A REAL-TIME FPGA BASED ALGORITHM FOR THE COMBINATION OF BEAM LOSS ACQUISITION METHODS USED FOR MEASUREMENT DYNAMIC RANGE EXPANSION

M. Kwiatkowski, C. Zamantzas, M. Alsdorf, B. Dehning, W. Vigano,
CERN, Geneva, Switzerland

Abstract

The aim of the Beam Loss Monitoring Dual Polarity (BLEDP) module under development at the European Organisation for Nuclear Research (CERN) is to measure and digitise with high precision the current produced by several types of beam loss detectors.

The BLEDP module consists of eight analogue channels each with a fully differential integrator and an accompanying 16 bit ADC at the output of each analogue integrator. The on-board FPGA device controls the integral periods, instructs the ADC devices to perform measurements at the end of each period and collects the measurements. In the next stage it combines the number of charge and discharge cycles accounted in the last interval together with the cycle fractions observed using the ADC samples to produce a digitised high precision value of the charges collected.

This paper describes briefly the principle of the fully differential integrator and focuses on the algorithm employed to process the digital data.

INTRODUCTION

The LHC Injectors Upgrade project was launched at CERN to provide higher intensity beam for the LHC, which will allow to increase further its luminosity. A new Beam Loss Monitoring (BLM) system is under design [1] for the monitoring of the beam losses and the machine protection.

The BLM Dual Polarity (BLEDP) module is the first stage of that system. The acquisition crate will be able to host up to 8 BLEDP modules each having 8 analogue inputs to attach various types of detectors. The BLEDP module should be able to digitise input current in the wide range from 10 pA up to 200 mA.

In specific, the range is split into two partially overlapping sub-ranges and for each of them a different measurement method is used. The current from 100 μA up to 200 mA should be measured directly by the ADC as a voltage drop on the input resistor. The current in the lower range from 10 pA to 10 mA is measured by making use of a low noise differential integrator. The BLEDP module is equipped with a Cyclone IV GX FPGA which is responsible for processing of the integrator output combined with the ADC samples to reach higher accuracy. The digitised result will be transmitted via the fibre optic link into the processing part of the system.

In the stand-alone version of the system, the data is send via a Gigabit Ethernet link into a dedicated JAVA supervision application which is used for on-line view of the measurements as well as offline data storage and analysis. More information about this version of the system and the methods employed can be found in [2].

MEASUREMENT METHODS

The analogue front-end of the BLEDP module can measure input current in a wide dynamic range. To achieve this, the input current in the lower range is measured by a Fully Differential Frequency Converter (FDFC) circuit and in the higher range by a direct Analogue to Digital Conversion (DADC) circuit. In the DADC method the ADC converter is attached to the input resistor on which the voltage drop is measured. In the FDFC method the ADC is attached to the differential output of the integrator.

The analogue switch, that selects which of the two circuits is active at any given time, is controlled by the FPGA device. A module operating in the FPGA is monitoring the data stream and depending on the measurements it receives selects the most suitable measurement method for the next period.

This paper focuses on the FDFC method which require processing of both the ADC samples and the comparator pulse stream. A simplified schematic of the FDFC analogue front-end is shown in Fig. 1.

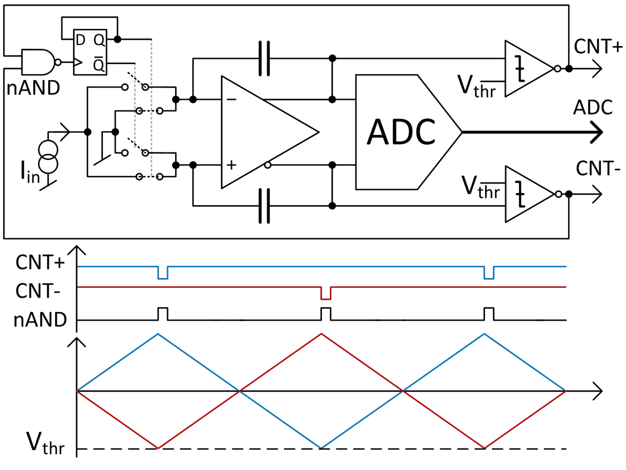

Figure 1: Fully Differential Frequency Converter.

