Open DSP Platforms for INDUSTRIAL hearing instrument applications?!?

Stefan Launer,
Sonova, Science & Technology,
HADF June 2017, Oldenburg
Intro Industry – vs Research community?

- Industry does not need an *open DSP platform*….
- We need a full hearing instrument SYSTEM with an open DSP platform
  - Algorithm portfolio
  - DSP & uC SW tools (DSP simulator, multi-core debugger, compiler optimization …)
  - Analogue front-end, power management (low voltage behaviour) output stage …
  - Fitting SW incl all different layers and transformations
  - Development and test environment
- Open platforms (Onsemi) have existed for years, devil is in the details of system Architecture
- Algorithm development on one platform for research ⇔ transformation to another platform? Continous development is more efficient
- Very strong technology convergence.…
- «Make» or «Buy»
Modern hearing systems are *intelligent* systems with …

... a *portfolio of solutions* for different listening and communication situations (“programs”), different external input sources

↔ computation & “house keeping”

Code not modular, highly integrated code structure,
Feature innovation is rather complex ↔ systems innovation
Digital signal processing (dsp) – generation I
Digital signal processing (dsp) – generation II
Digital signal processing (dsP) – generation III
Digital signal processing (dsp) – generation IV
Reprogrammable Platforms - ASIP

Power and delay constraints ⇔ HI-DSP platforms are highly optimized and integrated to work as a seamless ecosystem
- Specialized DSP architecture & functionalities
  - Co-processors (FFT), instruction sets/multi-core parallelism, SIMD,
  - DSP cores vs uController,
  - Dynamic clock
  - ROM / RAM
  - Simulink, C / C++, Assembler
- Algorithm design ⇔ system design not isolated «feature» development
- Development tools ...

SYSTEM Design!

Analog front end ..... DSP / uC .... Wireless .... Memory (ROM / RAM)

01/06/2017
Open DSP Platforms, S. Launer, Oldenburg HADF 2017
Reprogrammable ASIP / DSP Core development – Sonova

• Technologie
  Palio 1  Palio 2  Palio 3

• Produkte
  PALIO  CORE  SPICE  QUEST  VENTURE

• Core Algorithm SW on DSP-Platform
  CORE  SPICE  QUEST  VENTURE

• DSP-Platform
  130 nm  250 nm  180 nm  10.5 MHz

• ASP-Platform
  65 nm  250 nm  180 nm  10.5 MHz

• Wireless - Platform
  65 nm dual-core  90 nm

• Wireless - Frequency
  10.5 MHz

Process: Collaboration with Fabless design house...
Transition between generations: 50 py of work to maintain status quo
Evolution of the Digital Signal Processor (DSP)

- **2004**: First single platform for complete portfolio
  - RAM: 0.45 MBits
  - ROM: 1.03 MBits
  - Technology: 130 nm
  - Transistors: > 6'000'000

- **2008**: More DSP Cycles + more RAM
  - RAM: 1.53 MBits
  - ROM: 1.03 MBits
  - Technology: 65 nm
  - Transistors: > 14'000'000

- **2014**: Multi processor DSP architecture and integrated analog front-end
  - RAM: 2.48 MBits
  - ROM: 1.48 MBits
  - Technology: 65 nm -> 45…. 18 nm, 10.7 mm2
  - Transistors: > 31'000'000
  - Low noise analogue front-end

Technology node level -> smaller? cost for masks, handling in manufacturing....
HI-SW DSP involvement and contribution

>90% of workload is in Platform & Product development

- **DSP functional development** for next generation products
  - **DSP software technology and methodology** for application development, serving **multiple Sonova users**, meeting demands for **quality, safety** and **time-to-market**
  - **DSP application development**, time and quality, ready for mass production
Requirements

• Power and delay constraints
  ⇔ HI-DSP platforms are highly optimized and integrated to work as a seamless ecosystem
  • Specialized DSP architecture & functionalities
    • Co-processors, instruction sets, DSP cores vs uController, ROM / RAM
  • Algorithm design ⇔ system design not isolated «feature» development
  • Fitting SW
  • System-delay, power consumption, low voltage behavior, output stage, front-end pad-outs ...
  • implement solution where most effective and efficient, e.g. process calculations off-line vs online (FSW vs DSP)
    • save battery power
    • implement in a technology that allows for fast development (software high-level language vs hardware chip)

• Tool set on different levels: high level SW ... DSP bit level
  • Maintenance...

• System design: Fitting SW, algorithm, «house keeping» & hardware Co-Design

• Today’s reporgammable platforms ⇔ «open»?
  • No «SDK» environment – «internal tools & docs», learning by doing

• Regulatory aspects
  new MDR: continuous development process

• Continuum: Research platform -> product development, no platform switching?

• Open source vs patent protect?
  • Challenge: Universities and Patent Trolls