A 3x Blind ADC-based CDR

M. Sadegh Jalali[1], Clifford Ting[1], Behrooz Abiri[1], Ali Sheikholeslami[1], Masaya Kibune[2], Hirotaka Tamura[2]
[1]Department of Electrical and Computer Engineering, University of Toronto, Canada, [2]Fujitsu Laboratories Limited, Japan
Email: {sadegh, cliff, behrooz, ali}@eecg.utoronto.ca, {kibune.masaya, tamura.hirotaka}@jp.fujitsu.com

***Abstract*—This paper uses a 3-bit ADC to blindly sample the received data at 3x the baud rate to recover the data. By moving from 2x to 3x sampling, we reduce the required ADC resolution from 5-bit to 3-bit, thereby reducing the overall power consumption by a factor of 2. Measurements from our fabricated test chip in Fujitsu's 65nm CMOS show a high frequency jitter tolerance of 0.19UIpp for a 5Gbps PRBS31 with a 16″ FR4 channel.**

I. Introduction

Blind ADC-based CDRs simplify the design and simulation of ADC-based CDRs by removing the feedback from the digital CDR to the analog ADC [1]-[4]. As shown in Fig. 1, while in a conventional ADC-based CDR [5], the analog feedback ensures sampling at the center of the UI, a blind ADC-based CDR samples the data with a blind clock. The digital CDR then recovers both the data and its phase. To ensure error-free data recovery, the previous blind CDRs [1]-[3] sample the incoming data at twice the baud rate (2x) using 5-bit flash ADCs. The 5-bit resolution is necessitated by the required accuracy in phase recovery (i.e. in estimating the zero crossing location) when every UI is sampled twice (i.e. 2x sampling). The cost of using a high resolution ADC is area and power consumption as both increase exponentially with the resolution. Alternatively, we can reduce the required ADC resolution without loss of accuracy by increasing the oversampling ratio. In this work, we present a 3x 3-bit ADC-based CDR and we will show a reduction in both area and power consumption, without compromising the CDR performance.

Fig. 2 shows the basic building blocks of a blind ADC-based CDR. First, an n-bit ADC blindly oversamples (at 2x, 3x, or 4x) the incoming data. The blind samples are then fed into the digital CDR, which is comprised of a phase detector (PD), low pass filter (LPF), and data decision (DD) block. The PD determines the locations of the zero crossings, which are then filtered by the LPF to provide the average location of transition, ϕ_{avg}. The ADC output and the ϕ_{avg} are used by the DD block to recover the transmitted bits.

To maintain the same accuracy in data recovery, one can use a higher ADC resolution (i.e. more levels in the voltage domain) with lower oversampling ratio (i.e. less levels in time domain) or a lower ADC resolution with higher oversampling ratio. While a higher ADC resolution increases the power exponentially, a higher phase resolution increases the power linearly. Accordingly, one may favor the latter option. However, reducing the ADC resolution limits the ability of the ADC-based CDR in digital equalization. Fig. 3 compares a 2x system using a 5-bit ADC [1] against a 3x system using

