continues to operate as is
  - UI and other host controlled aspects “disappear”
  - At restart the HW is not reprogrammed
  - The MIPS re-synchronizes with trimedia
  - A trimedia application cleans up, buffering, connections,
  - Finally MIPS reselects a use case setting everything effectively in a mode

- All by all a complex mechanism, that itself needs a lot of testing.

Current solutions

- And of course testing, testing, testing
  - Long test cycles, many sets in parallel, duration tests
    - One set makes about 15000 hours in its lifetime
    - Many test-hours are needed to have some statistical proof
    - What does this say in a “software” world
  - Field tests to cope with differences in conditions
    - Trucks driving through difficult areas in the world
    - Recordings of all relevant “streams” and exceptions
  - Compatibility tests with e.g. different media, streams, other equipment etc
    - A million types of crappy disks
    - Different types of encoded data, non-standard encodings
    - Many many peripheral devices to interface with

Future

- The situation will get worse in the future
  - Inherent reliability of silicon devcies decreases e.g.
  - More statistical errors in deep submicron devices (45, 32…)
  - E.g. NAND flash error rate keep rising
  - The growth in formats and functionality will not stop
  - As will the growth in software sizes.

- So we have to explore approaches over and above the mechanisms

Future

- More isolation of components
  - More multi-CPU use, with more de-coupled resources
  - Use of standard operating systems with memory protection
  - Linux, WindowsCE
  - Use of process structure and memory protection to safeguard partial restart of the system,

Future

- Various investigations ongoing, NXP research as well as “Trader”

- Areas of investigation:
  - Hardware monitoring of real time data
    - Audio,
    - Video
    - Memory bus access
  - Dynamic adaptation of behaviour (rescheduling)
  - Error detection
    - HW-monitoring
    - Software state consistency checks, potentially hardware assisted
    - Architecture/model based “reliability” check

Future

- Areas of investigation:
  - Program spectra, computer aided de-bugging
  - Indication of “suspicious” code through likelyhood ordering
  - Stresstest upport for memory, bus, CPU etc load

- Usability investiagtion
  - What do users perceive as an error
  - And what as an adequate recovery mechanism