ISBN: 978-3-95450-230-1 ISSN: 2673-5350 doi:10.18429/JACoW-IBIC2021-WEPP27

HIGH SPEED PARALLEL DIGITAL SIGNAL PROCESSING STRUCTURE IN BUNCH-BY-BUNCH POSITION MEASUREMENT BASED ON FPGA*

Ruizhe Wu, Leilei Tang†, Ping Lu, Bao-gen Sun, National Synchrotron Radiation Lab. (NSRL), University of Science and Technology of China (USTC), Hefei, Anhui, China

Abstract

In storage ring, the measurement of bunch-by-bunch positions can help to obtain abundant beam dynamics characteristic information, diagnose the instability of beam motion and provide a basis for the suppression of instability. However, the measurement of bunch-by-bunch requires one analog-to-digital converter (ADC) with high sampling rate and one processor with fast digital signal processing (DSP) ability. With the development of electronics, high sampling rate ADCs are no longer a problem. Therefore, high-speed DSP has become the key. In this paper, a parallel digital signal processing architecture based on polyphase decomposition is proposed. This architecture realizes the GHz DSP speed on the programmable gate array (FPGA), which can be used as the infrastructure of high-speed DSP in the bunch-by-bunch position measurement system.

INTRODUCTION

Beam diagnostics is an indispensable part of synchrotron light source to ensure its stability in operation. In beam diagnostics, the beam position monitor (BPM) used to monitor electron beam can obtain the position and current intensity information of the beam, then shows the running state of the synchrotron light source. In recent years, with the improvements of digital devices in terms of working frequency, DSP is of interest in the field of BPM.

With different DSP dealing rates, different types of beam information can be obtained by BPM:

- Closed Orbit: Sample at 10 Hz for high precision position measurement. The average value of multi turn beam position in the storage ring can be obtained.
- Fast Orbit: Sample at 10 kHz for fast orbit feedback. The average value of multi turn beam position in the storage ring can be obtained.
- Turn-By-Turn: Sample at the time for the bunch to move one turn in the storage ring. The average position data of all bunches in the storage ring can be obtained.
- Bunch-By-Bunch: Sample at the time interval between adjacent bunches. The position of each bunch in the storage ring can be obtained.

At present, BPM can obtain the beam information from closed orbit to turn-by-turn through the development of digital devices [1]. However, the obtain of bunch-by-bunch beam information usually imposes higher requirements on DSP dealing capacity. Hence the BPM of bunch-by-bunch, limited by the DSP part, is still in its initial stage.

The DSP part of BPM is usually implemented with FPGA. By coding, digital logic can be mapped to FPGA, so as to achieve various digital logic functions. This great programming flexibility of FPGA is provided by its internal interconnect bus architecture, but this architecture also limits its processing speed to about 500 MHz. Therefore, the DSP speed of BPM is basically at 500 MHz, and then it is difficult to realize the DSP work of bunch-by-bunch.

In FPGA, static timing analysis technology [2] can verify the correctness of digital circuit timing and predict the working frequency of digital system implemented by FPGA.

$$F_{max} \leq \frac{1}{T_{co} + T_{logic} + T_{routing} + T_{su} - T_{skew}} \quad (1)$$

As shown in the Eq. (1), the highest DSP speed F_{max} depends on T_{co}, T_{logic}, $T_{routing}$, T_{su} and T_{skew} parameters. Among them, T_{co} and T_{su} are intrinsic parameters of FPGA that cannot be changed. $T_{routing}$ and T_{skew} are dynamic parameters in FPGA implementation process optimized by Tcl timing constraints, but its change is quite limited. Therefore, only the code logic T_{logic} can be modified to achieve the high speed DSP structure meeting the requirements of bunch-by-bunch.

The high speed parallel DSP structure proposed in this paper will be described in five parts: selection of DSP form, analysis of DSP structure, implementation of DSP structure, data parallelization and performance of parallelization implementation.

SELECTION OF DSP FORM

Generally, in DSP digital filters has two forms, finite impulse response FIR and infinite impulse response IIR.