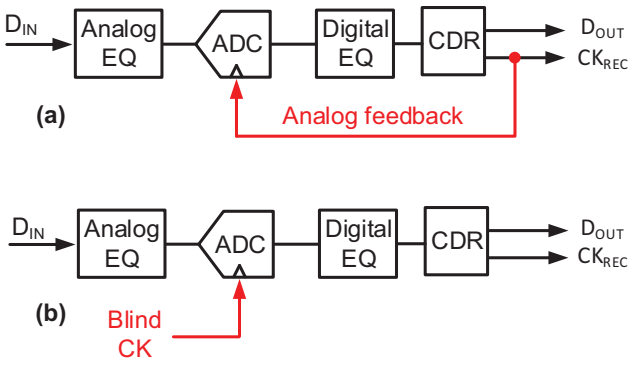

Fig. 1. ADC-based CDRs (a) phase tracking, (b) blind

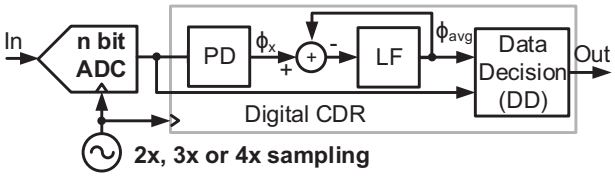

Fig. 2. Basic architecture of a blind ADC-based CDR

a 3-bit ADC (this work). Our simulations show that under the same conditions and with a channel whose loss is 7dB at Nyquist frequency, both systems have almost identical jitter tolerance (both low and high frequency) while they use 110 mW and 38 mW of measured ADC power, respectively. This is no surprise because [1] uses 62 comparators per UI while this work uses only 21 comparators per UI.

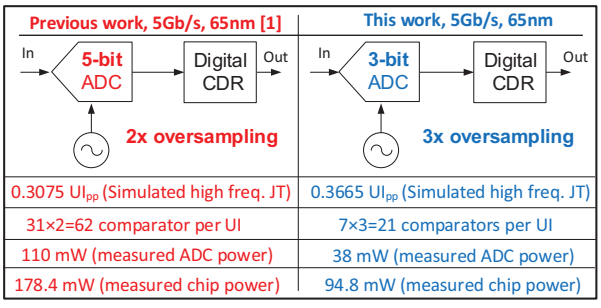

Fig. 3. Comparison with previous work

Fig. 4(a) shows the high frequency (500MHz) jitter tolerance (JT) simulation results of a blind ADC-based receiver, comprised of an n-bit ADC with an oversampling ratio of OSR. The number of comparators used per UI, representative

978-1-4799-0280-4/13/$31.00 ©2013 IEEE

of the analog power consumption, is also shown for each case. As expected, the jitter tolerance increases by increasing OSR or n. Fig. 4(b) compares the JT for different OSRs as we vary the number of comparators per UI (or equivalently, as we vary the allowable analog power consumption). Here, we observe that the jitter tolerance of the system improves by about 0.2UIpp when we move from 2x to 3x, but it almost remains constant when we move to 4x. For this reason, we have chosen to implement a 3x system with a 3-bit ADC.

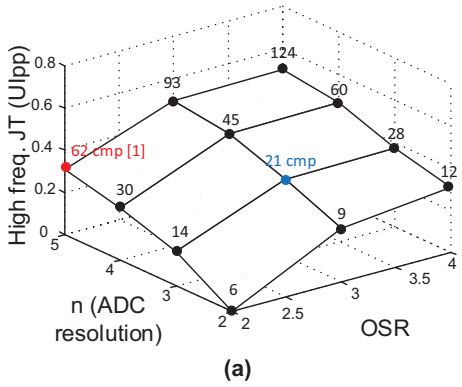

Fig. 4. Simulated JT as a function of (a) ADC resolution and OSR, (b) analog power consumption and OSR

II. Digital Receiver Implementation

Fig. 5 shows a system diagram of the analog front end that samples the 5Gbps data signal at 15GS/s. The channel is connected to CML buffer that drives the 8 interleaved, 3-bit flash ADCs. The CML buffer has fixed capacitive source degeneration in order to equalize the large load of the ADCs and to provide a flat frequency response up to 2.5GHz. The 3-bit ADCs blindly sample the received signal using 8 phases of a 1.875GHz clock. The samples are demuxed to 32 and are fed to the digital CDR. The blind clock, provided by an off-chip 7.5GHz clock source, is divided by a CML shift register into the 8 phases required by the ADCs. One of the 1.875GHz phases is further divided into a 470MHz clock that drives the digital CDR.

In the rest of this section, we explain the design details of the receiver building blocks.

