ISBN: 978-3-95450-241-7 ISSN: 2673-5350 doi:10.18429/JACoW-IBIC2022-MOP14

DESIGN AND IMPLEMENTATION OF AN FPGA-BASED DIGITAL PROCESSOR FOR BPM APPLICATIONS

M. Colja*, S. Carrato, University of Trieste, Trieste, Italy
G. Brajnik, R. De Monte, Elettra-Sincrotrone Trieste, Trieste, Italy

Abstract

Digital processing systems have been proven to often outperform analog elaboration. Indeed, thanks to high-density DSPs and FPGAs, operations in digital domain give results that are impossible to achieve in other ways. On the other side, dealing with this great performance and flexibility is not always straightforward: the processing chain needs to be accurately planned to reach the desired goals, avoiding erratic behaviours in the digital domain. In this paper, we focus on the design and implementation of an FPGA-based digital processor that will be used in the electron beam position monitors of Elettra 2.0. After digitizing the 500 MHz beam signals from the pickups, the system executes a digital down conversion, followed by several filtering and demodulating stages, in order to have a selectable data rate that is suitable for both diagnostics and feedback. The position calculation is also performed in FPGA as well, with the well-known difference-over-sum algorithm. According to results provided by a fixed-point simulation, the overall system has been implemented in an Intel Arria 10 FPGA, demonstrating the correct design functionality that meets the specified requirements.

INTRODUCTION

Digital signal processing has become a standard approach for realizing beam position monitor systems (BPM), thanks to the availability of high-speed, high-performance ADCs and FPGAs [1]. In synchrotrons, it is well known that information about beam position lies in the amplitude of the signal detected by the pickups: due to the typical beam fill patterns used in those machines and after a mandatory analog signal conditioning, it can be seen as a sine with a frequency equal to RF frequency of the accelerator. So, to extract its amplitude after digitization, there is the need for a digital receiver and demodulator that can do it properly for extracting its amplitude after digitization. Then, position calculation algorithm like difference-over-sum can be applied on demodulated amplitudes, converting them into a real beam position, related to vacuum chamber dimensions using a suitable scale factor.

This architecture will be used for the new eBPM system of Elettra 2.0, the low-emittance upgrade of the current machine. The overall modular system and some of its components (e.g. the pilot tone front end) have been already presented [2]; but in this paper we will focus on simulation and FPGA implementation of the complete digital processing chain, from ADC data to calculated position.

* matija.colja@elettra.eu

DIGITAL PLATFORM AND A/D CONVERSION

The digital platform (Fig. 1) has already been presented in previous papers [2,3]: analog signals from the front end are digitized by an FMC card with four LTC2107, 16-bit ADCs running at 150.38 MS/s, that is 130 times the revolution frequency and phase-locked with it.

Thanks to bandpass sampling technique [4] there is no need for an analog downconversion stage: the 499.654 MHz signal is seen by the ADCs as a 48.514 MHz sine, since it is folded back in the Nyquist zone corresponding to three times the sampling frequency. Obviously, this approach needs a high-performance, low-jitter clocking stage and an analog bandpass filter to reduce out-of-band noise.

Then, ADC data are processed by an Intel Arria 10 GX FPGA: the high amount of logic gates and DSP resources render it possible to realize in digital the complete processing chain. Connections with external devices are assured by internal transceivers attached to SFP+ modules capable to go at a speed up to 10 Gb/s.

Figure 1: Digital platform.

ISBN: 978-3-95450-241-7 ISSN: 2673-5350 doi:10.18429/JACoW-IBIC2022-MOP14

DIGITAL RECEIVER

Moving to the digital domain, we chose a direct conversion digital receiver (or digital down converter, DDC) paired with a quadrature amplitude demodulator. The input signal from the ADCs is shifted to the baseband by mixing (multiplying) it with a sine and a cosine generated by a numerical controlled oscillator (NCO). A low-pass CIC decimator filter [5] suppresses the high frequency components and reduces the data rate to turn-by-turn (TbT, 1.156 MHz) frequency. Since CIC filters have a magnitude response that causes a droop in the passband region, a compensation FIR corrects it. To avoid dealing with complex numbers, there are separate I and Q paths so that it is possible to reuse the same filters for both components.