The input current is directed by the analogue switches to the positive or negative branch of the differential integrator. The current arriving is charging the capacitor of

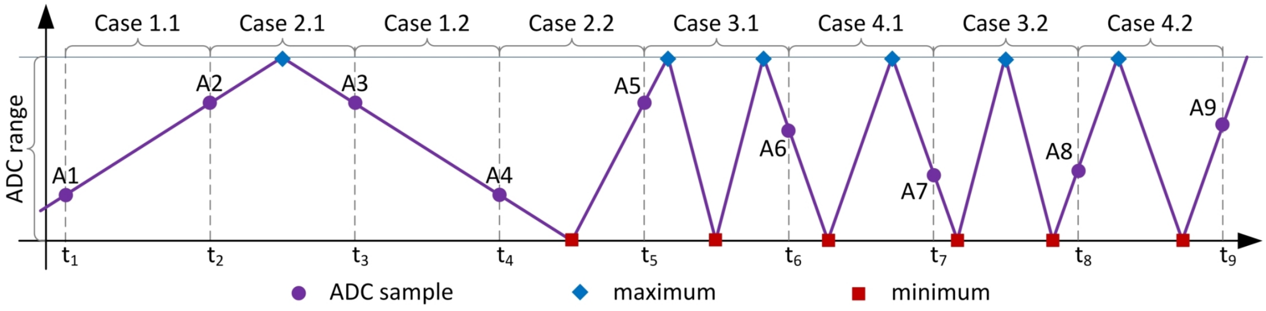

Figure 2: Fully Differential Frequency Converter data acquisition cases.

the connected branch and in the same time it is discharging the capacitor of the opposite branch. Both branches are connected to the analogue comparators which create a negative pulse as soon as the voltage reach the threshold V$_{thr}$. The V$_{thr}$ is provided by the digital potentiometer, which is also controlled by the FPGA device. Its role is to limit the voltage range on the differential input of the ADC. Thus, the V$_{thr}$ is always below the reference voltage V$_{ref}$ of the ADC. The pulses of both the comparators are combined by a NAND gate, which toggles a flip-flop. Change of the flip-flop value forces the connection of the input current to be directed in the opposite branch. More information on the analogue front-ends of both the DADC and FDFC can be found in [3].

FDFC PROCESSING ALGORITHM

The FDFC circuit's output is processed by the FPGA device. The outputs of both analogue comparators are complemented by the ADC samples to reach higher measurement precision. The ADC is triggered by the FPGA every 2 μs. Those triggers mark also the integration period, in which the measurement result is produced. The algorithm has to take into account 4 cases of the input data, each having 2 sub-cases in order to calculate the result.

FDFC Sampling Cases

All possible sampling scenarios of the FDFC signal within the integration period are presented graphically in Fig. 2. It will be shown further that a group of these cases can be handled by a generic equation after applying some optimisations. The Y axis in Fig. 2 is scaled in bits of the ADC. The ADC used in the BLEDP module converts the differential voltage to a 16 bits value in two's complements format. The maximum ADC range includes values from -32768 to 32767, but the input voltage is always limited by the digital potentiometer.

