FLOATING-POINT-BASED HARDWARE ACCELATOR OF A BEAM PHASE-MAGNITUDE DETECTOR AND FILTER FOR A BEAM PHASE CONTROL SYSTEM IN A HEAVY-ION SYNCHROTRON APPLICATION*

Faizal Arya Samman, Pongyupinpanich Surapong, Christopher Spies and Manfred Glesner
FG Mikroelektronische Systeme, Technische Universität Darmstadt, Darmstadt, Germany

Abstract

A hardware implementation of an adaptive phase and magnitude detector and filter of a beam-phase control system in a heavy ion synchrotron application is presented in this paper. The main components of the hardware are adaptive LMS filters and phase and magnitude detectors. The phase detectors are implemented by using a CORDIC algorithm based on 32-bit binary floating-point arithmetic data formats. The floating-point-based hardware is designed to improve the precision of the past hardware implementation that were based on fixed-point arithmetics. The hardware of the detector and the adaptive LMS filter have been implemented on a programmable logic device (FPGA) for hardware acceleration purpose. The ideal Matlab/Simulink model of the hardware and the VHDL model of the adaptive LMS filter and the phase and magnitude detector are compared. The comparison result shows that the output signal of the floating-point based adaptive FIR filter as well as the phase and magnitude detector agree with the expected output signal of the ideal Matlab/Simulink model.

INTRODUCTION

In a synchrotron, different modes of coherent longitudinal beam oscillations may occur due to an initial mismatch or to non-linearities such as wake fields. These oscillations are characterized by their mode number m and n [4] and take place at the characteristic synchrotron frequency, which depends on the system state (more precisely, on the magnetic flux derivative, accelerating voltage, and particle energy). In order to eliminate undesired dipole oscillations, a beam phase control system [5] has been devised, which was initially designed to deal with in-phase dipole oscillations $(m = 1, n = 0)$ only [5]. The addition of amplitude detectors is intended to make it suitable for damping higher-order modes [4, 6].

This paper presents a new floating-point based architecture designed to improve both the precision and processing speed of the digital controller over previous implementations [2, 11]. The floating-point arithmetic units are used to overcome issues with explosive divergence and stalling effects caused by fixed-point implementations [8]. Such problems may arise due to input or output signal saturation caused by changes in the dynamic signal range at runtime. Using a floating-point representation eliminates the need for input and output scaling that would have to be performed in a fixed-point implementation to accomodate these changes. A floating-point implementation can also be adapted to different dynamic range requirements of other control applications and is therefore more suitable for an integrated circuit implementation (ASIC).

In the following section, the digital signal processing chain of the beam phase control is presented. Its core components are phase and magnitude detectors and adaptive filter blocks, which are discussed in the subsequent sections.

BEAM PHASE CONTROL

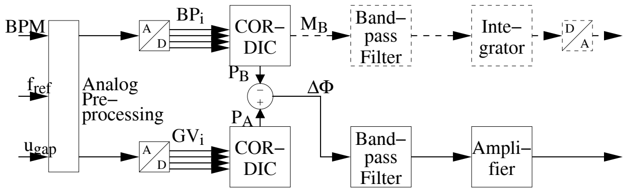

Figure 1: Block Diagram of the Beam Phase Signal Processing.

The signal processing pipeline implemented in the DSP system is shown in Fig. 1. It performs the following steps:

1. Some analog preprocessing is performed on both the gap voltage and the beam position signal [3]. Note that the beam position signal is quite noisy and has a high level of uncertainty since the bunch may have an arbitrary shape.

2. Four successive samples are used to form the in-phase and quadrature components of both signals.

3. The phase of both signals is computed using Alg. 3.

4. The phase difference is fed into a programmable FIR filter in order to eliminate noise and other disturbances in the phase difference. The filter is tuned to a frequency slightly above [5] the synchrotron frequency and has to be re-tuned continuously since this frequency changes.

5. The output of the filter is fed into a variable-gain controller whose gain also depends on the characteristic frequency. This controller yields a frequency correction which is then applied to the cavity in order to intentionally mistune it, thereby pulling the bunch back to its desired position.

BPM is the signal from the beam position monitor. u_{gap} is the voltage across the gap of the reference cavity. f_{ref} is the reference frequency (see [3]). Future extensions [6] are

[*] The work is supported by German Federal Ministry of Education and Research in the frame of Project FAIR (Facility for Antiproton and Ion Research), Grant Number 06DA9028I.

dashed. In order to enable real-time processing, the phase detector and filtering blocks have been implemented on an FPGA.

