KU LEUVEN

Digital Signal Processing

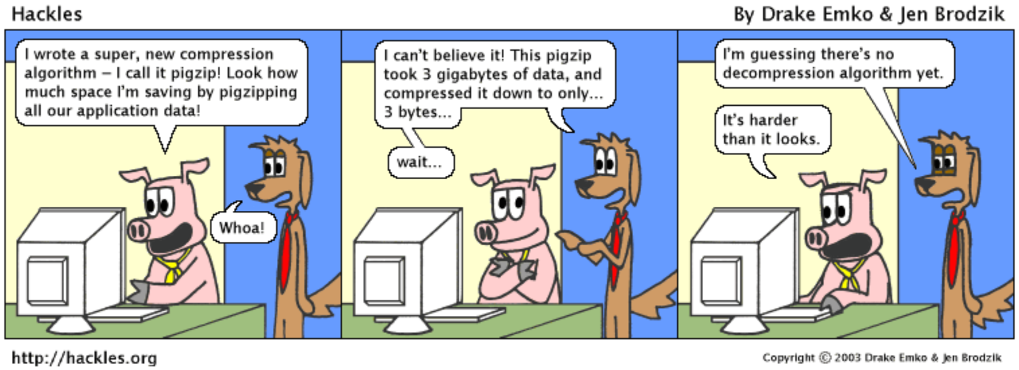
Lab 2: Embedded DSP implementation of energy-based voice activity detector

Toon van Waterschoot, Marc Moonen

ESAT – Departement of Electrical Engineering
KU Leuven, Belgium

Digital Signal Processing: Lab Sessions

- **Session 1:** Signal generation & analysis in Matlab
- **Session 2:** Embedded DSP implementation of energy-based voice activity detector
- **Session 3:** Filter analysis & implementation in Matlab
- **Session 4:** Embedded DSP implementation of FIR filter
- **Session 5:** NLMS adaptive filtering in Matlab
- **Session 6:** Embedded DSP implementation of NLMS adaptive filter
- **Session 7:** Embedded DSP implementation of acoustic echo canceller



KU LEUVEN

Image source: hackles.org

DSP = Digital Signal Processor

- **DSP** = Microprocessor specifically designed to perform digital signal processing operations (e.g. multiply-accumulate, FFT, …) while satisfying real-time constraints
- **Real-time** = processing of one sample takes no more time than one sampling period
- **DSP advantages** over general-purpose microprocessors:
  - faster execution
  - less power consumption
  - smaller chip area
  - relatively small development time

DSP alternatives

- **ASIC** = Application-Specific Integrated Circuit
- **FPGA** = Field Programmable Gate Array

	ASIC	FPGA	DSP
Performance	*Very high*	*High*	*Medium high*
Flexibility	*Very low*	*High*	*Medium high*
Power consumption	*Very low*	*High*	*Low medium*
Development time	*Long*	*Medium*	*Short*
Cost (area)	*Low*	*High*	*Medium*

Source: Reddy Serandra, "Embedded DSP Devices", in *Secure Smart Embedded Devices, Platforms, and Applications,* Springer, 2014.

Some typical DSP properties

- **Low cost:** as low as $2/processor in volume
- **Deterministic latency:** deterministic interrupt service routine latency guarantees predictable I/O rates
  - On-chip direct memory access (DMA) controllers
    - Processes streaming input/output separately from CPU
    - Sends interrupt to CPU when frame read/written
  - Ping-pong buffering
    - CPU reads/writes buffer 1 as DMA reads/writes buffer 2
    - After DMA finishes buffer 2, roles of buffers switch
- **Low power consumption:** 10-100 mW (but can be more for some tasks)

