Port of a fixed point MPEG2-AAC encoder on a ARM platform

Romain Pagniez

University College Dublin

Information Hiding Laboratory
Department of Computer Science
Belfield, Dublin 4 – Ireland

http://ihl.ucd.ie/

Overview

- Introduction
- Perceptual Audio Coding
- Fixed Point Elements
- Development Toolset
- Implementation
- Results
- Summary

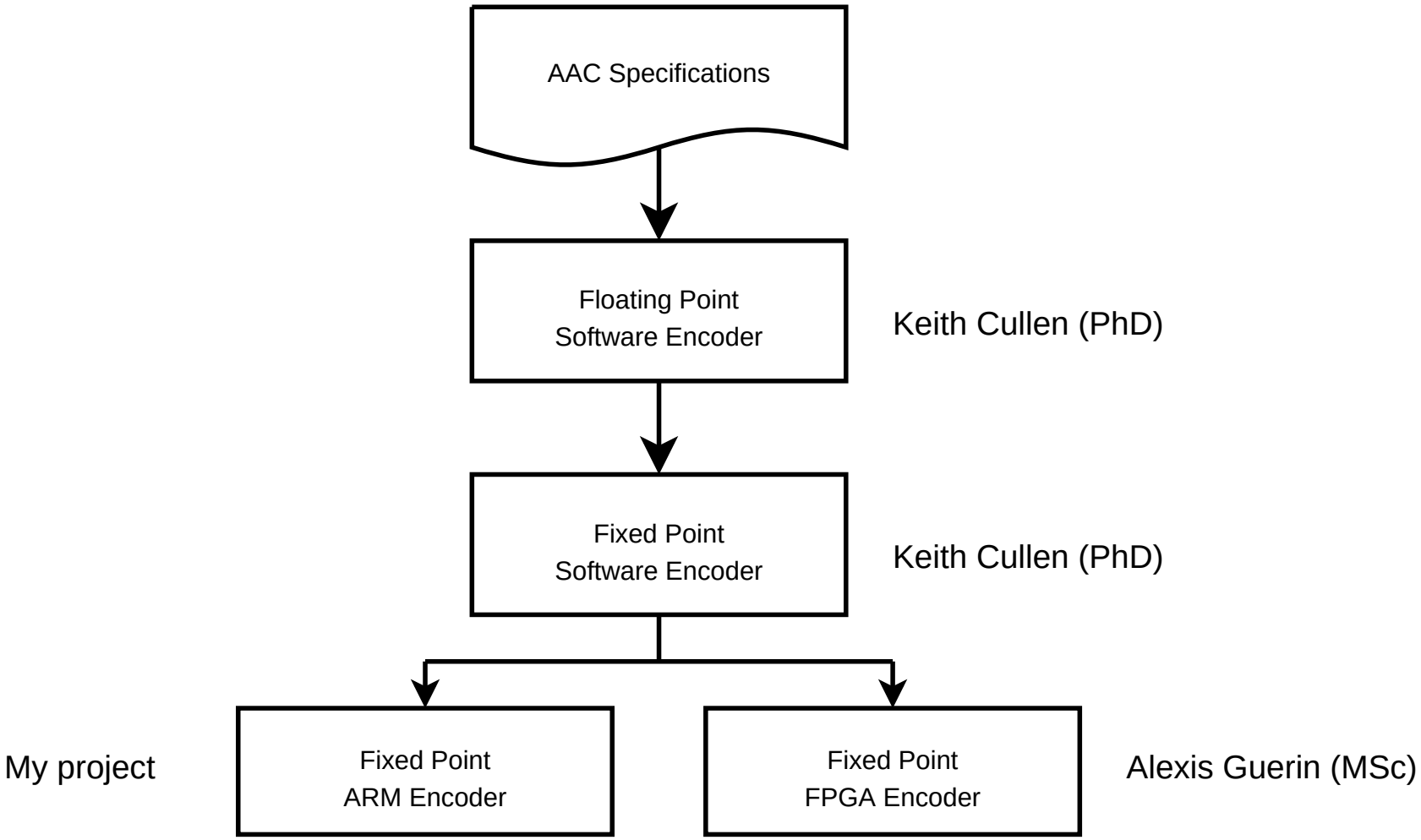
Introduction: hardware encoding

- MPEG2 AAC: state of the art in perceptual audio compression. It achieves a compression rate about 30% higher than mp3. Compressed stereo audio at 128 kbit/s is indistinguishable from the original 1.4 Mbit/s CD quality.
- MPEG-like perceptual encoders are quite complex and involve a significant amount of calculations.
- Computing the algorithm on a dedicated hardware chip could be an efficient solution allowing fast encoding at reduced cost.
- Hardware encoding research is been conduced in the IHL on that specific subject and particularly highlights fixed point encoding.

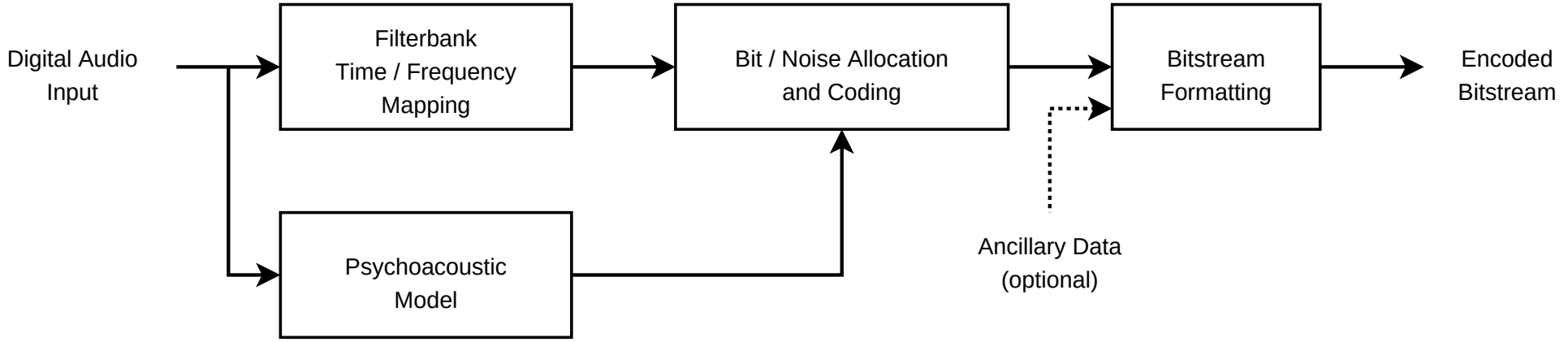
Introduction: main points of my work

- Having the encoder properly working on a ARM platform.
- Communication functions:
  - serial port of the board.
  - Ethernet port.
- Double precision multiplication algorithm.