The algorithm that will be described would work correctly for any ADC range, including non symmetric with just positive values. In each integration period a digital value is produced that combines two ADC samples at its boundaries with the number of times the integrator output signal reached the minimum and maximum extremes. The calculations that could be applied to produce the result are presented in Table 1. The case numbers correspond to these shown in Fig. 2:

Table 1: Processing of the FDFC Cases

Case	Calculations
1.1	$A_{n+1} - A_n$
1.2	$A_n - A_{n+1}$
2.1	$(max - A_n) + (max - A_{n+1})$
2.2	$(A_n - min) + (A_{n+1} - min)$
3.1	$(max - A_n) + (max - A_{n+1}) + (M-1) \cdot (max - min)$
3.2	$(A_n - min) + (A_{n+1} - min) + (M-1) \cdot (max - min)$
4.1	$(A_n - min) + (max - A_{n+1}) + (M-1) \cdot (max - min)$
4.2	$(max - A_n) + (A_{n+1} - min) + (M-1) \cdot (max - min)$

- **Case 1**: there were no extremes reached by the output signal during the integration period. The result is therefore just a difference of two ADC samples. The sub-case distinguishes if the samples were acquired on the rising, i.e. sub-case 1.1, or on the falling, i.e. sub-case 1.2, slope of the signal.
- **Case 2**: there was just one extreme reached during the integration period. The sub-case takes into account if it was a maximum, i.e. sub-case 2.1, or a minimum, i.e. sub-case 2.2, extreme. The result is the sum of the two parts. The difference between the first sample A_n and the extreme, and the difference between the extreme and the second sample A_{n+1}.
- **Case 3**: there was an odd overall number of minimum and maximum extremes reached during the integration period. In this case, the first sample A_n and the second sample A_{n+1} were acquired on the opposite slopes, rising and falling, i.e. sub-case 3.1, or falling and rising, i.e. sub-case 3.2, respectively. The result is the sum of the two differences between the samples and extremes at the boundaries of the integration pe-

riod and the third component $(M-1)\cdot(max-min)$. Where M is the total number of extremes. That component adds to the result a number of slopes expressed by the ADC ranges.

- **Case 4**: there was an even overall number of minimum and maximum extremes reached during the integration period. In this case, the first sample A_n and the second sample A_{n+1} were acquired on the same slopes either falling, i.e. sub-case 4.1, or rising, i.e. sub-case 4.2. The third component is similar to the one shown in case 3.

Optimisation of the Cases

The first optimisation can be simply done by noticing that case 2 is similar to case 3, but miss the third component. The equations in the Table 1 for case 3 are valid also for case 2 due to the fact that the total number of extremes is 1 and therefore the third part of the equation is multiplied by 0, i.e. M-1. For this reason, the two cases can be merged.

The second optimisation method implies the use of the absolute value (ABS) in the calculations. With the use of the ABS operation the importance of the order in the subtraction's operands, which is expected to always have a positive result, is effectively removed. This simplifies case 1 to the following Eq. 1.

$$|A_n - A_{n+1}| \tag{1}$$

The third optimisation makes use of the fact that in cases 3 and 4 the only important information is on which slope the samples were acquired. This means that cases 2, 3 and 4 can be reduced to just one case. That case will be named 2 in the following sections and it is valid for the total number of extremes $M > 0$. The ABS operation is applied to that case as well. The first difference, i.e. A_n-extreme, is now calculated according to the Eq. 2. The first part of the alternative in the Eq. 2 should be applied if the first sample A_n was acquired on the rising slope and the second alternative otherwise.

$$|A_n - max| \vee |A_n - min| \tag{2}$$

Similarly the second difference, i.e. extreme-A_{n+1}, is expressed as the Eq. 3. The first part of the alternative in the Eq. 3 should be applied if the second sample A_{n+1} was acquired on the rising slope and the second alternative otherwise.

$$|min - A_{n+1}| \vee |max - A_{n+1}| \tag{3}$$

Third and last component of the optimised case 2 is the number of the ADC ranges reached and can be expressed by the Eq. 4 (where M=1,2,3,...).

$$(M-1)\cdot|max - min| \tag{4}$$

PSPICE AND MATLAB SIMULATION

The optimised algorithm described was written and simulated in the MATLAB environment. The input data was taken from the PSpice simulation of the analogue circuit. Some example of the results of the simulation are shown in Fig. 3.