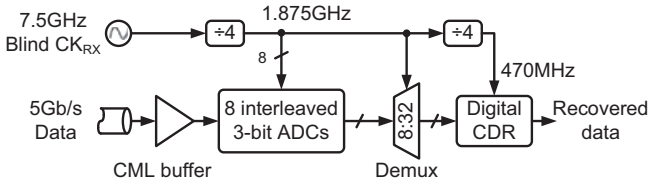

Fig. 5. Blind ADC-based receiver

A. ADC design

Fig. 6(a) shows the single-ended representation of one 3-bit flash ADC including 7 comparators and RS latches, and thermometer-to-binary decoder. In order to reduce the power consumption, the clocked comparators directly sample the data signal without preamplifiers. As illustrated in Fig. 6(b), we chose a modified StrongARM comparator for its low power consumption and narrow sampling aperture [6], [7]. In order to reduce kickback on the data signal and reference ladder, the NMOS transistors driven by CK are stacked on top of the 4 input transistors. Fig. 6(c) depicts the RS latch that follows the comparator. The latched thermometer code is converted into a binary sample by a Wallace adder [8].

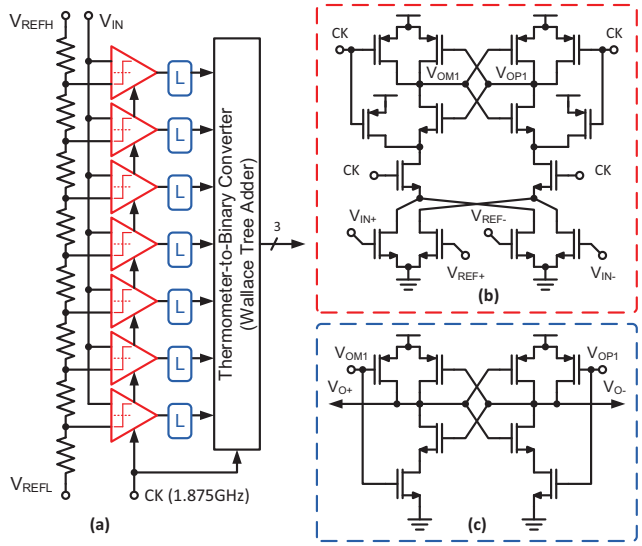

Fig. 6. Implementation of (a) 3-bit ADC, (b) comparator and, (c) RS latch

B. Digital CDR design

Fig. 7 shows the detailed implementation of the CDR. As mentioned before, 32 demuxed ADC samples corresponding to 10.667 UIs (3 samples per UI) enter the digital CDR. Since the CDR processes an integer number of UIs, the 32 samples first enter the variable UI controller block. The role of this block is to convert three of these 32-sample groups (which arrive in three consecutive clock cycles) into three groups of 30, 33 and 33 samples, corresponding to 10, 11 and 11 UIs. Dummy bits are inserted at the beginning of the first 30-sample group to make it equal-size with the other two groups. A flag ($UI\#$) denotes the number of non-dummy data samples in each group.

The 33-sample batch of data then enters the data formatter block, where the ADC output codes are converted from (0 to 7) to (-7 to 7), making the implementation of the phase detector (PD) and data decision (DD) blocks easier.

To find the instantaneous zero crossing phase (ϕ_x), the PD divides the UI into three regions, corresponding to the 3x sampling technique. This is shown in the inset of Fig. 7, where A, B, C and D are the samples taken by the ADCs and cover one full UI. The PD XORs the signs of adjacent ADC codes to yield which region ϕ_x belongs to. While interpolation between adjacent samples can be used to fine tune the position of the zero crossing, it is power hungry. Our simulations show that not using all 3 bits of amplitude information in the 3x PD results in a high freq. JT loss of only 0.05 UIpp while a 2x PD completely breaks if interpolation is eliminated from it. Thus, 3 levels (i.e. 1/6, 3/6 and 5/6) are used to represent ϕ_x.

