Efficient signal conditioning by a FIR filter for analog signal transmission over long lines

This article has been downloaded from IOPscience. Please scroll down to see the full text article.
2012 JINST 7 C01082
(http://iopscience.iop.org/1748-0221/7/01/C01082)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 193.170.243.211
The article was downloaded on 25/01/2012 at 08:47

Please note that terms and conditions apply.
Efficient signal conditioning by a FIR filter for analog signal transmission over long lines

C. Irmler\textsuperscript{a,1}, M. Friedl,\textsuperscript{a} J. v. Hoorne\textsuperscript{a,b} and H. Steininger\textsuperscript{a}

\textsuperscript{a}Institute of High Energy Physics, Austrian Academy of Sciences, Nikolsdorfer Gasse 18, A-1050 Vienna, Austria
\textsuperscript{b}CERN, Genève 23, CH-1211, Switzerland

E-mail: christian.irmler@oeaw.ac.at

ABSTRACT: In the Belle II SVD readout chain the analog signals will be transmitted over long lines. This leads to signal distortion, caused by the frequency dependent transfer function of the cable and also by reflections, which occur whenever the line impedance changes. One possibility to compensate these effects is a dedicated filter at the receiver end. This paper describes the approach to realize the required filter as a finite impulse response (FIR) filter. We further show how such a filter can be implemented in the firmware of an FPGA and the required FIR coefficients can directly be calculated from the output signal of the APV25 front-end chip.

KEYWORDS: Digital signal processing (DSP); Data acquisition circuits; Front-end electronics for detector readout; Data acquisition concepts
1 Introduction

In the readout chain of the Belle II Silicon Vertex Detector (SVD) [1, 2], the differential analog outputs of the APV25 [3] front-end chips will be connected to the back-end electronics boards by about 12 m long twisted pair copper cables. This will be done since there is not enough space inside the detector to add repeater boards to decouple and refresh the analog signals as it is usually done in similar systems. On the other hand, the distance between front- and back-ends is not far enough as to vindicate the usage of cost-intensive custom analog optical links, which moreover would add additional material and power to the sensitive volume. However, connecting the front-end directly to the back-end also implies to deal with the limitations of cable transmission. The APV25 output presents analog strip data, multiplexed at 40MHz, so the resulting bandwidth requirement is in the order of 100MHz.

The frequency-dependent cable loss can be compensated by a HF boost on the sender side (“pre-emphasis”) or a similar filter (“equalizer”) on the receiver side. In order to minimize the material budget and mitigate the problem of potential radiation damage, there will be no additional electronics in the sensitive volume. In other words, the APV25 outputs will directly drive the long cables. This was successfully tested, but of course the APV25 outputs are not specifically made for driving long lines. Consequently, we have to compensate the cable losses at the receiving end.

1.1 Belle II SVD readout system

The Origami Module [4] is the basic element of the Belle II SVD, reading out a 6” double-sided silicon strip detector with APV25 readout chips, which were originally developed for the CMS experiment at CERN.
In Belle II, the differential outputs of the APV25 front-end will be connected to a junction box outside the acceptance volume of the SVD. Apart from patch panels, the box will contain radiation-hard voltage regulators to supply the front-end hybrids. On the far end, FADC+PROC modules will receive the analog data, digitize and analyze them. In total, the analog signals of the APV25 chips will be transferred over 12 m long twisted pair copper cables (figure 1).

The FADC+PROC VME modules will perform analog level translation from ±bias voltages down to ground and digitize the incoming data. Figure 2 shows a simplified block diagram of the functionality of the FADC+PROC board. The signals will be processed inside the central FPGA, an Altera Stratix IV GX. As first step of signal conditioning, a FIR filter (see section 3) with 8 taps will be implemented, followed by pedestal subtraction, common mode correction, sparsification and hit time finding. Finally, the processed data are transmitted to the common Belle II data acquisition by optical links (FTB) or a gigabit ethernet interface.

2 Theory

2.1 Cable transfer function

Every cable has a frequency-dependent transfer function, which causes distortion and widening of the original signal. Moreover, changes of the impedance along the line, in particular at every interconnection between cables and boards, lead to reflections. Even though the latter can be theoretically avoided by proper termination, this is never perfect in real life. In the end, both effects result in an unwanted additional contribution to the noise of the readout system.
The frequency-dependent transfer function is an intrinsic property of the cable and thus cannot be avoided. It can be derived from the telegraph equations as

$$H(f) = e^{-k l (1 + j) \sqrt{f}}$$ (2.1)

where $f$ is the frequency, $k$ a cable constant and $l$ the cable length. Figure 3 shows the comparison of the measured transfer function of a 30 m long CAT7 ethernet cable and its theoretical curve calculated with equation (2.1). This transfer function can be compensated by a HF boost on the sender side (“pre-emphasis”), but also by an analog or digital filter (“equalizer”) on the receiver side. As the APV25 chips are in the radiation zone, we cannot put any commercial electronics there, so we are confined to the receiving end.