Introduction: workflow

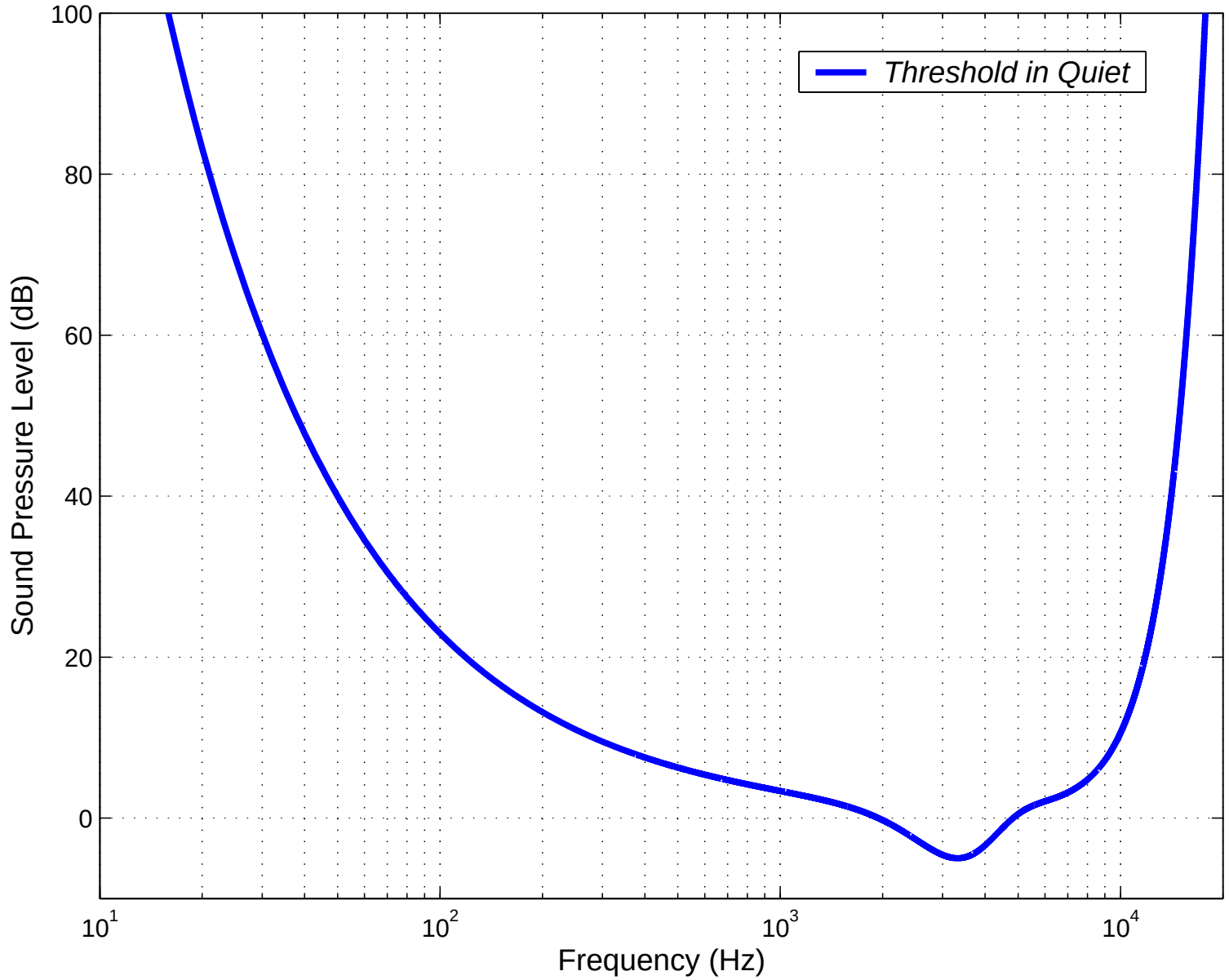


Perceptual Audio Coding: Encoding

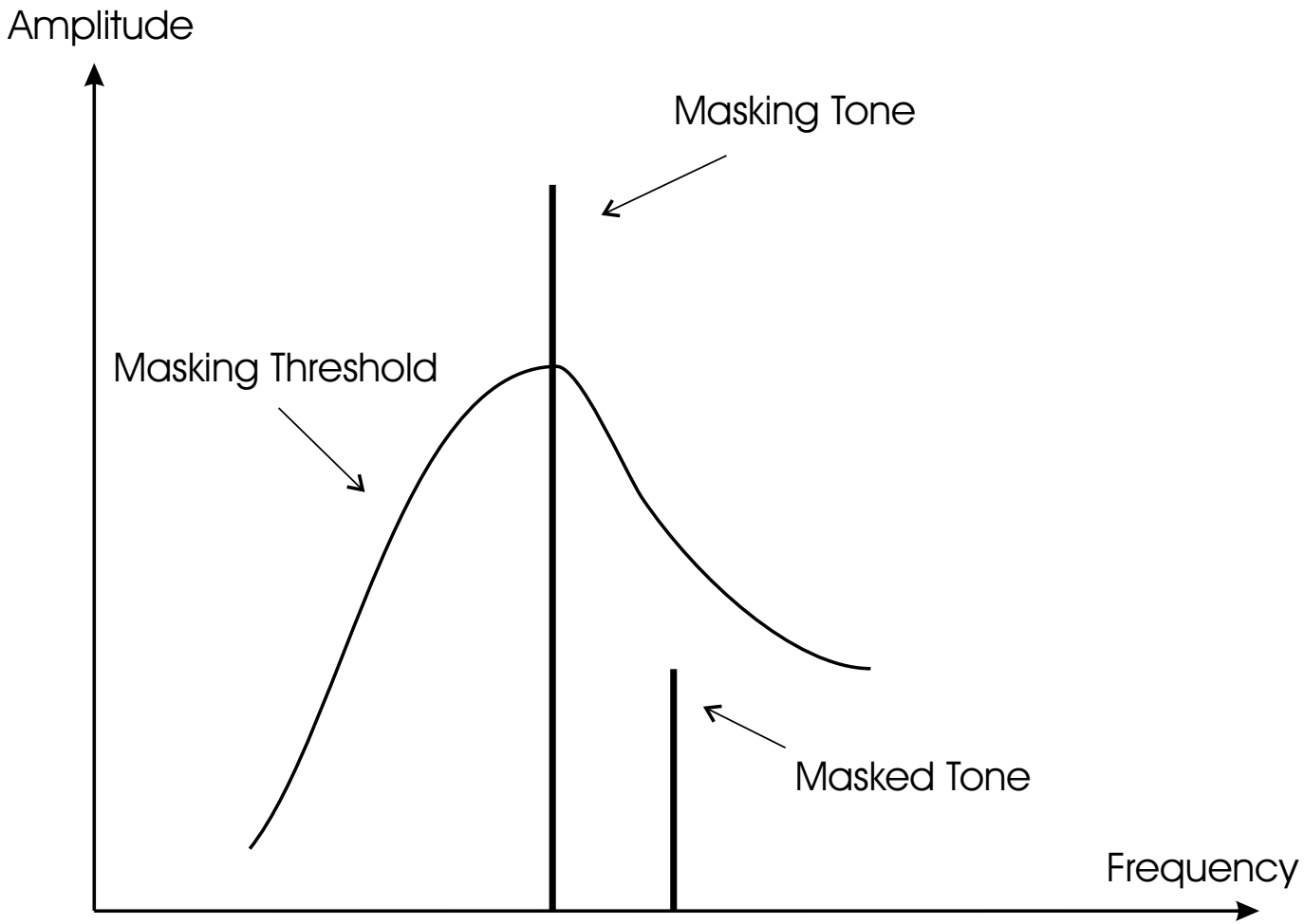


- Filter bank: divides the input stream into multiple subbands of frequency.
- Psychoacoustic model: simultaneously determines the overall masking threshold for each subband.