The instantaneous zero crossing phase is then subtracted from the average zero crossing phase (ϕ_{avg}) to obtain the phase error (ϕ_{err}). This error then goes through a third order loop filter (shown in the inset of Fig. 7) to update ϕ_{avg}. The phase at the eye center (ϕ_{pick}), used by the DD block, is found by adding 0.5 UI to ϕ_{avg}.

The cycle slip monitor (CSM) block handles the frequency offset between the TX and RX clocks. In case of a frequency offset, ϕ_{avg} could go past the 1-UI boundary. This is detected by the CSM block, which either adds or removes a bit to the recovered bit stream [2].

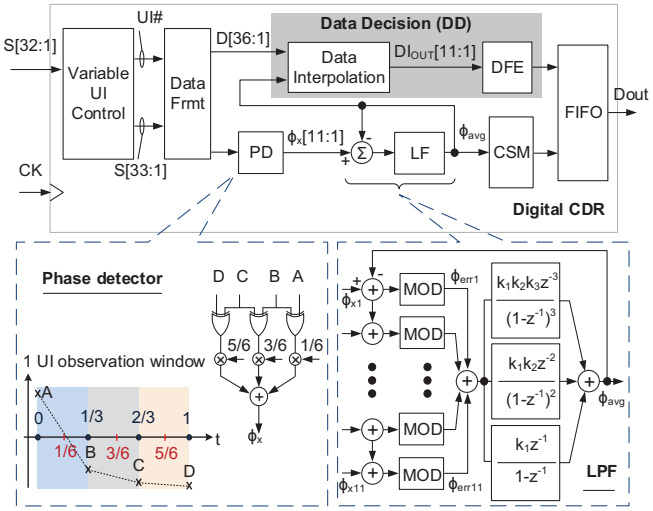

Fig. 7. Detailed system implementation

Fig. 8 shows the implementation of the data decision block and the DFE, where two ADC samples before ϕ_{pick} are A and B and two ADC samples after ϕ_{pick} are C and D. The best estimate of the actual UI center can be obtained by fitting a third order polynomial to these four points and finding the value of the polynomial at ϕ_{pick}. However, this approach is hardware intensive. Alternatively, the UI center can be found by linearly interpolating between B and C and finding the value of this line at ϕ_{pick}. However, the error in this case could be too large. Second order interpolation (estimating the UI center using a second order polynomial), shown in the inset of Fig. 8, seems to provide a good trade off between performance and simplicity. Here, the UI center is estimated by first extrapolating between samples A and B (FWD) and between C and D (BWD) and then performing a weighted sum on the values of these two lines at ϕ_{pick}. Our simulations show a 0.1UIpp increase in high frequency jitter tolerance when using second order interpolation instead of linear interpolation.

Finally, rearranging the equation for the second order interpolation in Fig. 8 yields:

$$DI_{out} = (B - A + C - D)p(1 - p) + (C - B)p + B \quad (1)$$

where p is the distance between ϕ_{pick} and B. Implementing the above equation is hardware intensive, but can be simplified by restricting p to discrete values. Our simulations showed that using a 2-bit resolution for p provides a good compromise.

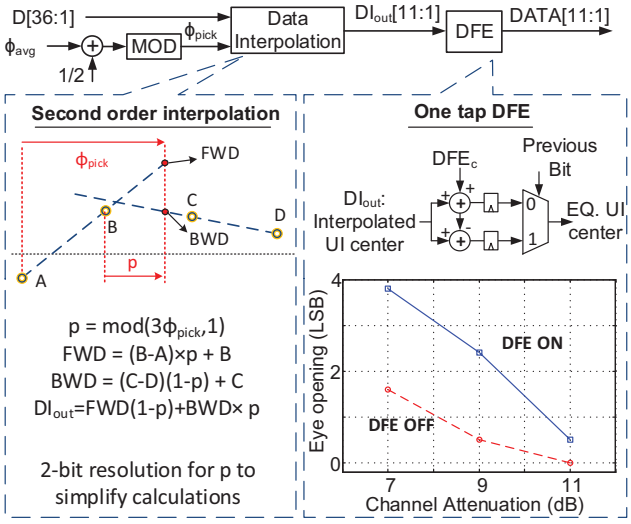

Fig. 8. Implementation of data interpolation and DFE blocks

A one-tap loop-unrolled DFE equalizes the interpolated eye center. The inset of Fig. 8 shows the eye opening of the equalized data versus channel attenuation (at the Nyquist frequency) for a PRBS31 input pattern. By subtracting the channel ISI (α) from the interpolated UI center, the DFE is able to significantly increase the eye opening.

