

# Digital Signal Processing with the Blackfin BF533

## Lab. Session 1

### 1 Blackfin and the EZ-KIT Lite

In these lab session we will be using the BF533 EZ-KIT Lite board featuring a BF533 Blackfin DSP (processor) but also (among other interfaces):

#### Audio codec

- it is the AD1836 audio codec
- it has two stereo inputs and three stereo outputs
- it is connected to the Blackfin via the SPORT0 port
- it is configured via the SPI port
- is capable of transferring data to the audio codec in:
  - TDM - Time division multiplexed mode
  - $I^2S$  - Inter-IC Sound mode

#### Video encoder/decoder

- the ADV7171 is the video encoder
- the ADV7183 is the video decoder
- it is connected to the Blackfin via the PPIO port

**Note.**  $I^2S$ , also known as Inter-IC Sound, Integrated Interchip Sound, or IIS, is an electrical serial bus interface standard used for connecting digital audio devices together. It is most commonly used to carry PCM information between the CD transport and the DAC in a CD player. The I2S bus separates clock and data signals, resulting in a very low jitter connection.

See the “how to” slides for chronograms and more details.

### 2 The Visual DSP++ environment

The programming of the Blackfin processor will be done via the Visual DSP++ environment. See the “how to” slides for a step to step introduction to this environment.

### 3 Tasks to be done in this lab

#### 3.1 Open the dotprod.c project

The dotprod.c project comprises two C language source files:

- dotprod.c and
- dotprod\_main.c

From the Project menu, choose **Build Project**.

- Visual DSP++ first checks and updates the project dependencies and then
- Builds the project by using the project source files.
- Output window displays status messages.

If you have an error, fix it! Upon successful build hit **Run** (F5).

- the result is displayed in the output window
- step by step simulation possible (press F11 or Debug → Step Into)

When done, close the project (File → Close → Project dotprodc.dpj)

### 3.2 Open the dotprod.asm project

The point here is mixing assembler and C and using the linker. See the “how to” slides for more details.

### 3.3 FIR Filter example

The point with this project is to see how to implement a FIR filter and displaying the input and output data. See the “how to” slides for more details.

**Understand how all projects work.**

### 3.4 Questions

Question 1: How do you define a scalar (dot) product (in mathematics)? Explain the function `a_dot_b` from the project `dotprod.c`. How is the dot product implemented in C?

Question 2: You have implemented a FIR filter as your last task. What type of filter is your FIR finally? Explain its effect on the signal `IN`. Could you have predicted beforehand the shape of the signal `OUT`? If yes, how?