Fixed-point vs. Floating-point DSP

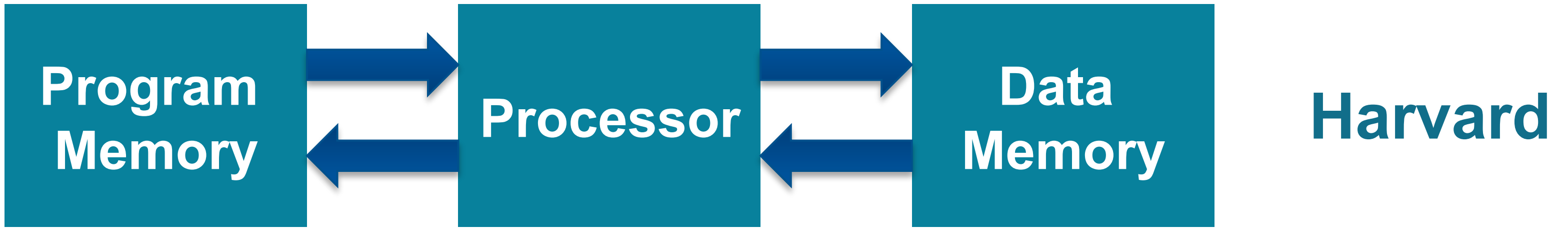
	Fixed-point	Floating-point
Per unit cost	*$2 and up*	*$2 and up*
Prototyping time	*Long*	*Short*
Power Consumption	*10 mW – 1 W*	*1 – 3 W*
Battery-powered products	*Cell phones* *Digital cameras*	*Very few*
Other products	*DSL modems* *Cellular base stations*	*Pro & car audio* *Medical imaging*
Sales volume	*High*	*Low*
Prototyping	*Convert floating- to fixed-point* *Redesign algorithms*	*Reuse desktop simulations* *Feasibility check before investing in fixed-point design*

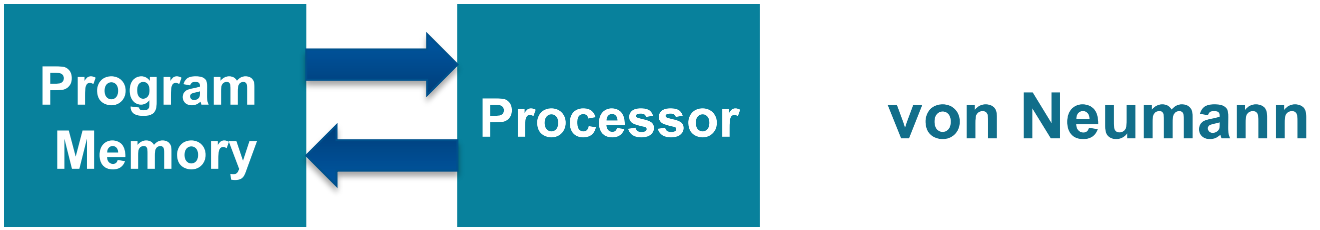
KU LEUVEN

Source: B. L. Evans, *Course notes*, Univ. Texas, USA.

Processor Architecture

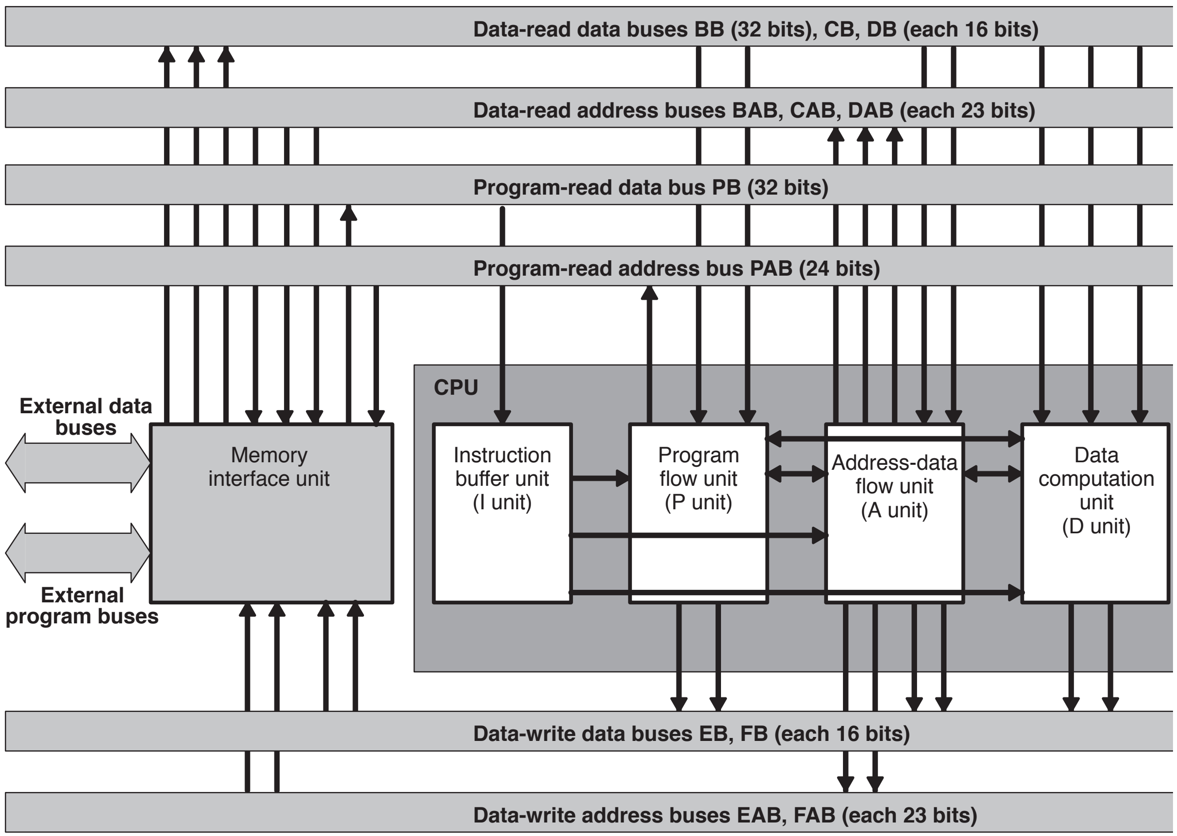
- **Microprocessor architecture:** Harvard or von Neumann