In the first approach, a commercial analog equalizer chip (AD8128 [5]) was applied with moderate success. Even tough the filter can be adjusted over a wide range, resulting in a significant improvement of the signal shape, the distortion could not be compensated completely. A disadvantage of this chip is that it has a single-ended output and thus interrupts the otherwise fully differential design of the signal transmission.

Another, but much more efficient possibility is to use a digital finite impulse response (FIR) filter as it is described in section 2.3.

### 2.2 Signal conditioning by filters at the receiver

From the viewpoint of signal theory, the readout chain of the Belle II SVD can be described as shown in figure 4. The discrete signals $s(k)$ and $v(k)$ and the system impulse response $h(k)$ are related by

$$v(n) = \sum_{k=-\infty}^{\infty} h(k) s(n-k)$$ (2.2)

Our goal is to recover the initial signal, hence a filter $f(k)$, fulfilling equation (2.3) is required.

$$s'(n) = \sum_{k=-\infty}^{\infty} f(k) v(n-k)$$ (2.3)
The discrete output signal of the filter \( s'(n) \), then ideally is identical to the original \( s(n) \). Since we presume causality, it is required that \( s(n - k) = 0 \) and \( v(n - k) = 0 \) for \( n < k \). For a practical implementation, \( k \) cannot become infinite and for easier handling we shift the sum to start at zero. So we obtain \( 0 \leq k \leq N \). The \( f(k) \) are then called filter coefficients of the filter system of the order \( N + 1 \). Then, equation (2.2) can be expressed by matrices as shown in equation (2.4),

\[
V_n = \sum_{k=0}^{N} H_{nk} s_k, \quad \text{or} \quad V = H S \tag{2.4}
\]

with \( H_{nk} = 0 \) for \( n < k \). The elements of the system matrix \( H \) can easily be obtained from the pulse response of the system, where \( s(n) = [1, 0, 0, \cdots]^T \):

\[
\begin{bmatrix}
h_0 & 0 & 0 & 0 & \cdots \\
h_1 & h_0 & 0 & 0 & \cdots \\
h_2 & h_1 & h_0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots \\
h_N & \cdots & \cdots & \cdots & \cdots & h_0
\end{bmatrix} \begin{bmatrix}1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix}h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} \tag{2.5}
\]

Once \( H \) is known, the original signal can be reconstructed by

\[
S' = H^{-1}V = FV = H^{-1}HS
\]

where the filter matrix \( F \) can be identified as the inverse matrix of \( H \). The first column of \( F \) contains the coefficients of the required filter.

### 2.3 Finite impulse response filter

A finite impulse response (FIR) filter is a non-recursive digital filter that calculates the output signal as the weighted sum of a certain number of samples of the input signal as described by equation (2.6).

\[
v(n) = \sum_{k=0}^{M} h(k)s(n - k) \tag{2.6}
\]

The number of filter coefficients \( M \), which is equivalent to the number of used samples, is called the order of the filter. One possible realization, the so-called direct form, is depicted in figure 5. Since a FIR filter always has a limited number of coefficients, the length of its impulse response is also limited, hence the name Finite Impulse Response filter. The following list summarizes the main properties of a FIR filter:
The number of coefficients is limited by its order.

It is a linear time-invariant (LTI) system, hence it follows the superposition theorem.

It requires no feedback from its output.

FIR filters are inherently stable and have a linear phase response.

Those features and the simple circuit allows the implementation of a FIR filter in the firmware of an FPGA chip on the FADC+PROC boards. How such an implementation can look like and how the filter coefficients can be obtained from the idle signal of the APV chip, will be shown in the sections below.

3 Implementation

As discussed in section 2.2, the impulse response of the system is required to calculate the filter coefficients. During its idle state, the APV25 chip sends a short pulse once every 35 clocks, the so-called tick mark. Figure 6 shows a scope picture of tick marks of two channels, without and with optimized cable termination, respectively. In both cases a tail caused by the transfer function of the cable is obvious. The non-optimized channel additionally shows a significant negative reflection due to the bad termination. Even for the optimized channel a small positive reflection is visible. This and its aftermath can be interpreted as the sampled impulse responses of the readout chain and thus be used to obtain the filter coefficients. Therefore, the tick marks are averaged over 300 events to cancel out random noise excursions of each single trace. Furthermore they are normalized so that the baseline is shifted to zero and the maximum value is set to one.