FIR is a stable all zero structure with linear response characteristics. While IIR, a structure with both poles and zeros, has no linear phase response. If IIR needs to realize linear phase response, it needs to use all-pass filter for correction, which consumes additional digital logic resources. Moreover, due to the nonlinear influence caused by the quantization effect in hardware implementation, the poles of IIR will change and even become unstable [3].

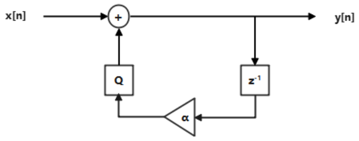


Figure 1: IIR transfer function.

As shown in Fig. 1, the IIR transfer function $\frac{1}{1-\alpha z^{-1}}$ has been transformed into the block design form of state-space

[*] National Synchrotron Radiation Laboratory

[†] tanglei@ustc.edu.cn

equation $y[n] = Q[\alpha y[n-1]] + x[n]$ in time domain, the nonlinear part quantization Q in the equation can be clearly seen. Therefore, FIR is usually selected for DSP.

ANALYSIS OF DSP STRUCTURE

For FIR, the key of structure optimization is to find out the critical path in the delay line of the signal and then transform the critical path.

$$y[n] = \sum_{n=0}^{N-1} h[n]z^{-n} \tag{2}$$

$$y[n] = h[0]z^{-0} + h[1]z^{-1} + \cdots + h[N-1]z^{-N+1} \tag{3}$$

Expand the transfer function equation Eq. (2) of FIR into equation Eq. (3). Then the addition chain direct structure of FIR shown in Fig. 2 can be obtained from Eq. (3), in which the path shown by the hollow arrow is the key path in the structure. Assuming T_S is the latency of the system, T_M is the time to complete a multiplication, T_A is the time to complete an addition, and N is the length of the impulse response function $H[n]$, then the latency of the addition chain direct structure is $T_S = T_M + (N-1)T_A$.

If the addition chain in the addition chain direct structure is modified to the form of addition tree, then the addition tree direct structure can be generated, as shown in Fig. 3. Its lantency is $T_S = T_M + \log_2(n-1)T_A$. Compared to the addition chain direct structure, the tree structure has been time optimized in addition calculation.

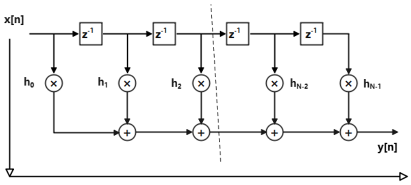


Figure 2: FIR Addition Chain Direct Structure.

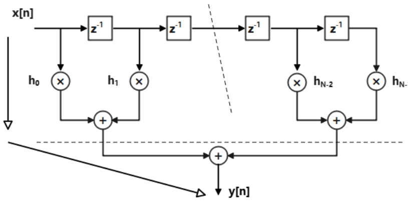
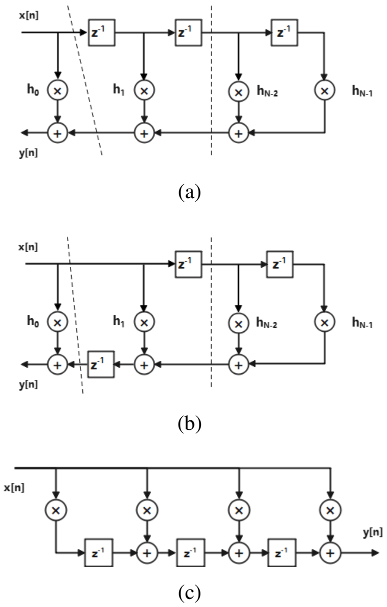


Figure 3: FIR Addition Tree Direct Structure.

The transpose structure can be generated by tranposing each stage in the addition chain direct structure. The transposition process of transpose structure is shown in Fig. 4.



(a)

(b)

(c)

Figure 4: Transpose Process of the Transposed Structure.

Figure 4(a) shows the addition chain direct structure, the vertical dotted line indicates that there are multiple repeated structural units in the middle, and the part between the oblique dotted line and the vertical dotted line indicates the structural unit to be transposed. Figure 4(b) shows the transposed structure unit compared to Fig. 4(a). The transposed structure in Fig. 4(c) can be obtained by repeatedly transposing the structural units. In this structure, the latency is $T_S = T_M + T_A$.