As mentioned earlier, moving from 2x to 3x sampling reduces the ADC power by lowering its required resolution from 5 to 3 bits. In addition, the system implementation as depicted in Fig. 7 and 8 reduces the digital power consumption in two ways: 1. In this work, the DD block only uses ϕ_{avg}, while in [1]-[3] both ϕ_{avg} and ϕ_x are used to make a decision. By dropping ϕ_x from the decision making process, we can afford to lower the accuracy in estimating ϕ_x to three levels (corresponding to 3 samples per UI) and simplify the PD design. Any high freq. error in ϕ_x is heavily attenuated and filtered by the ensuing LPF, maintaining a high accuracy for ϕ_{avg}. 2. Since we have access to the interpolated data at the UI center, we can directly equalize it. This is in contrast

with the previous work [3], in which the DD block needed both equalized blind samples and the resulting (equalized) ϕ_x values. Since the DFE was loop-unrolled and due to the blind nature of the equalizer, the PD and the DD blocks were repeated four times making the design of the equalizer power hungry. As we will see in the next section, our measurements confirm substantial power reduction in both the ADC and in the digital blocks.

III. Measurement Results

The chip, shown in Fig. 9, is fabricated in Fujitsu's 65nm CMOS process. The area of each block is also shown. Without calibration, the ADCs have a measured ENOB of 2.1 bits, which limits the amount of tolerable channel attenuation.

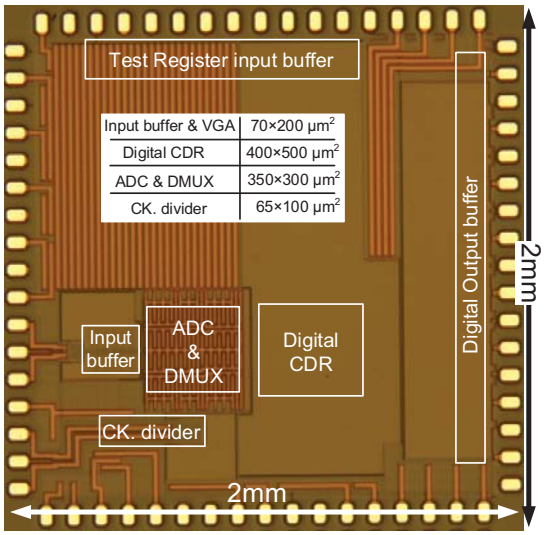

Fig. 9. Chip photo

Fig. 10 shows the measured ADC eye diagram and the JT results for a 5Gb/s PRBS31 input patterns, with no channel and with a 16″ FR4 channel (6dB loss at Nyquist frequency). In each case, the receiver eye diagram is obtained by superimposing the eyes of all 8 ADCs on top of each other. For the measurement with the channel, the DFE was used to improve the JT results. The measured JT for PRBS31 input was 0.27UIpp without a channel and 0.19UIpp with the 16″ channel. Repeating the measurements for a PRBS7 input yields a high freq. JT of 0.45UIpp without a channel and 0.26UIpp with the 16″ FR4 channel.

The ADC and DEMUX consume 38.4mW, the clock divider consumes 14.4mW, and the digital CDR consumes 42mW. Table I summarizes the results and compares this work against previous work.