PHASE-MAGNITUDE COMPUTATION

Architecture for Phase-Magnitude Computing

The phase and magnitude of the beam signal are detected by using a CORDIC (COordinate Rotation DIgital Computer) algorithm. The drawback of a conventional CORDIC algorithm is its high computation latency [7]. To accelerate the computation, an increment of rotation angle on each iteration becomes one solution. A double-rotation CORDIC algorithm is proposed as depicted in Alg. 1. In the double-rotation CORDIC algorithm, the rotation angle is extended two times (2ϕ) of the conventional CORDIC.

Algorithm 1 MICRO-DRCORDIC-VM

```
Input: : $X_i, Y_i, Z_i, T, \delta_i$
Output: : $X, Y, Z, \delta_{i+1}$
  $X = X_i - 2^{-2 \cdot i-2} \cdot X_i - \delta_i \cdot 2^{-i} \cdot Y_i$
  $Y = Y_i - 2^{-2 \cdot i-2} \cdot Y_i + \delta_i \cdot 2^{-i} \cdot X_i$
  $Z = Z_i - \delta_i \cdot 2 \cdot T$
```

The double-rotation CORDIC algorithm using the redundant method was proposed by Takagi et al. [13]. With the redundant method, the rotation direction of the double-rotation CORDIC algorithm is $\delta_i \in \{-1, 0, 1\}$. However, using the redundant method, the scaling factor is not constant and has to be calculated at run-time. To overcome this problem, we propose the double-rotation CORDIC algorithm as presented in Alg. 2 using the non-redundant method and with a constant scaling factor. Thereby, the online computation problem [1], as well as the scaling problem are also eliminated.

$$\begin{bmatrix} X_N \\ Y_N \\ Z_N \end{bmatrix} = \begin{bmatrix} K_{dr}\sqrt{X_0^2 + Y_0^2} \\ 0 \\ \arctan \dfrac{Y_0}{X_0} \end{bmatrix} \tag{1}$$

Algorithm 2 DRCORDIC-VM

```
Input: : $X_{in}, Y_{in}, N, LTAN, K_{dr}^{-1}$
Output: : $X_{out}, Z_{out}$
  $X_0$=$X_{in}$, $Y_0$=$Y_{in}$, $Z_{in}$=0, $\delta_0$=1 {Initialization}
  if $Y_0 \geq 0$ then
      $\delta_0$=1;
  else
      $\delta_0$=-1;
  end if
  for $i = 1 to N$ do
      [$X, Y, Z$]=MICRO-DRCORDIC-VM($X_o, Y_o, Z_o, TLUT(i), \delta_i$)
      if $Y \geq 0$ then
          $\delta_{i+1}$=1;
      else
          $\delta_{i+1}$=-1;
      end if
      $X_0 = X$
      $Y_0 = Y$
      $Z_0 = Z$
  end for
  $X_{out} = X \cdot K_{dr}^{-1}$; $Z_{out}$=$Z$
```

The constant scaling factor for the double-rotation CORDIC is given by $K_{dr} = \prod_{i=1}^{N}(1 + 2^{-2i-2})$ which approximately equals 1.084727 ($K_{dr}^{-1} = 0.9219$), and the maximum and minimum rotation angle $\sum_{i=1}^{N} 2tan^{-1}(2^{-i-1})$ is in the range of [-0.9885,+0.9885].

Alg. 3 shows the computation of the beam phase and magnitude based on online measurements of the gain voltages GV_i and beam position BP_i signals [2], which is illustrated in Fig. 1. The beam phase difference signal ($\Delta Phase$) is obtained from the difference between the phase detection signals P_A and P_B of the gap voltages GV_i and beam positions BP_i signals, respectively.

Algorithm 3 BEAM Phase and Magnitude

```
Input: : $GV_1$,$GV_2$,$GV_3$,$GV_4$,$BP_1$,$BP_2$,$BP_3$,$BP_4$
Output: : $M, \Delta\Phi$
  $Q_{1A} = GV_1$; $I_{1A} = GV_2$; $Q_{2A} = GV_3$; $I_{2A} = GV_4$
  $Q_{1B} = BP_1$; $I_{1B} = BP_2$; $Q_{2B} = BP_3$; $I_{2B} = BP_4$
  $\Delta X_A = Q_{1A} - Q_{2A}$
  $\Delta Y_A = I_{1A} - I_{2A}$
  [$M_A, P_A$]=DRCORDIC-VM($\Delta X_A, \Delta Y_A, N, TLUT$,$K_{dr}^{-1}$)
  $\Delta X_B = Q_{1B} - Q_{2B}$
  $\Delta Y_B = I_{1B} - I_{2B}$
  [$M_B, P_B$]=DRCORDIC-VM($\Delta X_B, \Delta Y_B, N, TLUT$,$K_{dr}^{-1}$)
  $M = M_B$
  $\Delta\Phi$=$P_A - P_B$
```