FIR structures are usually the above three structures: addition chain direct structure, addition tree direct structure and transpose structure. Table 1 shows the differences in the three structures.

Table 1: Comparison of FIR Structures

Structure	Latency	Multiplier	Adder
Addition Chain	$T_M + (N-1)T_A$	N	$N-1$
Addition Tree	$T_M + \log_2(N-1)T_A$	N	$2N-3$
Transpose	$T_M + T_A$	N	$N-1$

IMPLEMENTATION OF DSP STRUCTURE

Compared with similar products, Xilinx's FPGA board has higher performance and meets the needs of bunch-by-bunch position measurement better. Therefore, all the following structures are designed and implemented based on the FPGA of Xilinx company.

In order to archieve high performance DSP, Xilinx integrates DSP48E2 slice in FPGA. DSP48E2 slice includes one 27-bit preadder, one 27×18 multiplier and one 48-bit ALU to support multiple calculation functions [4]. Moreover, with dedicated clock lines, DSP48E2 slice has much higher performance compared to the conventional logic implemented by configurable logic block.

In most cases, DSP48E2 is in reuse mode for DSP. While to realize high speed DSP, DSP48E2 slices need to be completely used. With the dedicated clock lines of DSP48E2, the DSP speed can reach the clock limit of the board F_{clk}.

By mapping the transposed structure to DSP48E2 slice, the implementation as shown in Fig. 5 can be obtained [5].

ISBN: 978-3-95450-230-1 ISSN: 2673-5350 doi:10.18429/JACoW-IBIC2021-WEPP27

As shown in Table 1, its transpose latency is $T_M + T_A$, which means one processing can be completed only by one multiplication and one addition in DSP48E2. However, its fan-out is large, due to the same signal input in transpose structure. Besides, in FPGA implementation stage, large fan-outs will occupy a lot of wiring resources, resulting in signal congestion and timing deterioration, even timing fail. Therefore, the length N of the impulse response function $H[n]$ is limited by fan-outs, and then the transpose structure with multiple multiplication coefficients cannot be implemented.

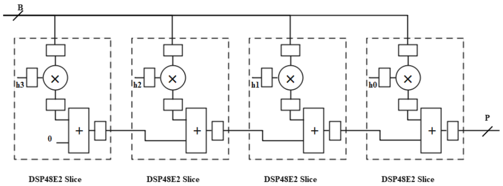


Figure 5: Transpose Structure Implemented by DSP48E2.

By mapping the addition chain direct structure to DSP48E2 slice, the implementation as shown in Fig. 6 can be obtained [5]. This implementation is called systolic, which solves the problem of fan-out at the cost of latency increasement. As shown in Table 1, assuming that the coefficient of DSP is N, the corresponding latency is $T_M + (n-1)T_A$.

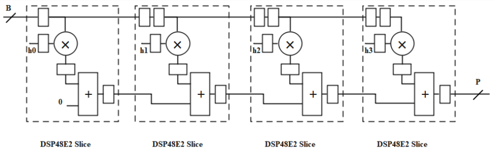


Figure 6: Systolic structure implemented by DSP48E2.

The addition tree direct structure can also be mapped to DSP48E2 slice, but requires more DSP48E2 slices. Due to the valuableness of DSP48E2 slices in FPGA, this structure does not have much implementation value.

Owing to the ultrafast speed and short data interval of DSP48E2 slice, the addition chain direct structure and transpose structure can achieve a speed close to the F_{clk}. Table 2 below shows their implementation.

Table 2: Comparison of Implementations

Structure	Interval	Latency	DSP48E2	N Limited
Addition Chain	$1/F_{clk}$	$T_M + (N-1)T_A$	N	No
Addition Tree	$1/F_{clk}$	$T_M + \log_2(N-1)T_A$	$2N-3$	No
Transpose	$1/F_{clk}$	$T_M + T_A$	N	Yes

However, in FPGA, the number of DSP48E2 slices is limited, so it is impossible to fully use DSP48E2 to implement all DSP functions, and the maximum working frequency of DSP48E2 is also limited to F_{clk}. Therefore, in order to achieve the speed of GHz DSP, one parallel structure is needed.