Then, amplitude demodulation is performed by squaring I and Q, summing them and taking the square root of the result. Another CIC decimator reduces data rate to 10 kHz for global feedback (FA data output): compensation is not required since after amplitude demodulation, signal is DC-centred. User can choose to retrieve amplitudes of the four channels at TbT or FA data rate.

All the processing is done using integers in place of floating point numbers. More details on the implementation are listed below, looking at Fig. 2, where only one channel is depicted for sake of simplicity:

1. ADC 16-bit signed data (two's complement) running at 150.38 MHz;
2. multipliers: digital version of mixers. They perform frequency translation of the input signal near to the DC and are implemented using 18x18 bit hardware multipliers in DSP cells [6];
3. NCO (digital version of local oscillator): it generates a sine and cosine for I and Q processing with sufficient precision (18 bit), higher than ADC data and running at sampling frequency. It is implemented using Intel IP core [7];
4. CIC decimator filter: it suppresses the high frequency component generated by multiplication and reduces data rate to TbT frequency. The rate change factor is 130, the number of stages is 6 and the differential delay is equal to 1. It is implemented using Intel IP core [8], input and output are 32-bit wide. Bits are reduced using Hogenauer pruning;
5. FIR compensator: it corrects CIC magnitude response, and has been generated with a MATLAB appropriate routine. The 19 coefficients are symmetrical, with a length of 17 bits. It is implemented with Intel IP core [9], input and output are 32-bit wide;
6. square operation: it is implemented directly in DSP cells, with 32-bit input and 64-bit output;
7. sum of I and Q squares: it consists in a simple 64-bit adder;
8. square root: it is implemented using Intel IP integer arithmetic core [10], 64-bit input, 32-bit output;
9. second CIC decimator for changing data rate to 10 kHz as required by feedback. The rate change factor is 116 with 6 stages, input and output are 32-bit wide.

After writing all the code in Verilog language, the design has been successfully compiled in Quartus Prime. Table 1 reports FPGA resource usage after synthesis and fitting in terms of Adaptive Logic Modules (ALMs), dedicated logic registers (REGs), 20 kbits memory blocks (M20K), block memory bits (BMBs), DSP blocks (DSP) for every Verilog entity: IQ-amp includes squares, summing and square root stages.

Table 1: FPGA Resource Usage

Entity	ALMs	REGs	M20k	BMB	DSP
Mult-18x18	0	0	0	0	1
NCO	64	224	2	27648	2
CIC-130	358	907	3	1092	0
FIR	244	360	2	8192	2
IQ-amp	600	1117	0	0	6
CIC-116	357	943	3	1092	0

It has to be noted that for the complete digital receiver (4 channels), we need 8 multipliers, 8 CICs (TbT rate), one NCO, one FIR (can be reused for all channels and IQ components thanks to time domain multiplexing) and 4 IQ-amp entities. Additional 4 CICs (FA rate) are included for the 10 kHz data stream.

Pilot-tone compensation feature [11] can be enabled implementing a second digital receiver with a dedicated NCO tuned to pilot frequency, doubling the amount of resources needed. However, this topic is beyond the scope of this paper.

POSITION CALCULATION

After demodulating the four amplitudes, the position is calculated with the well-known difference-over-sum (DoS) algorithm, widely used in BPM applications:

$$x = K_x \frac{(A+D)-(B+C)}{A+B+C+D}, \quad y = K_y \frac{(A+B)-(C+D)}{A+B+C+D} \tag{1}$$

where K_x and K_y are the scale factors related to vacuum chamber dimensions (20 mm for Elettra), and A, B, C, D are the demodulated amplitudes. Calculations inside FPGA are done scaling accordingly the coefficients in order to avoid a non-integer quotient and for obtaining x and y as 32-bit signed integers expressed in nanometers (LSB is 1 nm), mainly for backward compatibility with the existing Elettra control system.