The architecture of the 3-stage double-rotation CORDIC algorithm for the beam phase-magnitude detection is illustrated in Fig. 2. The 3-stage pipeline architecture is designed to improve the performance of the beam-phase detector.

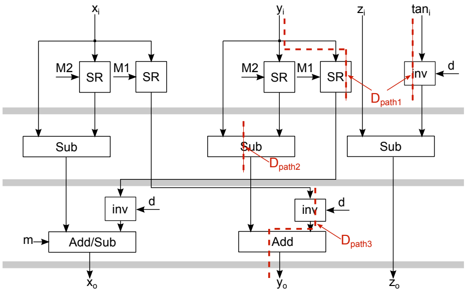

Figure 2: The micro-rotation architecture of the 3-stage double-rotation CORDIC algorithm, where $M_1 = -i$, $M_2 = -2i - 2$ and $d = \delta_i$.

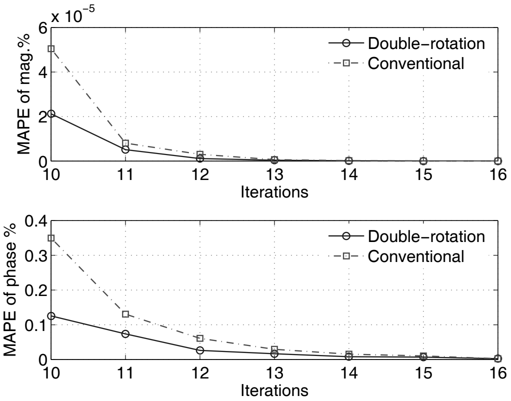

Figure 3: MAPE comparison of the conventional CORDIC and the double-rotation CORDIC.

Figure 3 depicts the performance comparison between the conventional and the double-rotation CORDIC algorithm by using the MAPE metric. MAPE stands for

Mean Absolute Percentage Error and is a statistical measurement of the CORIDC output accuracy formulated by $\frac{1}{n}\sum_{s=1}^{n}|\frac{c_s - m_s}{c_s}|$, where n is the number of measurements, c_s is the CORDIC output and m_s is the ideal output. The figure shows that the double-rotation technique gives better accuracy with the same number of iterations. In other words, for the same MAPE value, the double-rotation CORDIC requires less interations.

Simulation and Synthesis Results

The simulation result of the double-rotation CORDIC algorithm for the beam phase signal detection is depicted in Fig. 4. The figure presents the beam input signal, the reference signal and the the output of the phase detector. The output of the beam phase detector (beam phase signal) is interfered with noisy signal. Therefore, the noise signal should be filtered to acquire the noise-free beam phase signal. The filtering issue is described in the next section.

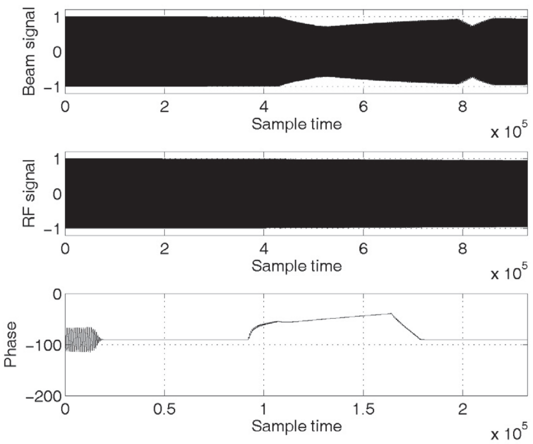

Figure 4: Beam input, reference and beam phase signals.

Table 1 shows the synthesis results of the micro-rotation of the double rotation CORDIC algorithm onto a Xilinx FPGA vlx110t-2ff1738 for different implementations (in terms of the number of pipeline stages). Logic area comparisons between the conventional (CV) and the double-rotaion (DR) CORDIC architectures are presented in the table.