Processor Architecture

- **C55xx DSP architecture:** extended Harvard architecture



KU LEUVEN

Image source: ti.com

Processor Architecture

- **Enhanced DSP architectures:**
  - More parallelism
    - increase number of operations that can be performed in each instruction (by adding more execution units, e.g. MACs, ALUs)
    - increase number of instructions that can be issued and executed in each clock cycle
  - Multi-core DSPs
  - System on Chip (SoC) products

Example DSP chip: C5535 eZdsp chip

- **TMS320C5000 DSP family**
  - 16-bit fixed-point DSPs with performance up to 300 MHz (600 MIPS)

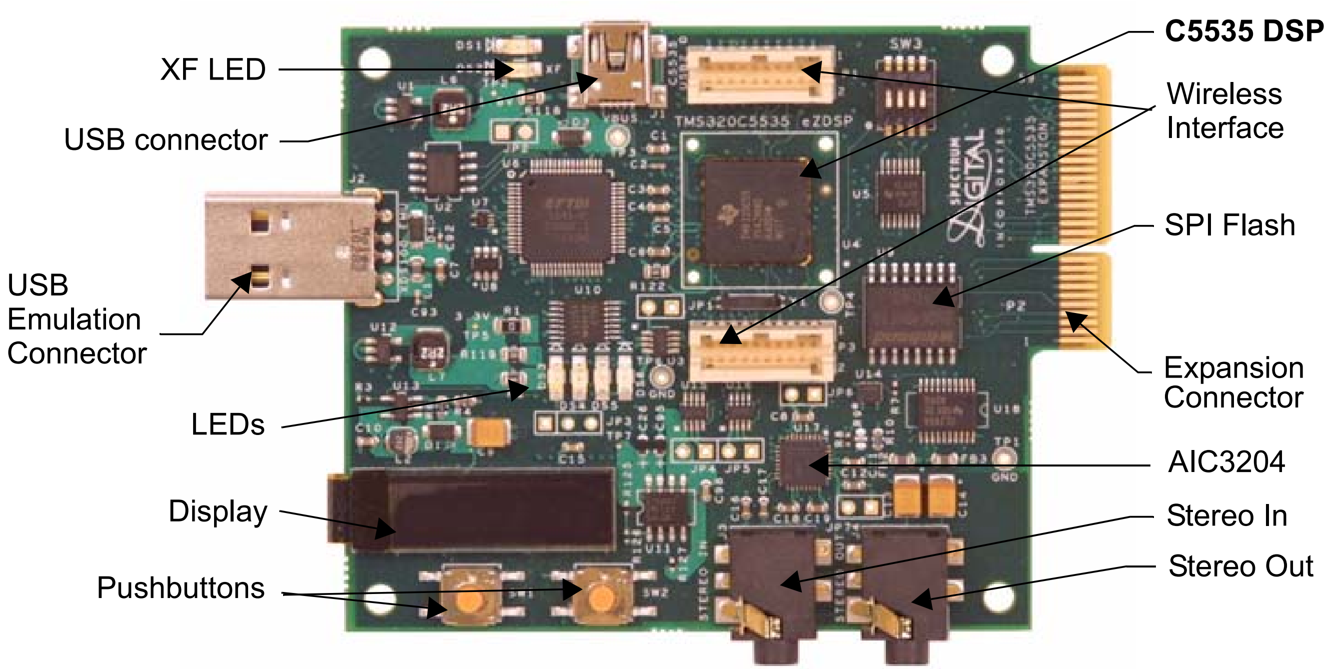
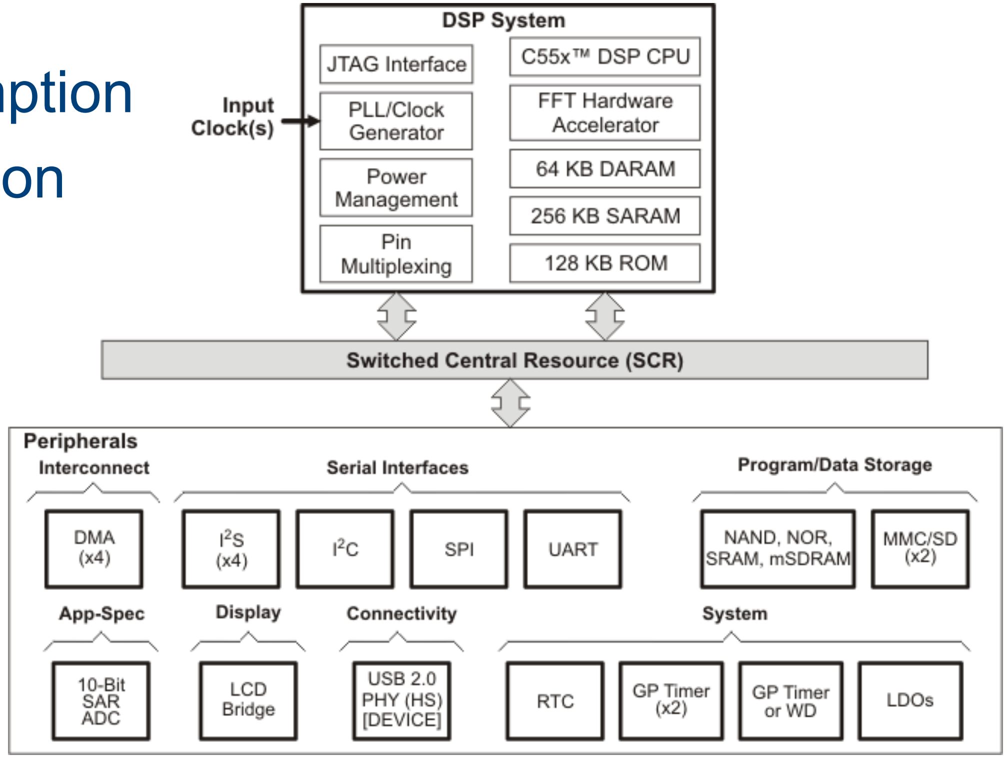


Figure 1-1, Key Features of the C5535 eZdsp (top)

Image source: ti.com

Example DSP chip: C5535 eZdsp chip

- **TMS320C5000 DSP family**
  – 16-bit fixed-point DSPs with performance up to 300 MHz (600 MIPS)
  – ultra-low power consumption
  – high peripheral integration



KU LEUVEN

Image source: ti.com

Embedded DSP implementation of energy-based voice activity detector

- **Experiment 1.1: Get started with CSS and eZdsp**
  - Install Texas Instruments Code Composer Studio v6.1.3 in Linux (see link on course webpage)
  - Download and unzip the folder Exp1.1
  - Open the PDF documentation and carry out the step-by-step procedure to get the DSP up and running

"EVOLUTION HAS BEEN GOOD FOR PMU's."

Image source: e2e.ti.com

Embedded DSP implementation of energy-based voice activity detector

- **Experiment 1.5: Audio Loopback using eZdsp**
  - Download and unzip the folder Exp1.5
  - Open the PDF documentation and carry out the step-by-step procedure to get the audio loopback up and running
    (skip “New experiment assignments”)
  - Have a look at the source code in `audioLoopTest.c` and try to understand how it operates

Embedded DSP implementation of energy-based voice activity detector

- **Experiment 2: Energy-based voice activity detector**
  - Design a VAD algorithm (on paper or in Matlab) that computes the energy for successive length-L segments of a length-N signal (with $L << N$)
  - Implement this VAD algorithm in C
  - Think of a way to check if the VAD algorithm works the way it should (e.g. using eZdsp display or by saving result to file)
  - Modify and test the source code of `audioLoopTest.c` such that the VAD operates on the real-time audio input