- Allocation block: uses the masking threshold to decide how many bits should be used.
- Bitstream formatting: multiplexes all the data to be transmitted.

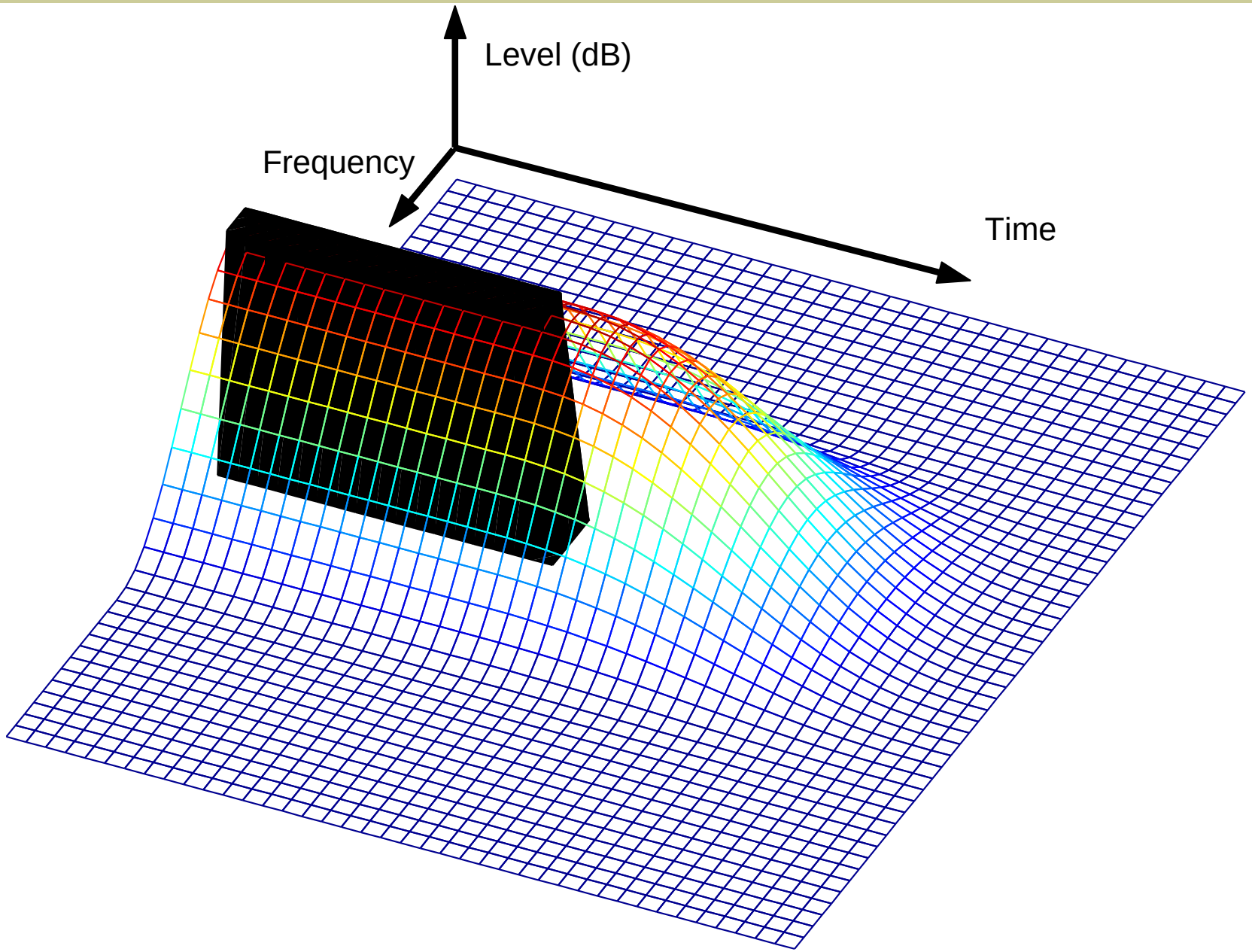
Perceptual Audio Coding: Threshold in Quiet