Table 1: Synthesis Results on Xilinx FPGA vlx110t

CORDIC	Utilization			Min. Delay	Max. Freq.
	SReg	SLUT	LUT-FF	(ns)	(MHz)
1-state CV	252	0	0	4.68	213.668
2-state CV	258	343	231	3.05	327.836
3-state CV	273	355	239	2.94	340.513
1-state DR	663	0	0	7.713	129.653
2-state DR	258	698	230	5.045	198.210
3-state DR	746	1228	405	4.107	243.496

SReg=Slice Register, SLUT=Slice LUT

PHASE DIFFERENCE SIGNAL FILTERING

Filter Architecture

The noise in the phase difference of the beam signal shown in Alg. 3 is filtered by using adaptive FIR filter in which the filter parameters are adaptively tuned by using a least-mean-square (LMS) algorithm. The hardware architecture of the adaptive FIR filter is designed by using the combination of serial and parallel structures. This combination is made to fulfill the time constraint of the application and the area constraint of the FPGA device. Figure 5 presents the serial-parallel architecture of the adaptive FIR filter with 64 taps. Two registers are controlled by a control unit and are used to store the sampled input signal $x(k-j)$ and the filter parameters w_j. The filter output computation is implemented in serial, while the filter parameter adaptation algorithm is implemented in parallel.

Floating-point adder, multiplier and multiplier-accumulator units were used in the filter architecture. The operands are IEEE single-precision binary floating-point numbers [12], i. e. they have 1 sign bit, 8 exponent bits and 23 mantissa bits. The floating-point adder architecture consists of three stages: exponent difference and alignment, adding/subtracting combined with leading-on-detection and normalization stage.

The floating-point multiplier architecture also consists of three stages: exponent adding and mantissa multiplication, mantissa shifting and normalization stages. In the normalization stages, the arithmetic computing results are normalized into the standard 32-bit (single precision) binary floating point format. The normalization stages include also the zero and infinity detections of the data values according to the IEEE standard [12]. This approach is the same as the one used in other projects documented in available literature [8].

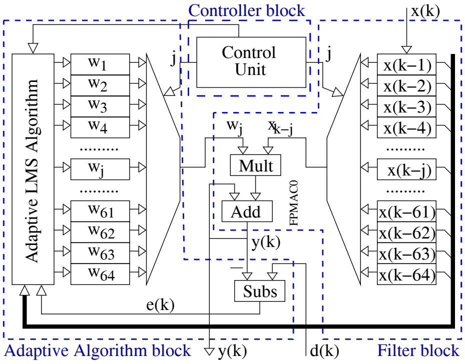

Figure 5: Adaptive FIR Filter architecture with 64 taps.

Simulation and Synthesis Results

Figure 6 shows the input-output signals of the adaptive filter, the desired output signal and the error signal between the filter output and the desired output signals. The main challenge of the adaptive FIR filter hardware in practice

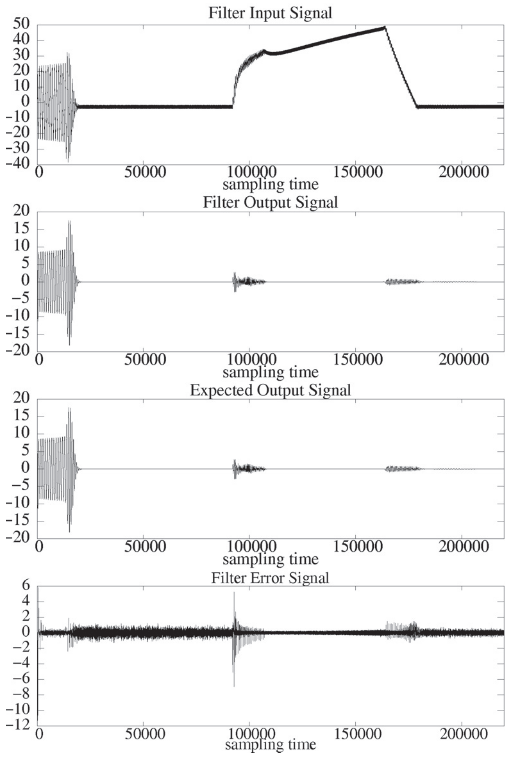

Figure 6: Filter input-output signals.

Table 2: Synthesis Result on FPGA Virtex 5 Device

	Utilization	% of Total
Number of slice registers	38,549	31%
Number of slice LUTs	75,485	61%
Minimum Delay	7.462 ns	
Maximum Frequency	134.018 MHz)	

is the generation of the expected filter output. At present, the expected signal is calculated off-line. In the simulation results shown above, the signal was captured from a system model [9, 10]. We are working on a hardware implementation of a simplified system model that can serve as a reference in the future.