DATA PARALLELIZATION

Polyphase decomposition is a common technology in multi-rate DSP [6]. Equation (4) is the form of polyphase decomposition.

$$H(z) = \sum_{k=0}^{M-1} z^{-k}E_k(z^M), E_k(z) = \sum_{i=0}^{\frac{L}{M}-1} z^{-i}h(iM+k) \tag{4}$$

Polyphase decomposition transforms the transfer function $H(z)$ of DSP into multiple sub filter structures, archieve the data shunting of each sub filter path, and reduces the amount of data on one data path. The multi-input single-output (MISO) DSP system will be obtained if the left delay line of polyphase decomposition is removed, as shown in Fig. 7. In this DSP system, the speed reduction is realized.

$$H(z) = \sum_{k=0}^{M-1} \sum_{i=0}^{\frac{L}{M}-1} z^{-k}z^{-iM}h(iM+k) \tag{5}$$

$$Y(z) = H(z)X(z) = \sum_{k=0}^{M-1} \sum_{i=0}^{\frac{L}{M}-1} z^{-k}z^{-iM}h(iM+k)X(z) \tag{6}$$

If the two equations in Eq. (4) are combined, the Eq. (5) can be obtained, and then the Eq. (6) of the output $y(n)$ in the z domain is also obtained.

By transforming Eq. (6) from z domain to time domain, the Eq. (7) between input $x(n)$ and output $y(n)$ in time domain can be obtained, where $H_k(n)$ is the pulse response function of sub transfer function $E_k(n)$.

$$\begin{aligned} y(n) &= \sum_{k=0}^{M-1} \sum_{i=0}^{\frac{L}{M}-1} h(iM+k)x(n-(k+iM)) \\ &= h_0(n) * x(n) + \cdots + h_k(n) * x(n) \\ &= \sum_{k=0}^{M-1} h_k(n) * x(n-k) \end{aligned} \tag{7}$$

According to the linear time invariant property of DSP, the output $y(n-(L-1))$ Eq. (8) can be obtained.

$$\begin{aligned} & y(n-(L-1)) \\ &= \sum_{k=0}^{M-1} \sum_{i=0}^{\frac{L}{M}-1} h(iM+k)x(n-(k+iM+L-1)) \\ &= \sum_{k=0}^{M-1} h_k(n) * x(n-k-(L-1)) \end{aligned} \tag{8}$$

Transform the output $y(n-(L-1))$ into the z domain again, Eq. (9) can be obtained.

$$\begin{aligned} & Y(z-(L-1)) \\ &= H(z)X(z-(L-1)) \\ &= z^{-(L-1)}[\sum_{k=0}^{M-1} \sum_{i=0}^{\frac{L}{M}-1} z^{-k}z^{-iM}h(iM+k)X(z)] \end{aligned} \tag{9}$$

Equation (9) has only one more time shift unit $z^{-(L-1)}$ compared with Eq. (6), so the M-Input L-Output structure (MILO) can be obtained, as shown in Fig. 8. Obviously, L times of resources are required for data parallelization.

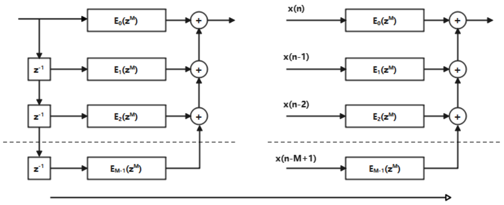


Figure 7: MISO.

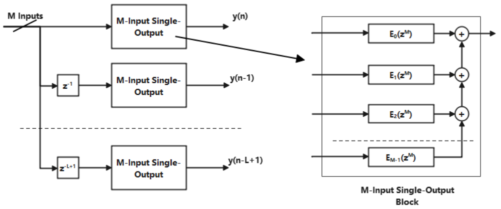


Figure 8: MILO.