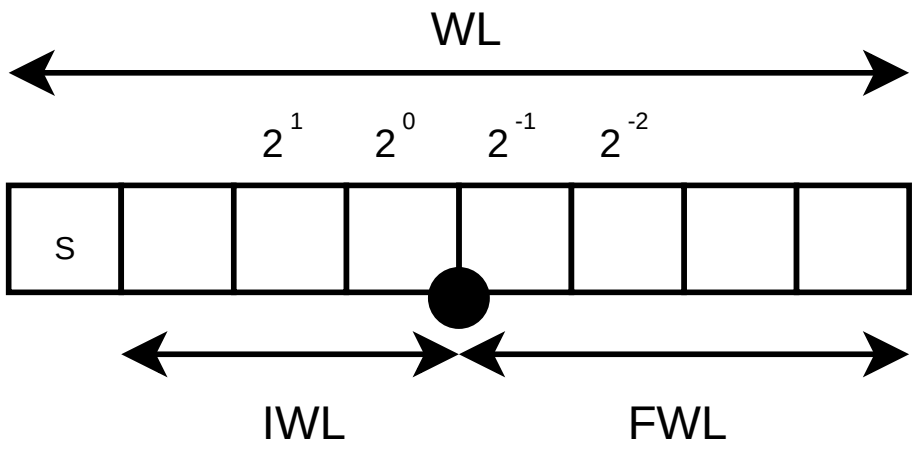
Perceptual Audio Coding: Threshold in Quiet



Perceptual Audio Coding: Masking

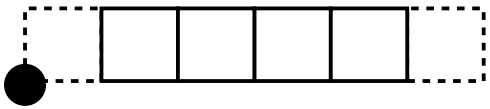
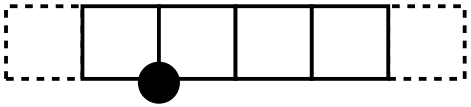
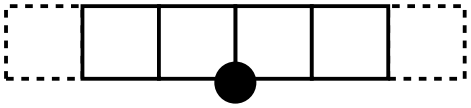
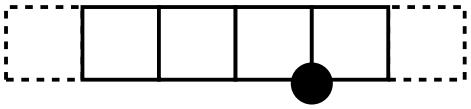
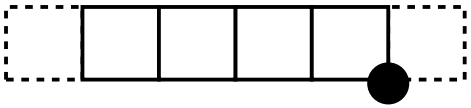
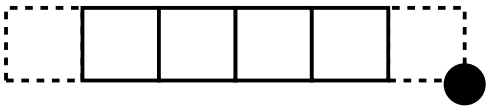


Fixed point



- Fixed point representation virtually places a radix point somewhere in the middle of the digits and uses integer arithmetics.
- This is equivalent to considering integers of portion of some unit.
- For example, one might represent 1/100ths of a unit; with 4 decimal digits, 10.82 or 00.01 can be represented.
- What is the more accurate position for the radix point ?

Fixed point: range and error

Range	4-bit Unsigned	Step Size
[0, 0.46875]		0.03125
[0, 0.93750]		0.06250
[0, 1.87500]		0.12500
[0, 3.75000]		0.25000
[0, 7.50000]		0.50000
[0, 15.00000]		1.00000
[0, 30.00000]		2.00000