In the PSpice simulation the FDFC circuit was modelled and 1 nA input current was injected. The PSpice data, shown in the plot at the top of the figure, was interpolated for better accuracy in the simulation and the voltage was converted to ADC values. By using this method it allows to simulate as close as possible the ADC output in the real BLEDP module. The input data was processed by the algorithm and the result of processing was scaled to the current units. That scaling takes into account the analogue parameters of the FDFC circuit like the threshold voltage, the gain and the capacitance of the integrator's capacitor. The processed data was averaged due to the low input current. The 2μs integration period is not sufficient to measure 1 pA. The averaging simulates extension of the integration time. The result of the simulation is visible at the bottom of the figure.

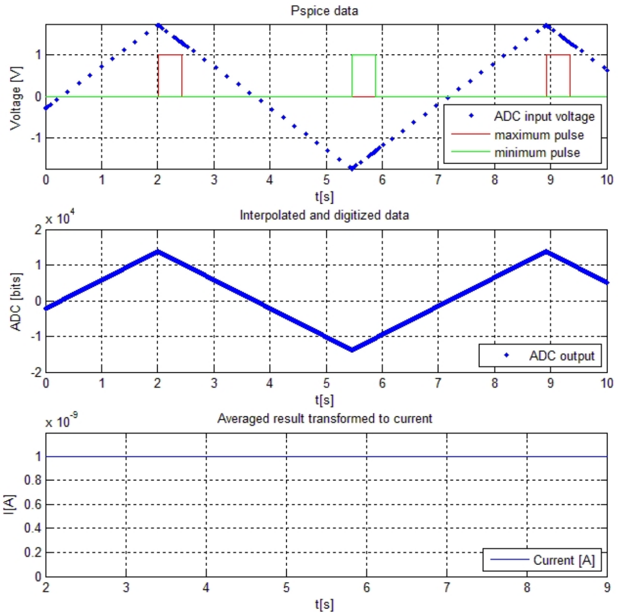

Figure 3: Examples of the PSpice and MATLAB simulation outputs.

REALISATION IN THE FPGA

This section will discuss the implementation of the simplified algorithm for being embedded in the FPGA device. The FDFC processing is split into two functional groups. The first one has the task to register all the necessary information and the second to perform the calculations.

ISBN 978-3-95450-119-9

Input Data Storage

The FDFC input data storage is based on the pipeline principle. Its simplified block diagram is shown in Fig. 4. Registers on that diagram are shown in blue. The main control element is a free running timer which produces a trigger pulse, every 2 μs, that mark the integration period end. The ADC sample (CNV) is acquired also by that pulse. For simplicity reasons, it is assumed here that the ADC data is available immediately, and it is stored in the "Sample 2" register. At the same time, the old value of that register is written to the "Sample 1" register. This requires at least 2 ADC samples after the algorithm start-up, to obtain a valid result (pipeline). The second sample, written to the "Sample 2" register, corresponds to the generalised name A_{n+1} and the first sample, written to the "Sample 1" register, to A_n. The same names were used also in the previously defined equations.

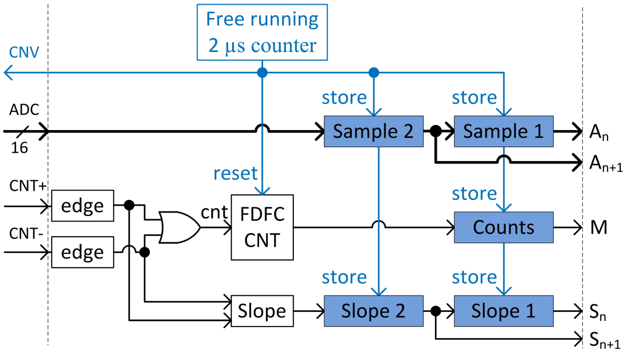

Figure 4: Block diagram of the FDFC input data storage.