IV. Conclusion

A 3x blind ADC-based CDR was introduced. By increasing the over-sampling ratio and reducing the ADC resolution, we have traded accuracy in the voltage space for accuracy in the phase space. Also, the additional phase information has simplified the digital CDR design. In this work, we have managed to reduce the overall power consumption by a factor of 2 compared to previous work.

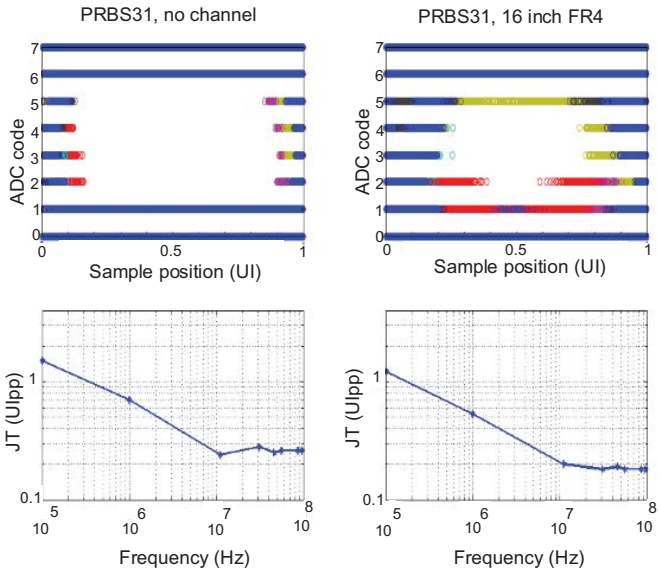

Fig. 10. Measured ADC eye and measured jitter tolerance for PRBS31

TABLE I
Comparison of Digital CDR Results

CDR	Data Rate (Gb/s)	Tech. (nm)	ADC power (mW)	Digital power (mW)	No. of DFE taps	Total power (mW)
[1]	5	65	110	68.4	0	178.4
[2]	5	65	NA	NA	0	280
[3]	5	65	NA	57.6	1	211.2
[4]	10	65	109	111.6	2	306
This work	**5**	**65**	**38.4**	**42**	**1**	**94.8**

Acknowledgment

The authors would like to acknowledge Neno Kovacevic for his help with Verilog verification, and also CMC Microsystems for providing the test equipment and the CAD tools.

References

[1] O. Tyshchenko, et. al., "A 5Gb/s ADC-Based Feed-Forward CDR in 65nm CMOS," *IEEE JSSC*, Vol. 45, No. 6, pp. 1091-1098, June 2010.

[2] H. Yamaguchi, et. al., "A 5-Gb/s Transceiver with an ADC-Based Feed-Forward CDR and CMA Adaptive Equalizer in 65-nm CMOS," *ISSCC, Dig. of Tech. Papers*, pp. 168-169, Feb. 2010.

[3] S. Sarvari, et. al., "A 5Gb/s Speculative DFE for 2x Blind ADC-based Receivers in 65-nm CMOS," *IEEE Symposium on VLSI Circuits, Dig. of Tech. Papers*, pp. 69-70, June 2010.

[4] C. Ting, et. al., "A Blind Baud-Rate ADC-Based CDR," *ISSCC, Dig. of Tech. Papers*, pp. 122-123, Feb. 2013.

[5] M. Harwood, et. al., "A 12.5Gb/s SerDes in 65nm CMOS Using a Baud-Rate ADC with Digital Receiver Equalization and Clock Recovery," *ISSCC, Dig. of Tech. Papers*, pp. 436-437, Feb. 2007.

[6] J. Montanaro, et. al., "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *IEEE JSSC*, Vol. 31, No. 11, pp. 1703-1714, Nov 1996.

[7] M. El-Chammas, and B. Murmann, "A 12-GS/s 81-mW 5-bit Time-Interleaved Flash ADC With Background Timing Skew Calibration," *IEEE JSSC*, Vol.46, No. 4, pp. 838-847, April 2011.

[8] C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, Vol. EC-13, No. 1, pp. 14-17, Feb. 1964.