For an effective debug, user can choose data source and data rate nearly at every processing stage. High amount of samples can be stored in 8 GB an external DDR3 RAM memory, otherwise real-time data stream at 10 kHz (FA) or

ISBN: 978-3-95450-241-7 ISSN: 2673-5350 doi:10.18429/JACoW-IBIC2022-MOP14

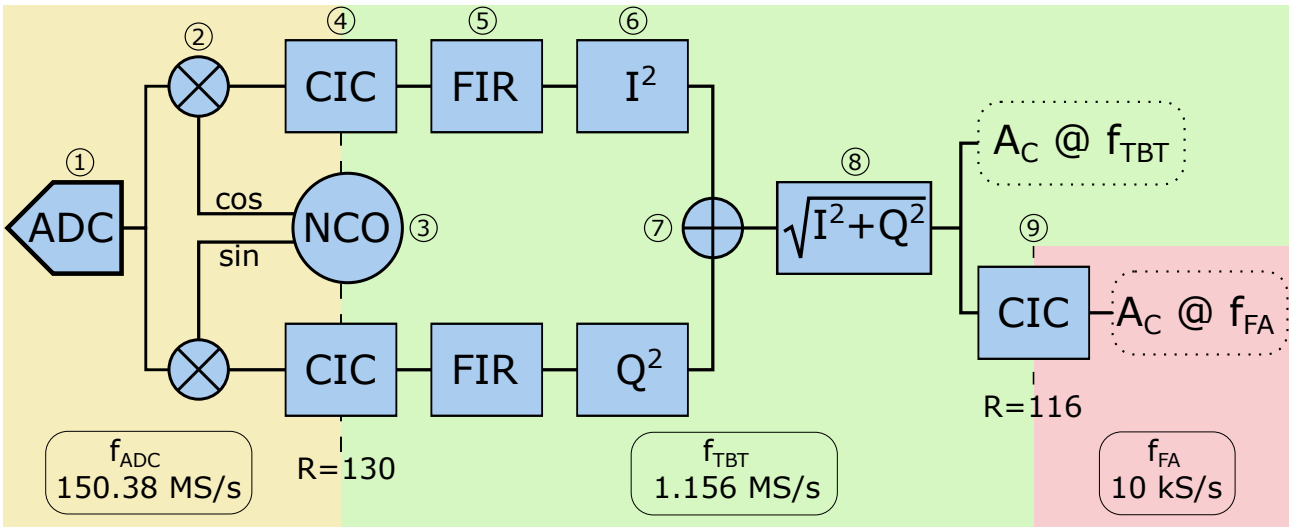

Figure 2: Block diagram of digital receiver: only channel A is shown.

at TbT data rate can be sent to external systems through multiple Ethernet connections (1 Gb/s or 10 Gb/s). The overall simplified FPGA block diagram can be seen in Fig. 3.

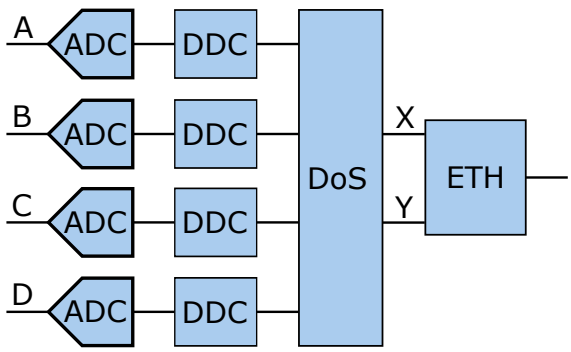

Figure 3: Simplified block diagram of whole processing chain.

SIMULATION AND RESULTS

In order to validate the design, the entire processing chain has been simulated in MATLAB using the fixed-point designer toolbox, thus having the same behaviour of FPGA logic. Every step of the chain has been compared with theoretical results, based mainly on signal-to-noise ratio (SNR) measurements. Decimation and filtering increases SNR since bandwidth is reduced, so we can express the quantity called decimation gain in decibels:

$$\mathrm{DG_{dB}} = 10\log_{10}(R), \tag{2}$$

where R is the rate change factor. In this way, the process gain obtained at TbT data rate is about 21 dB ($R = 130$), while at FA data rate is about 42 dB ($R = 130 \cdot 116$).