During high-speed signal processing, with the help of the MILO, the data stream can be shunted, so that the signal processing frequency on the data input branch path can be reduced to $1/M$, and the GHz signal processing can be easily realized. At the same time, the L outputs can match the parallel input transfer function of the next stage and fully implement the parallel processing of data.

PERFORMANCE OF PARALLEL DSP STRUCTURE

In order to improve the implementation efficiency of parallelized digital signal processing logic, the high-level synthesis tool HLS provided by Xilinx is used here. HLS can convert C\C++ language code that meets certain hardware rules into finite state machine logic in hardware description language, which helps to quickly realize the hardware logic related to the algorithm. In HLS, the parallelized digital signal processing logic implemented in C\C++ language is described in the form of time domain, so the code can be written with the help of Eqs. (7) and (8).

When the HLS sets the target clock cycle as 5 ns, the performance comparison between common FIR and the 5-Input 5-Output ParallelFIR is shown in Table 3. The target period setten as 5 ns represents the system is expected to work at 200 MHz and above; the estimated period 3.42 ns shows the system's maximum operating frequency is 209 MHz; the two cycles latency indicates that the time required for the data from input to output is 6.84 ns; the one cycle interval indicates that the data input is the same as the working speed of the system.

Table 3: HLS implementation Performance Comparison

Type	Target	Estimated	Latency	Interval	Data Deal Rate
FIR	5 ns	3.42 ns	2 cycles	1 cycle	209 Msps
ParallelFIR	5 ns	3.42 ns	2 cycles	1 cycle	1045 Msps

In Table 3, the performance of the two systems are the same. It is found that the performance of common FIR determines the performance of MILO system, but there are differences in data deal rate between the two systems. For common FIR, the data deal rate is 209 Msps; for the ParallelFIR, the data deal rate is 209 Msps×5 = 1045 Msps. As a result, if the parallelism of the MILO structure is increased, higher signal processing speed can be achieved and the needs of bunch-by-bunch can be meeted.

Cite an instance for Hefei Light Source II. Its storage ring operates at 204 MHz. In the case of N points sampling for one bunch in storage ring, the bunch-by-bunch signal processing speed should be 204 MHz×N. Based on MILO system, if the N parallellization is carried out for bunch-by-bunch signal processing, the DSP implemented by FPGA only needs to work at 204 MHz. For FPGA, the frequency of 204 MHz can be easily achieved. Therefore, this MILO structure can meet the requirements of bunch-by-bunch BPM simply.

CONCLUSION

The parallel data processing structure based on polyphase decomposition is a method of exchanging resources for processing performance. When the degree of parallelism is L, L times resource consumption is needed, but there is an M-fold improvement in data processing rate requirement.

At present, the conventional bunch-by-bunch BPM mostly uses simple time-domain algorithms, such as difference-ratio-sum algorithm, interpolation algorithm and look-up table matching algorithm, but there are very few algorithms in digital frequency domain. The main reason is that the digital frequency domain algorithm is limited by its implementation structure, and it is difficult to realize the frequency domain processing under the condition of high data input rate. The parallel data processing method based on polyphase decomposition breaks through this limitation, can easily deal with the data with GHz input rate, and makes a variety of frequency domain processing methods possible in bunch-by-bunch BPM.

REFERENCES

[1] X. Zhang, "Development of Bunch-by-bunch Beam Position Measurement Electronics for High Energy Photon Source", University of Chinese Academy of Sciences, 2020.

[2] Gangadharan, S. and S. Churiwala, *Constraining Designs for Synthesis and Timing Analysis - A Practical Guide to Synopsys Design Constraints (SDC)*, New York, USA: Springer, 2013, p. 226.

[3] Mitra, S. K., *Digital Signal Processing, Computer-Based Approach*, NY, USA, McGraw-Hill Companies, 2011.

[4] Xilinx, *UltraScale Architecture DSP Slice User Guide*, 2020.

[5] Hawkes, G.C., *DSP: Designing for Optimal Results*,. Xilinx Advanced Product Division, 2005.

[6] P, Vaidyanathan., *Multirate Systems And Filter Banks*, Upper Saddle River, NJ, USA, Prentice-Hall, 1993.