The number of times the FDFC signal has reached the extremes of range during an integration period are accounted on the basis of the CNT+ and CNT- inputs coming from the analogue comparators. These inputs require synchronisation to the FPGA clock domain and use of an edge detection logic. Pulses from both comparators are counted by a counter (FDFC CNT). At every integration period end, the counter value is reset just after storing the result in the "Counts" register. The result is called M.

The CNT pulses are additionally used by the storage logic to distinguish the actual slope of the FDFC signal. The "Slope" bit is set when the rising slope starts and it is cleared when the falling slope starts. The "Slope" output is also shifted into the pipeline registers at each integration period pulse. In this way, the actual A_{n+1} and the previous A_n acquired samples are complemented by their slope information, i.e. S_{n+1} and S_n respectively.

In the next step the stored data is passed to the calculation logic.

Result Calculation

In the FPGA logic responsible for the result calculation, the FDFC data given is processed in either of the two paths of the calculation logic shown in the block diagram of Fig. 5.

As it was shown, the simplification of the algorithm allowed the reduction of the number of cases to just two. Case 1, i.e. when $M = 0$, is realised using signed addition of the previous A_n and the actual A_{n+1} acquired samples with the ABS operation. These operations correspond to the Eq. 1. Case 2, i.e. when $M > 0$, is realised by making a sum of the following three components:

- Difference between the previous sample A_n and the extreme (see Eq. 2). The $|A_n - max|$ is selected by the multiplexer when the slope was rising, i.e. $S_n = 1$. The $|A_n - min|$ is selected by the multiplexer when the slope was falling, i.e. $S_n = 0$.
- Difference between the actual sample (A_{n+1}) and the extreme (see Eq. 3). The $|min - A_{n+1}|$ is selected by the multiplexer when the slope was rising, i.e. $S_{n+1} = 1$. The $|max - A_{n+1}|$ is selected by the multiplexer when the slope was falling, i.e. $S_{n+1} = 0$.
- Number of full slopes in the integration period, between the previous A_n and actual A_{n+1} acquired samples (see Eq. 4). One slope is expressed as the actual ADC range, i.e. $|max - min|$. To count all the slopes that number is multiplied by the number of counts decreased by 1 ($M - 1$). The ADC range, i.e. $|max - min|$, differs depending on the actual setting of the digital potentiometer, which changes the V_{thr}. Therefore, the use of a binary multiplier is required.

Finally, the last multiplexer from the right side is used to provide the output of the used path depending on the number of counts accumulated during the integration period.

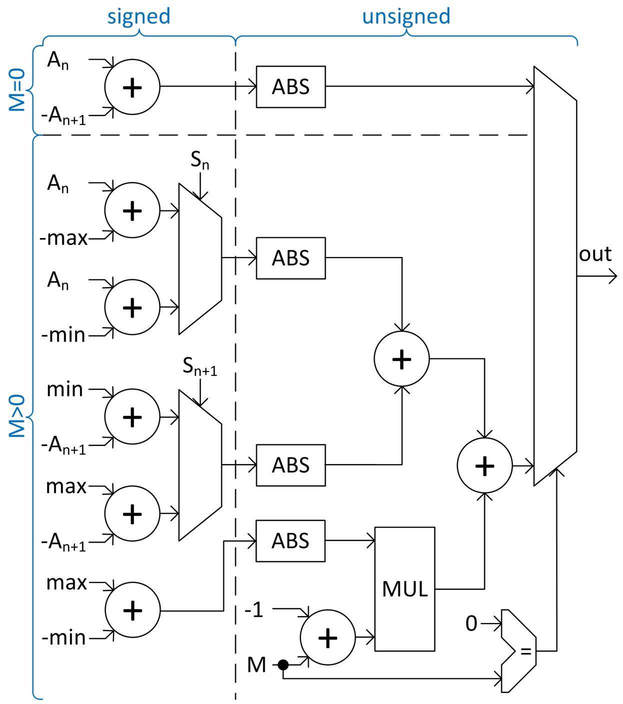

Figure 5: Block diagram of the FDFC result calculation logic.