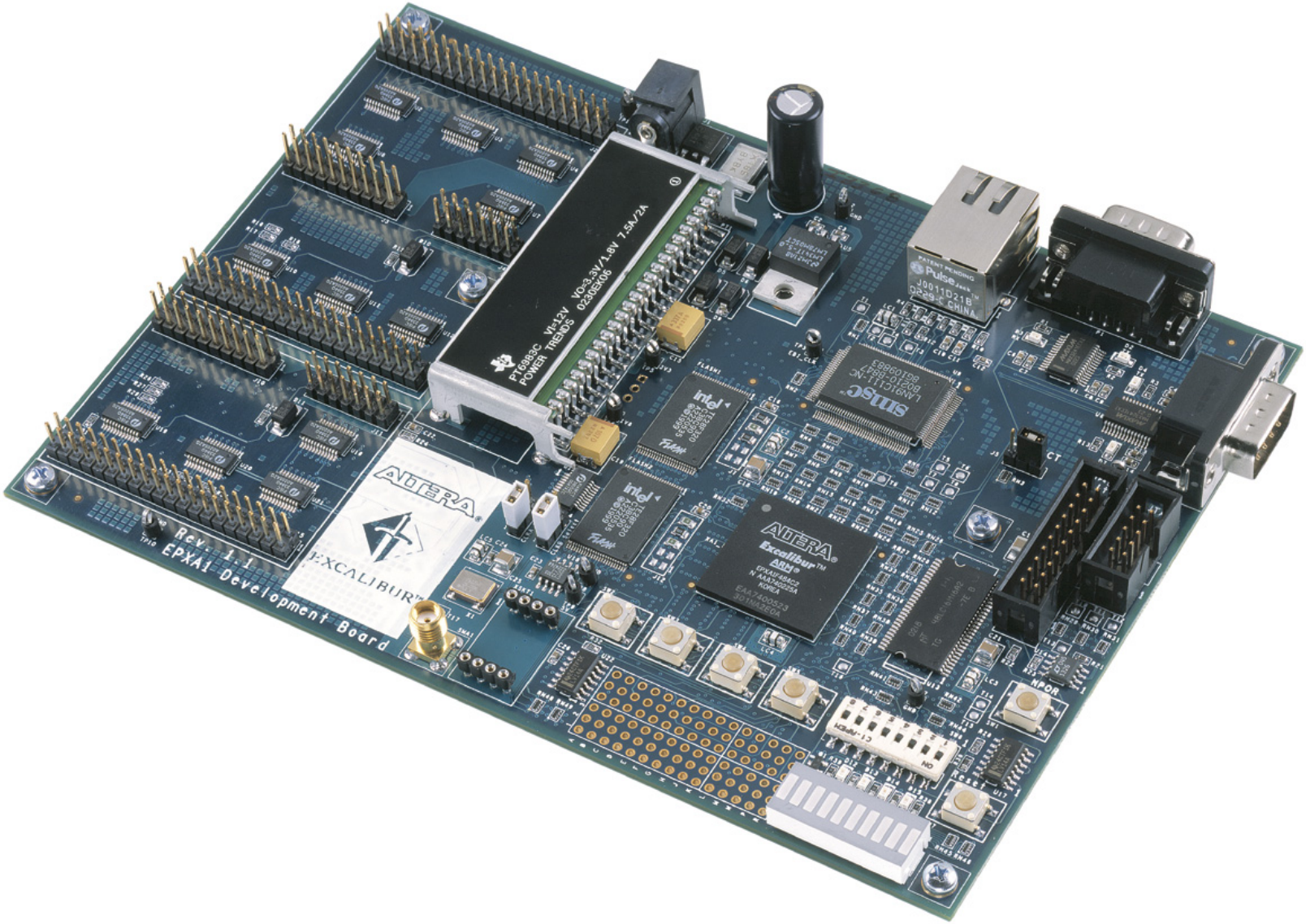
Fixed point: optimal position

- The optimal position of the radix point must be chosen considering precision and range of data.
- The goal is to minimize error: overflow must be avoided and precision maximized.
- Position of the binary point may vary along the encoding process.
- Simulations must be carried out to determine the optimal position of each binary point.

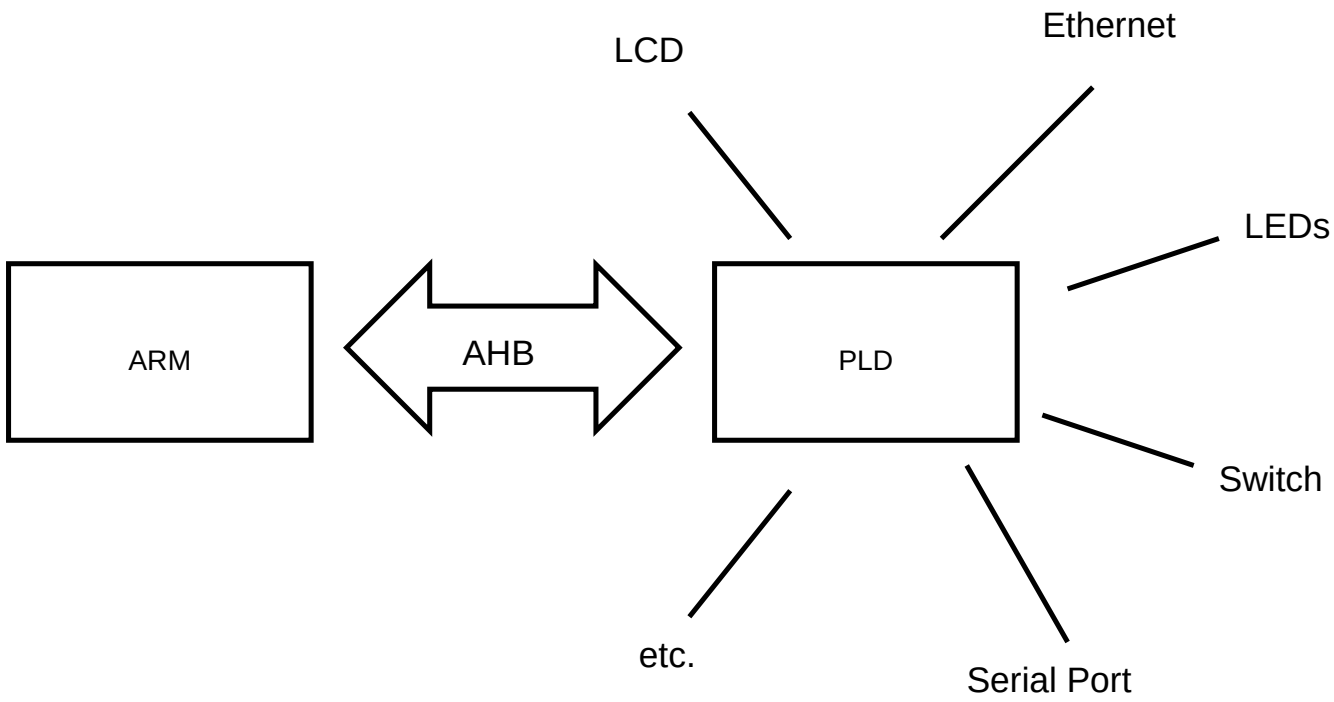
Development Toolset: development kit

- EPXA1 development board:
  - EPXA1F484C device
    - ARM922T 32-bit RISC microprocessor
    - 100 k gates APEX 20KE FPGA
  - 32 Mo RAM
  - 8 Mo flash
  - 100 Mbit Ethernet
  - JTAG header
  - LCD display
- Quartus II software
  - GNUpro compiler