The adaptive FIR filter with LMS algorithm has been implemented on the Virtex5 FPGA device (device type 5vfx200tff1738-2). The device has a total number of 200k logic gates. The synthesis result is shown in Table 2.

CONCLUSIONS AND OUTLOOKS

The beam phase-magnitude hardware detector based on the double-rotation CORDIC algorithm has been presented. Compared to the conventional CORDIC algorithm, the double-rotation CORDIC algorithm provides better accuracy. The adaptive FIR filter architecture that combine serial and parallel computation modes is also presented. The adaptive LMS algorithm is driven by the error signal between the filter output and the desired signal. For now, the desired signal is calculated off-line. In the future, we plan to include a simplified system model in order to supply the adaptive FIR filter with an expected signal that is not known a priori.

ACKNOWLEDGEMENTS

The author would like to thank German Federal Ministry of Education and Research (*Bundesministerium für Bildung and Forschung*) for supporting the work in the framework of Project FAIR (Facility for Antiproton and Ion Research), Grant Number 06DA9028I.

REFERENCES

[1] M. D. Ercegovac, T. Lang "Redundant and on-line CORDIC: Application to matrix triangularization and SVD". *IEEE Transactions on Computers*, pp. 725–740, 1990.

[2] A. Guntoro, P. Zipf, O. Soffke, H. Klingbeil, M. Kumm, and M. Glesner, "Implementation of Realtime and High-speed Phase Detector on FPGA," in *Proc. Int'l Workshop on Applied Reconfigurable Computing*, 2006, pp. 1–11.

[3] H. Klingbeil. "A Fast DSP-Based Phase-Detector for Closed-Loop RF Control in Synchrotrons". *IEEE Trans. Instrumentation and Measurement*, 54(3):1209–1213, June 2005.

[4] H. Klingbeil, D. Lens, M. Mehler, B. Zipfel. "Modeling Longitudinal Oscillations of Bunched Beams in Synchrotrons". http://www.arxiv.org/abs/1011.3957.

[5] H. Klingbeil, B. Zipfel, M. Kumm, and P. Moritz. "A Digital Beam-Phase Control System for Heavy-Ion Synchrotrons". *IEEE Trans. Nuclear Science*, 54(6):2604–2610, Dec. 2007.

[6] D. Lens, H. Klingbeil, T. Gußner, A. Popescu, K. Groß. "Damping of Longitudinal Modes in Heavy-Ion Synchrotrons by RF-Feedback, " in *Proc. IEEE Conf. on Control Applications*, 2010, pp. 1737–1742.

[7] P.K. Meher, J. Valls, Juang Tso-Bing, K. Sridharan, K. Maharatna "50 Years of CORDIC: Algorithms, Architectures, and Applications". *IEEE Transactions, Circuits and Systems*, pp. 1893–1907, 2009.

[8] B. W. Robinson, D. Hernandez-Garduno, and M. Saquib. "Fixed and Floating-Point Implementations of Linear Adaptive Techniques for Predicting Physiological Hand Tremor in Microsurgery". *IEEE Journal of Selected Topics in Signal Processing*, 4(3):659–667, July 2010.

[9] C. Spies, P. Zipf, M. Glesner, and H. Klingbeil, "Bandwidth Requirement Determination for a Digitally Controlled Cavity Synchronisation in a Heavy Ion Synchrotron Using Ptolemy II," in *Proc. Int'l Workshop on Rapid System Prototyping*, 2008, pp. 196–202.

[10] C. Spies, "Model-Based Feasibility Analysis of Digital Beam Phase Control in a Heavy-Ion Synchrotron," in *Proc. Design Automation and Test in Europe, PhD Forum*, 2011.

[11] P. Surapong, M. Glesner, and H. Klingbeil, "Implementation of realtime pipeline-folding 64-tap filter on FPGA," in *PhD-Forum: PhD Research in Microelectronics and Electronics (PRIME)*, 2010, pp. 1–4.

[12] Standards Committee of the IEEE Computer Society. "IEEE Standard for Binary Floating-Point Arithmetic". ANSI/IEEE Std. 754-1985.

[13] N. Takagi, T. Asada, S. Yajima "Redundant CORDIC methods with a constant scale factor for sine and cosine computation". *IEEE Transactions on Computers* pp. 989–995, 1991.