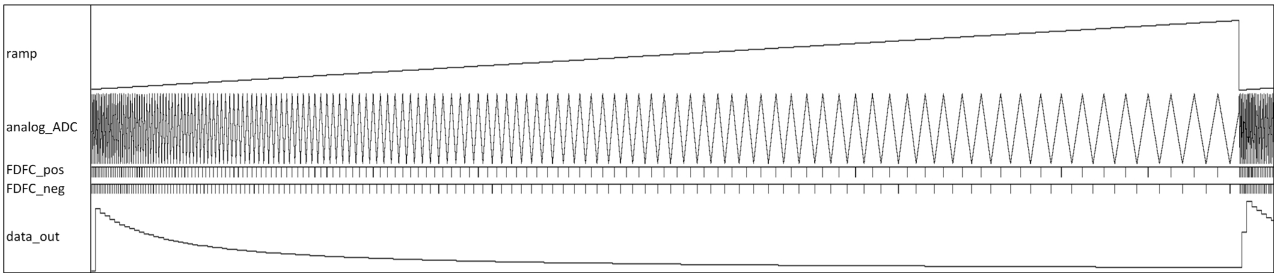

Figure 6: Example of the ModelSim simulation output.

VHDL SIMULATION

The algorithm to be embedded in the FPGA was implemented with the VHDL language. For the resulting code, a test-bench was also written in VHDL and the ModelSim simulation tool was used to perform the functional verification of the algorithm implementation correctness.

The test-bench creates the FDFC output signal represented in ADC output values. In the figure it is called analog_ADC. The period of the analogue signal is slowly changing proportionally to the provided ramp signal. The ramp signal chosen simulates a continuous increase of the input current. In addition to the ADC signal, the test-bench also generates the pulses of the FDFC analogue comparators every time the analogue signal changes its polarity.

The response of the implemented algorithm is registered and plotted using ModelSim. Example results of the simulation are presented in Fig. 6. The response is non linear as expected. The reason is that the result corresponds to the real number of FDFC counts accumulated over the integration period and it must be inverted to show the frequency. The input current is proportional to the frequency of the Fully Differential Frequency Converter.

CONCLUSIONS

The algorithm presented in this paper was implemented with the aim to be embedded in the FPGA device of the digitiser module. With its help the FPGA, which has the ability to control and read the outputs of the analogue circuits digitising the input channel, can provide in real-time the current value integrated over a reference period.

The new BLM system will be progressively deployed in the CERN Injectors Complex. A prototype version of the complete acquisition crate has been installed in the Proton Synchrotron accelerator at CERN. First tests with beam and several types of detectors are on-going.

REFERENCES

[1] C. Zamantzas, M. Alsdorf, B. Dehning, S. Jackson, M. Kwiatkowski, W. Vigano, “Architecture of the System for Beam Loss Monitoring and Measurements Under Development for the Injector Complex at CERN”, 1st International Beam Instrumentation Conference, IBIC2012, Tsukuba, Japan, Oct 1-4, 2012.

[2] M. Kwiatkowski, M. Alsdorf, E. Angelogiannopoulos, B. Dehning, S. Jackson, W. Vigano, C. Zamantzas, “Development of a Beam Loss Measurement System with Gigabit Ethernet Readout at CERN”, 1st International Beam Instrumentation Conference, IBIC2012, Tsukuba, Japan, Oct 1-4, 2012.

[3] W. Vigano, B. Dehning, E. Effinger, G. Venturini, C. Zamantzas, “Comparison of Three Different Concepts of High Dynamic Range and Dependability Optimised Current Measurement Digitisers for Beam Loss Systems”, 1st International Beam Instrumentation Conference, IBIC2012, Tsukuba, Japan, Oct 1-4, 2012.

ISBN 978-3-95450-119-9