Development Toolset: development board

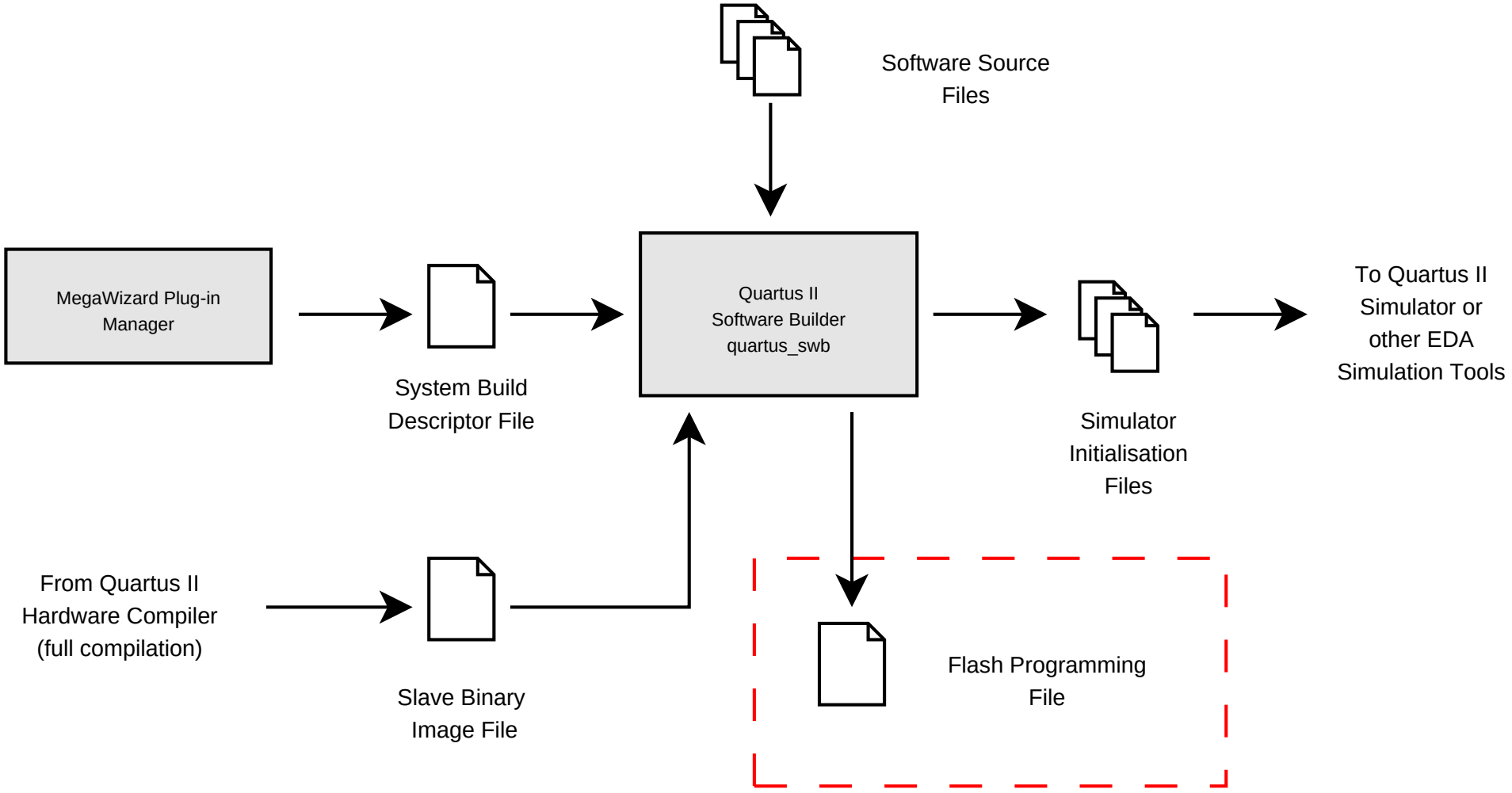


Development Toolset: architecture

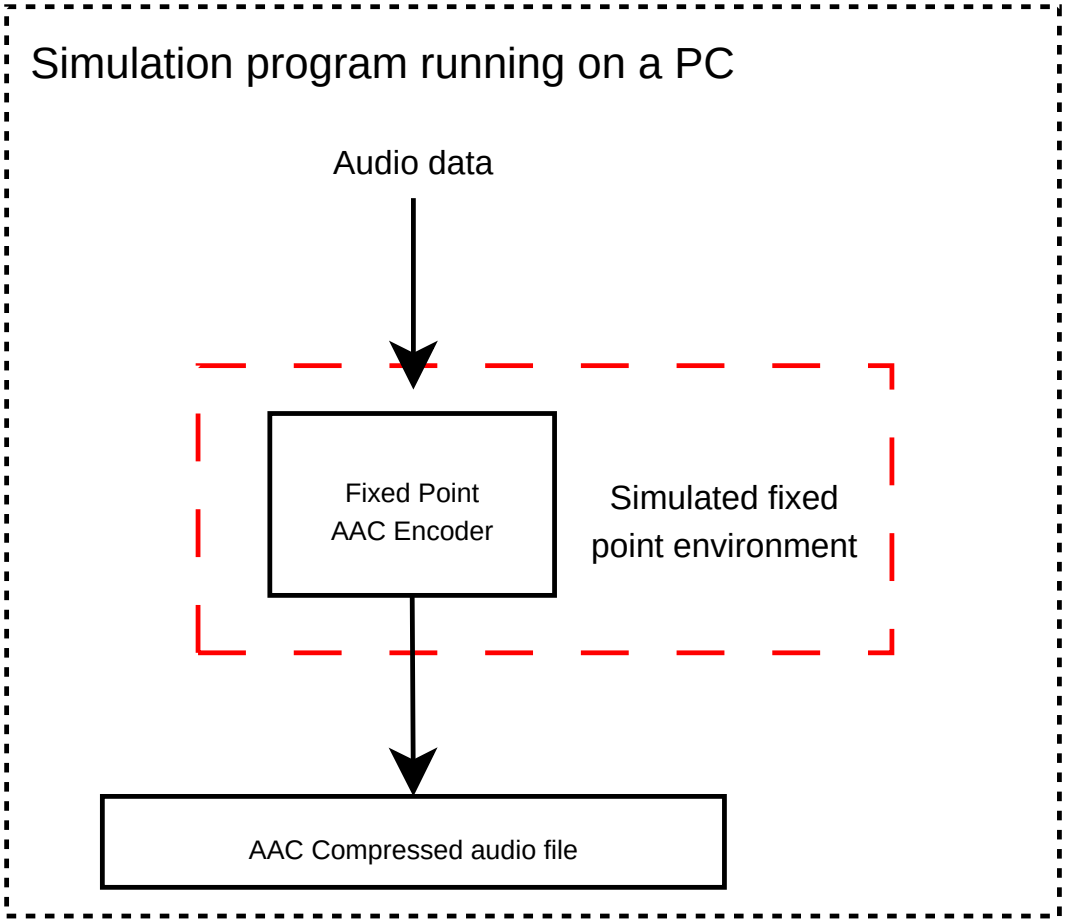


- The PLD is the only interface of the ARM processor.
- We must configure the FPGA.