Nevertheless, there is a relation between SNR of demodulated amplitudes and standard deviation of calculated positions [12]:

$$\sigma_x = \frac{K_x}{2} \cdot \left(\sqrt{\mathrm{SNR}}\right)^{-1}, \quad \sigma_y = \frac{K_y}{2} \cdot \left(\sqrt{\mathrm{SNR}}\right)^{-1}. \tag{3}$$

These expressions allow us to have a direct relation between the SNR at ADCs inputs and the resulting noise on positions:

$$\sigma_x = \frac{K_x}{2} \cdot \left(\sqrt{10^{\frac{\mathrm{SNR_{dB}}}{10}}}\right)^{-1}, \quad \sigma_y = \frac{K_y}{2} \cdot \left(\sqrt{10^{\frac{\mathrm{SNR_{dB}}}{10}}}\right)^{-1}, \tag{4}$$

where $\mathrm{SNR_{dB}}$ is the sum of input SNR and processing gain ($\mathrm{SNR_{in,dB}} + \mathrm{DG_{dB}}$).

Simulations

First of all, we decided to check the entire MATLAB chain, generating a sinusoidal signal with an additive white gaussian noise (different realizations of the statistic process for every channel) to emulate a stable and centred beam. Checks on effective and expected SNR are performed at every stage of the simulation, to understand if quantization or processing errors could corrupt it. Varying input SNR, resulting noise on position follows in a good way Eq. (3), meaning that the entire processing does not limit the resolution up to wanted specifications, at least 200 nm at 10 kHz data rate for Elettra 2.0 for 60 dB of input SNR (Fig. 4).

Then, as an intermediate step, we moved to real data: a 499.654 MHz signal from a RF generator, splitted in four and fed to ADCs, has been used to emulate a stable beam. Raw data from ADCs have been acquired and used in MATLAB simulation. Standard deviation on position saturates at about 90 nm, even if SNR is increased (Fig. 5). This means that there is a noise contribution that increases with signal amplitude and dominates thermal noise after a break even point (located at about 9000 ADC bins, corresponding to −3 dBm at ADC input). ADC full scale is about at +3 dBm, so there is a range of 11 dB where there is no improvement. Since

ISBN: 978-3-95450-241-7 ISSN: 2673-5350 doi:10.18429/JACoW-IBIC2022-MOP14

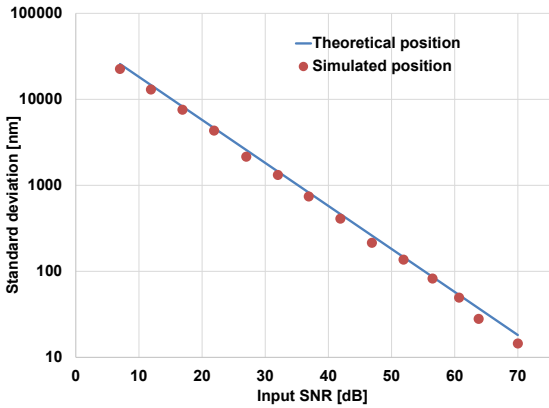

Figure 4: Standard deviation versus input SNR at 10 kHz data rate of theoretical and simulated positions.

raw data are involved, ADCs are the most likely suspects for causing this: further investigations are ongoing, mainly related to jitter of ADC sample-and-hold stage; the absolute jitter of sampling clock is not involved because, by simultaneously affecting all four amplitudes, can be compensated by the difference-over-sum algorithm, while the observed effect is related to a difference between the channels.

The same behaviour is observed moving to full FPGA processing: in that case all the calculations are done inside the unit, logging only the positions, which confirms the correct operation of the overall processing chain.

