XCorr IP core

Stefan Kull, Roy Seitz, Marco Zollinger

December 2, 2016

1 Overview

This Document describes the functionality and usage of the XCorr IP core. It is supplied as a complete, ready to use IP core but without high level software support. This is due to the project extent which is higher than expected. Low level software access is provided but must be used carefully. High level software support will be added in a future release.

The project Soundloc contains three microphones that deliver a $\Sigma\Delta$-modulated bitstream. This is processed by another IP core SDM_DECIMATOR, that delivers signed microphone data and an interrupt that indicates new values.

This IP core then calculates the cross-correlation between the three microphone signals. To do this efficiently and in real-time, the correlation is calculated iteratively, using fast block RAM and DSP slices. Detailed information to each stage is provided in the following sections.

2 Parameter description

The IP core can be configured at compile time by several parameters, listed in table 1. The number of stored samples is calculated according to equation 1. The calculated number of Taus is derived from equation 2. Tau ranges from Tau_{min} to Tau_{max}, given in equation 3 and 4

$$N_{Sample} = 2^{D\_SAMPLE\_ADDR\_WIDTH} - 1 \tag{1}$$

$$N_{Tau} = 2^{D\_TAU\_ADDR\_WIDTH} - 2 \tag{2}$$

$$Tau_{min} = -2^{D\_TAU\_ADDR\_WIDTH-1} + 1 \tag{3}$$

$$Tau_{max} = 2^{D\_TAU\_ADDR\_WIDTH-1} + 1 \tag{4}$$

Table 1: Parameters for the XCorr

Parameter	Default	Range	Type	Description
D_WIDTH	16	1...18	integer	Width of incoming microphone data
D_SAMPLE_ADDR_WIDTH	12	6...16	integer	Address width for stored microphone samples
D_TAU_ADDR_WIDTH	6	1...6	integer	Address width for calculated Taus

3 Register description

There is a register to clear the internal RAM and one register for each correlation and Tau. Each can be accessed directly by their address as described in section 5.

To clear the internal RAM, Address 0 needs to be set to 0x1 for at least N_{Sample} clock cycles. Because only one internal RAM address can be set at once, asserting this bit for less than N_{Sample} cycles results in corrupted data.

The Taus can be read by using their representation in two's complement and the correct S from table 2 to calculate their addresses according to equation 5. The multiplication with four comes from the byte wise addressable memory and four byte width correlation data.

$$ADDR_{Tau} = S \mid (\text{0x0FC} \,\&\, (4 \times Tau)) \tag{5}$$

Table 2: Tau address

S	Cross-correlation between
0x200	Microphone 2 and 3
0x300	Microphone 2 and 1

4 Cross-correlation

This block calculates the cross-correlation xcorr01 and xcorr02. The first index indicates the reference microphone, which is fixed to mic0 (microphone 2 on PCB). The correlation is recalculated each time a new value is available. The recalculation takes $N_{Tau} + 4$ clock cycles. An interrupt is asserted each time the cross-correlation has been recalculated.

4.1 Implementation

The cross-correlation of two discrete signals is defined per equation 6. Since only N_{Sample} values are stored and available, the summation simplifies to equation 7 for positive Taus and to 8 for negative Taus.

$$X_{ab}(\tau) := \sum_{n=-\infty}^{\infty} a\,[n]\, b\,[n+\tau] \tag{6}$$

$$= \sum_{n=0}^{N_{Sample}-\tau-2} a\,[n]\, b\,[n+\tau]\, \forall \tau \geq 0 \tag{7}$$

$$= \sum_{n=0}^{N_{Sample}+\tau-2} a\,[n-\tau]\, b\,[n]\, \forall \tau < 0 \tag{8}$$

$$\tag{9}$$

Each time new data is available, all stored values are shifted by one storage position. Therefore not the whole cross-correlation has to be calculated each time. Only the newest value pair must be added and the oldest pair subtracted, resulting in two MAC-instructions per Tau and correlation.

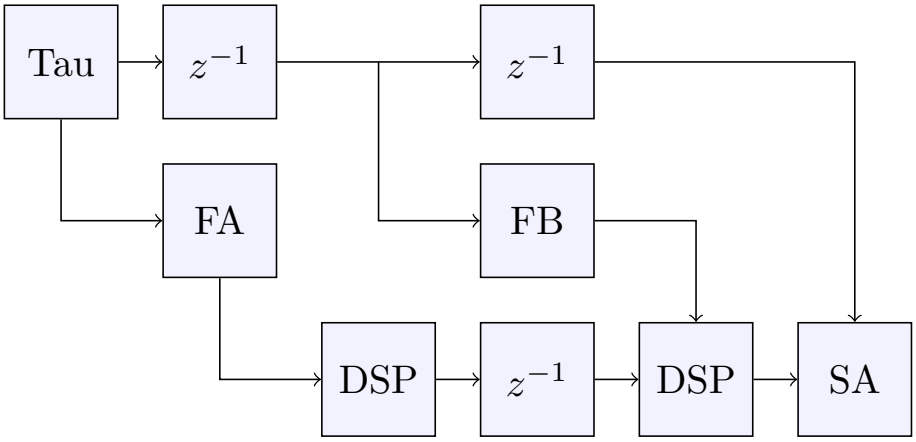

Figure 1: Block diagram of pipelined calculation of cross-correlation

To further improve performance, the calculation is pipelined. Figure 1 shows the data flow during recalculation.

- Block Tau generates the actual Tau that is to be calculated.
- FA denotes fetch on RAM port A. It reads the corresponding cross-correlation and the (old) microphone values that are to multiply and subtract.
- The first DSP performs $P = -(AB) + C$, with A and B connected to the microphone values from FA and C to the cross-correlation value.
- FB fetches the (new) microphone values that are to multiply and add.
- The second DSP calculates $P = (AB) + C$, with the microphone values from FB and the prepared cross-correlation from the first DSP.
- Finally, SA denotes store on port A, which performs a write-back of the new calculated cross-correlation value.

Flip-flops are needed to store and shift the Taus corresponding to the single stages, as each block RAM fetch and store needs one clock cycle. Consequently the store command comes always three clock cycles after load. Therefore neither collision nor simultaneous read / write is possible and no collision-detection is implemented.

5 Driver

For now, the core can only be used by addressing the registers directly. High-level access will be added in a future release. Two low-level functions are available. These functions are implemented as macros and should be used carefully. Care must be taken to not confuse the microphones. On the PCB they are numbered from 1 to 3 but in logic the numbering goes from 0 to 2.

- XCORR_mWriteReg(BaseAddress, RegOffset, Data)
  Write data to the specified Register.
- XCORR_mReadReg(BaseAddress, RegOffset)
  Read the content of the specified register.

6 Test and verification

The core can easily be simulated by using the AXI traffic generator IP from Xilinx and feeding some well known values to the inputs. No additional simulation files are available. Software verification is not supported.

7 Compatibility and License

The core is tested under Vivado version 2016.2 and on Artix7 and Zynq7 FPGA. The core does use hardware specific resources. It is therefore not guaranteed to run on other FPGAs. Since only 4 DSP slices and block RAM is used, it should run on nearly every FPGA with enough block RAM available. However, this is not tested and may require changes to the core hdl files. The core is supplied under no license or copyright but is the intellectual property of the authors.