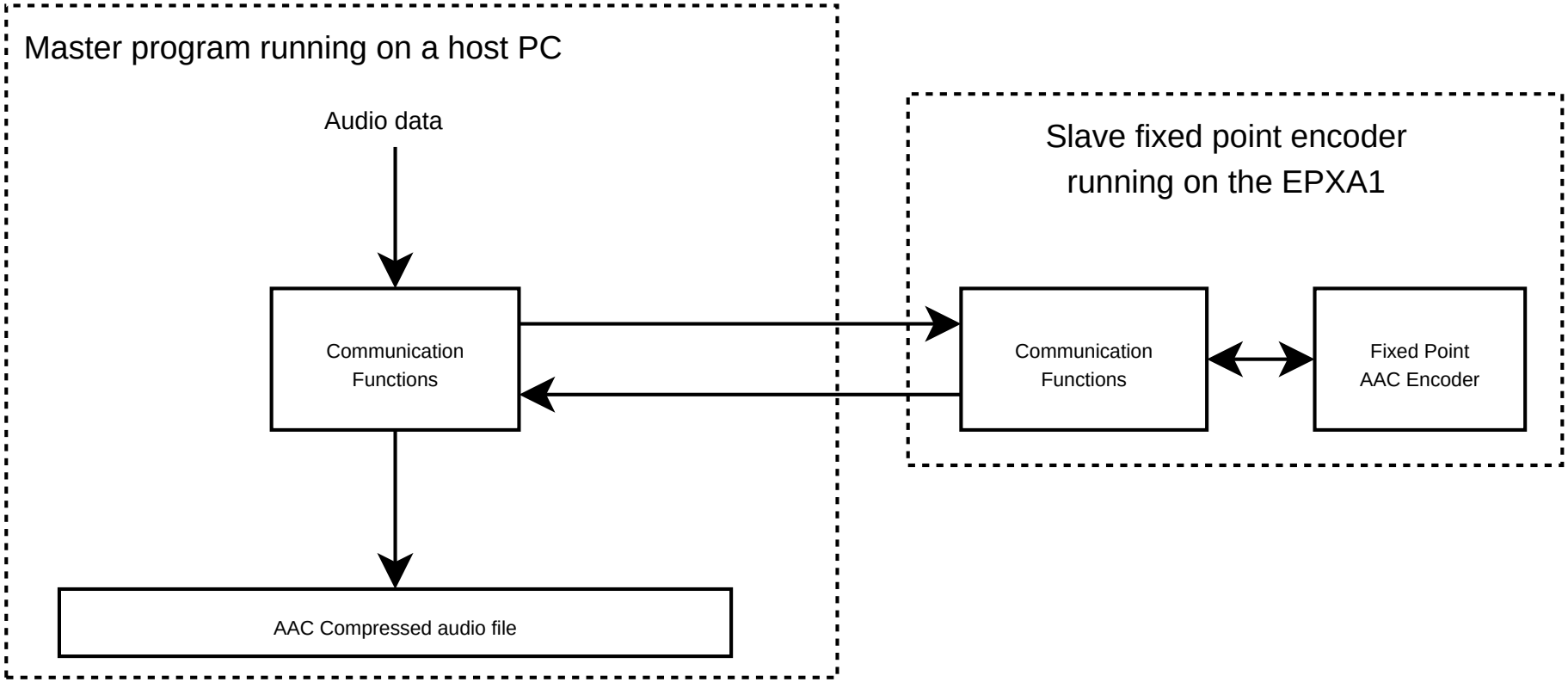
Development Toolset: flash programming



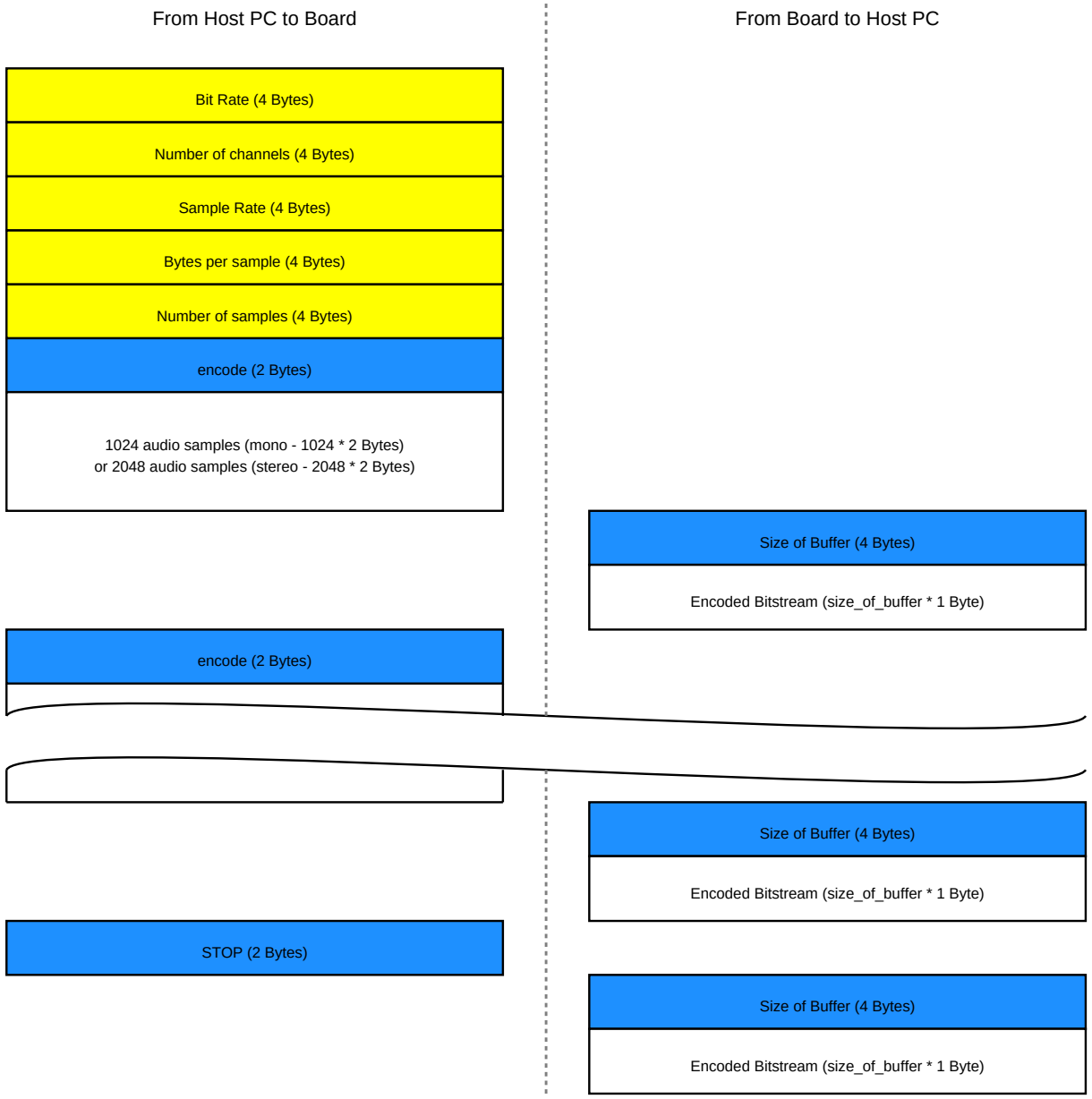
Implementation: simulation files



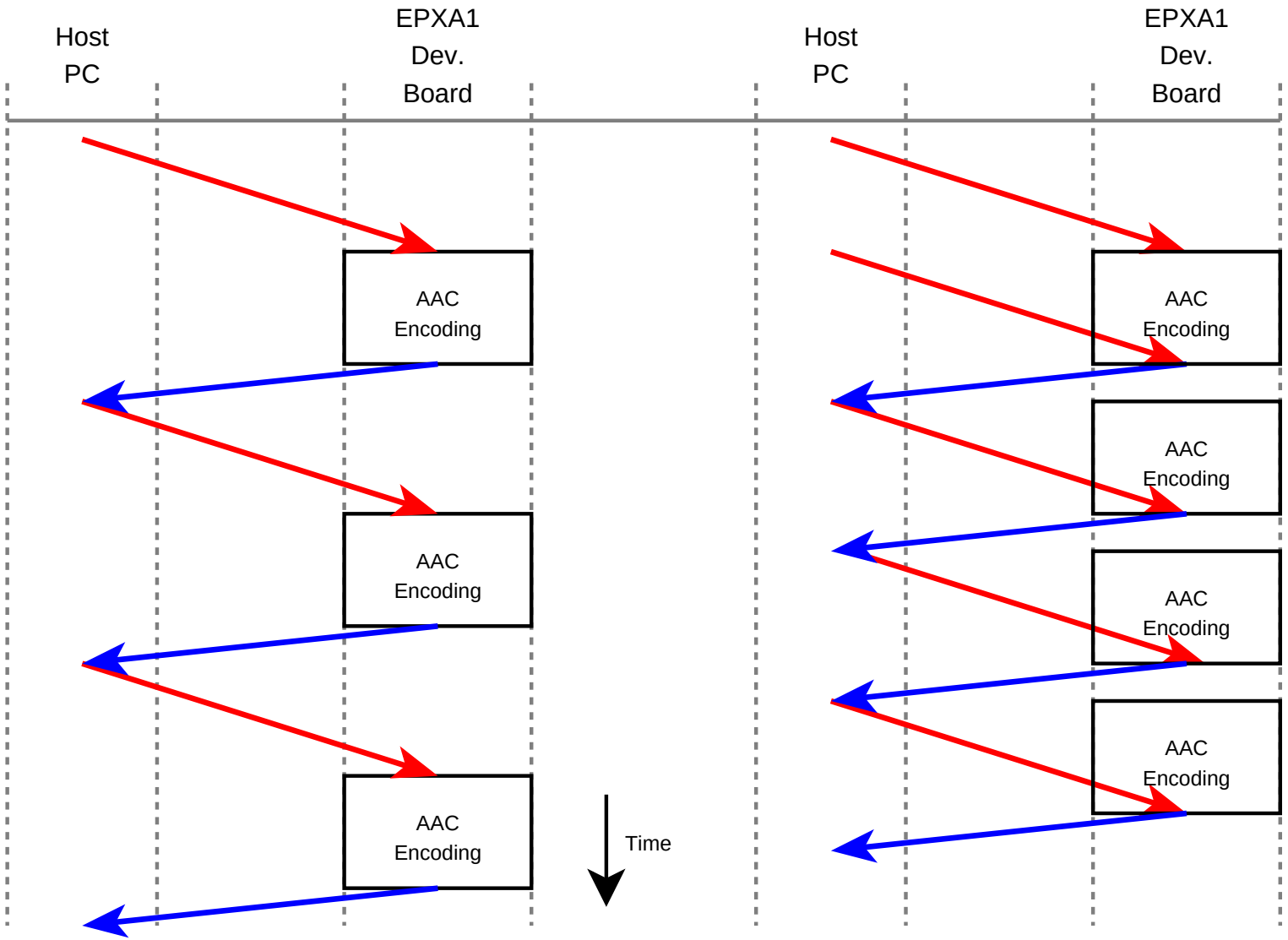
Implementation: master slave architecture



Implementation: communication sequence



Implementation: naive vs overlapped protocol



Results: communication vs computations

- Naive implementation

			Proportion	Real Time
Communications	8.63 s	0.411 s/frame	48.6 %	$\times$ 0.056
Computations	8.88 s	0.434 s/frame	51.4 %	$\times$ 0.053
Total	17.51 s	0.845 s/frame	100 %	$\times$ 0.027

- Communication overlapped with (faster) computations

			Proportion	Real Time
Communications	140 s	0.392 s/frame	100 %	$\times$ 0.059
Computations	93 s	0.260 s/frame	66.3 %	$\times$ 0.089
Total	140 s	0.392 s/frame	100 %	$\times$ 0.059

Results: overlapped and high speed mult.

- Encoding time for the communication overlapped with computations encoder (high speed multiplication) at 128 kbit/s.

		Encoding Time		Real Time
86.1 ko	0.48 s	8.30 s	0.415 s/frame	$\times$ 0.058
1.41 Mo	8.45 s	139.3 s	0.385 s/frame	$\times$ 0.063
10.9 Mo	1 min 04 s	1065 s	0.384 s/frame	$\times$ 0.060

Conclusion

- The encoder works as expected
- Communications on the serial port, even if slower the Ethernet, are sufficient to manage the encoding of a complete file