For first tests, a FIR filter of the order 32 was implemented in software and verified. Aiming to get a realization that can be programmed into an FPGA, the order was reduced to 8, which still yields excellent results. Such a firmware implementation of a FIR filter with 8 coefficients is shown in figure 7. Since it could be added to the firmware of the existing FPGAs (Altera Stratix 1) for each channel, it does not require additional hardware or space on the prototype FADC boards. The filter uses dedicated digital signal processor (DSP) blocks, which can perform 16 bit signed multiplication and adding at 40 MHz, the APV25 clock speed. Since the filter coefficients are small fractional numbers, they are pre-scaled by $2^{13}$ in order to enable pure integer calculation at high precision. A simulation had shown, that the error of these integer calculation compared to the exact computation with double precision values is lower than 1 ADC count when multipliers larger
Figure 6. Scope picture of the APV25 tick marks for a non-optimized (a) and the optimized channel (b); some “aftermath” (slow decay) of the tick mark pulses and reflections are clearly visible; x-axis: time in multiples of 25ns, y-axis: amplitude in ADC counts.

Figure 7. Schematics of a FIR filter, implemented in an ALTERA Stratix 1 (EP1S20F672C7N).

than $2^8$ are used. Since the DSP blocks offer signed 16 bit integers, we used $2^{13}$ as multiplier, corresponding to an error of less than 0.5 ADC. In that case absolute coefficient values up to 4 are possible. During our tests most of the values were around 1 or below, with some outliers which were slightly above 2. In the practical implementation, the coefficients where first calculated by the data acquisition software, then programmed into a memory of the FPGA and finally applied to the DSP calculations by the firmware.
4 Measurements

We testet the filter with the APVDAQ system, a prototype of the Belle II readout system. The test setup is shown in figure 8, where an APV25 hybrid is connected to a repeater board by a 12 m long twisted pair cable. The connection to the VME digitizer boards is done by short CAT7 cables.

Two APV25 channels with different line termination were measured in order to evaluate the efficiency of the implemented FIR filter. While the cable termination was optimized for the first channel, it was not for second one. Hence, on the non-optimized channel both negative effects, the signal distortion caused by the transfer function of the cable and significant reflections were observed without the FIR filter. The results of these comparative measurements are shown in figures 9 and 10 for both, the optimized and non-optimized channel, respectively. In each case, the APV25 output signal before (a) and after (b) switching on the FIR filter with 8 coefficients is depicted. It is obvious that not only the frequency-dependent effects, but also the reflections, which occur within the filter length of 8 clock samples, are completely removed by the filter. It is clear that a FIR filter can not compensate reflections in the cable itself, they are still there and need to be minimized by correct termination in a real system. However, this comparative measurement demonstrates how efficient even significant signal deterioration can be removed from the resulting digital data by a simple FIR filter.

The efficiency of the FIR filter is also visible in the noise plot of the measured APV chip as shown in figure 11, where the RMS noise of the 128 APV channels is displayed for a properly
Figure 10. APV25 raw output data of a channel without optimized termination (a). The same after filtering with a FIR filter (b). x-axis: time in multiples of 25ns, y-axis: amplitude in ADC counts.

Figure 11. Noise figure with no filter applied (a) and after filtering with a FIR filter of order 8 (b). Note the different y-axis scales when comparing the graphs. x-axis: APV25 channel number, y-axis: RMS noise in ADC counts.

terminated transmission line. Without a filter (left plot), the noise of the channels 0,32,64,96,8… is between 11 ADC (channel 0) and 4.5 ADC (channel 96) and thus considerably higher than the average of the other ones, which is around 1 ADC. The reason is, that the above mentioned channels are the first ones in the transmission order of the APV chip and thus suffer from the huge amplitude of the preceding digital data frame header containing a variable 8 bit number that denotes the pipeline cell address. This “aftermath” results from the frequency-dependent signal distortion of the cable. Figure 11(b) shows the channel noise after applying the FIR filter. Now, the noise peaks are totally removed and the noise of the previously affected channels is in the same order than that of the remaining channels.
5 Summary

In the Belle II SVD, the APV25 front-end will be connected to the VME back-end electronics by about 12 m long twisted pair cables. Using such long transmission lines requires to cope with signal distortion as a result of the frequency-dependent transfer characteristic of the cable. Moreover, every change of the impedance, especially at interconnections, causes reflections. Without any compensation, both effects contribute to a higher noise figure. We introduced a finite impulse response filter as a very powerful method to compensate these nasty effects of the long cable. Such a FIR filter with 8 coefficients was implemented into the firmware of an FPGA on a Belle II readout system prototype. The required filter coefficients were obtained from the idle signal (tick marks) of the APV25 chip. Measurements have shown that this filter can efficiently compensate the distortion of the frequency-dependent transfer function of the cable and thus allows to restore the original signal even in case of imperfect transmission over long lines.

References