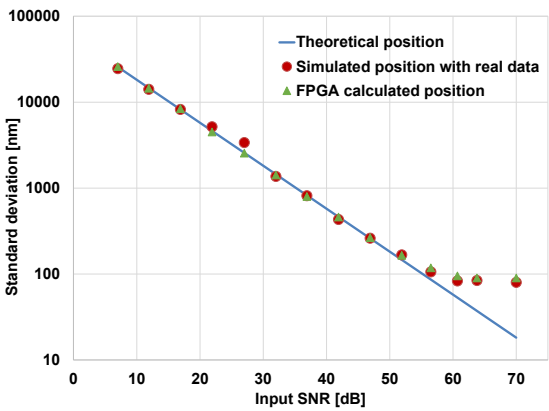

Figure 5: Standard deviation versus input SNR at 10 kHz data rate of theoretical and positions from real data.

CONCLUSION

A complete digital eBPM processor has been successfully implemented in an FPGA: both simulations and experimental data confirm the validity of all processing stages with good agreement with theoretical results. Using real data from ADCs sets a limit on higher achievable resolution: this is most probably due to jitter issues on ADC's sample-and-hold stage, since this effect is correlated with amplitude of input signal. Tests are underway to further investigate this behaviour and to propose some solutions.

REFERENCES

[1] M. Wendt, "BPM Systems: A brief Introduction to Beam Position Monitoring", in *Proc. CAS Beam Instrumentation 2018*, Tuusula, Finland, Jun. 2018, pp. 355-393. doi:10.48550/arXiv.2005.14081

[2] G. Brajnik, S. Cleva, R. De Monte, and D. Giuressi, "A Common Diagnostic Platform for Elettra 2.0 and FERMI", in *Proc. IBIC'19*, Malmö, Sweden, Sep. 2019, pp. 280–282. doi:10.18429/JACoW-IBIC2019-TUPP003

[3] G. Brajnik, M. Cargnelutti, R. De Monte, P. Leban, P. Paglovec, and B. Repič, "Current Status of Elettra 2.0 eBPM System", in *Proc. IBIC'21*, Pohang, Rep. of Korea, May 2021, pp. 71–74. doi:10.18429/JACoW-IBIC2021-MOPP16

[4] R. G. Vaughan, N. L. Scott and D. R. White, "The theory of bandpass sampling", *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 1973-1984, Sep. 1991. doi:10.1109/78.134430

[5] E. Hogenauer, "An economical class of digital filters for decimation and interpolation", *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, no.2, pp. 155–162, Apr. 1981. doi:10.1109/TASSP.1981.1163535

[6] Intel Arria 10 Native Fixed Point DSP IP Core User Guide, https://www.intel.com/content/www/us/en/docs/programmable/683583/current/intel-arria-native-fixed-point-dsp-ip.html

[7] Intel NCO IP Core: User Guide, https://www.intel.com/content/www/us/en/docs/programmable/683406/17-1/about-the-nco-ip-core.html

[8] CIC Intel FPGA IP: User Guide, https://www.intel.com/content/www/us/en/docs/programmable/683246/19-3-19-3/about-the-cic.html

[9] Intel FIR II IP Core: User Guide, https://www.intel.com/content/www/us/en/docs/programmable/683208/17-1/about-the-fir-ii-ip-core.html

[10] Intel FPGA Integer Arithmetic IP Cores User Guide, https://www.intel.com/content/www/us/en/docs/programmable/683490/20-3/altsqrt-integer-square-root-ip-core.html

[11] G. Brajnik, S. Bassanese, G. Cautero, S. Cleva, and R. De Monte, "Integration of a Pilot-Tone Based BPM System Within the Global Orbit Feedback Environment of Elettra", in *Proc. IBIC'18*, Shanghai, China, Sep. 2018, pp. 190–195. doi:10.18429/JACoW-IBIC2018-TUOC01

[12] G. Brajnik, S. Carrato, S. Bassanese, G. Cautero, and R. De Monte, "Pilot tone as a key to improving the spatial resolution of eBPMs", *AIP Conf. Proc.*, vol. 1741, p. 020013, 2016. doi:10.1063/1.4952792