20.5 A Pattern-Guided Adaptive Equalizer in 65nm CMOS

Shayan Shahramian[1], Clifford Ting[1], Ali Sheikholeslami[1],
Hirotaka Tamura[2], Masaya Kibune[2]

[1]University of Toronto, Toronto, Canada,
[2]Fujitsu Laboratories, Kawasaki, Japan

The use of adaptive equalizers at the front end of receivers is becoming a necessity as the data rates increase without channel improvements. Adaptive equalizers can be implemented using data-aided or non-data-aided schemes [1], with the latter requiring less area and power. Previous non-data-aided adaptive schemes [2-3] implement an asynchronous analog algorithm where the power spectrum of the received signal is checked for balance around a threshold frequency. Similarly, [4] proposes a digital adaptive algorithm which is based on the detection of specific 5-bit patterns. In all three works [2-4], however, adaptation is provided only for equalizers with a single coefficient, which are suitable for well-behaved channels. In contrast, this paper presents a digital adaptive engine for an equalizer with two coefficients: one adjusting the equalizer gain at the Nyquist frequency (f_N) and one at $f_N/2$. Furthermore, the proposed engine is asynchronous; it can function when driven by a blind clock at the receiver. This is useful as it allows the adaptation process to start even when the CDR has not yet achieved lock. This also avoids a deadlock situation where the CDR and equalizer require simultaneous access to the equalized data and the recovered clock. Our measured results of the proposed adaptive equalizer in 65nm CMOS confirm that the adaptation converges to within 2.6% of the optimal vertical eye opening in less than 400µs for two different channels at a data rate of 6Gb/s with a 25,000ppm frequency offset.

Figure 20.5.1 illustrates the concept behind the proposed adaptive equalizer. We assume an equalizer with controllable gains at f_N (C1) and $f_N/2$ (C2) to compensate for the channel loss at f_N and $f_N/2$, respectively. To control the equalizer gain at f_N, we count the number of 4-bit patterns whose signal power is concentrated around f_N (and have no signal power at $f_N/2$). It can be shown that the two 4-bit patterns that have this property are 0101 and 1010. We refer to these patterns as Type 1. Similarly, to control the gain at $f_N/2$, we count the number of 4-bit patterns whose signal power is concentrated around $f_N/2$ (and have no power at f_N). There are four 4-bit patterns among the possible 16 that have this property. We refer to these patterns as Type 2. It can be shown that the remaining ten 4-bit patterns have identical signal power at f_N and $f_N/2$. These patterns are not utilized in the proposed equalization scheme.

As shown in Fig. 20.5.2, the incoming data is attenuated by the channel and subsequently boosted by the equalizer with two independent, adjustable gains, C1 and C2. The equalized signal is sampled by two slicers (S1 and S2) whose thresholds differ by ΔV (mV). If the signal amplitude is above ΔV (mV), the outputs of S1 and S2 will be identical, signifying a vertical eye opening larger than ΔV (mV). The adaptive controller adjusts C1 and C2 to equalize the vertical eye opening to ΔV for both Type 1 and Type 2 patterns. This is achieved by counting each pattern at the output of the two slicers and forcing their respective differences to zero. This completes the equalization for a given ΔV. S1 and S2 are both driven by a clock signal, CK_{RX}. Ideally, CK_{RX} is the recovered clock from a CDR (CK_{REC}). However, our measured results confirm that the proposed architecture adapts to near-optimal coefficients even when there is a frequency offset between CK_{RX} and CK_{REC}. When the sampling phase deviates from the center of the eye, the slicers may underestimate the eye opening and cause the adaptation controllers to overestimate the required gains. To counteract this, the implemented adaptation algorithm has mechanisms (as explained later) to prevent coefficients from converging to over-equalizing gains.

For simplicity, Fig. 20.5.2 omits the two deserializers that convert serial data from the slicers into 16-bit words. The deserializers allow the counters to run at a frequency of $f_N/16$. To count patterns that span word boundaries, the design includes 4 sets of counters to cover all possible cases, but only selects the highest of the 4 counts. We have confirmed by measurement that the additional counters increase the algorithm's robustness and consistency as compared to when only one counter is used.

Figure 20.5.3 shows the C1, C2, and ΔV controllers. The C1 controller iteratively adjusts the gain at f_N until it converges to the lowest value such that the Type 1 count difference is less than or equal to an error tolerance that is programmable between 0 and 50. Eventually, C1 will reach steady state if it toggles between two adjacent values (e.g. {4, 5, 4, 5, 4, 5, 4, 5}), decreases to zero (i.e. {0, 0, 0, 0, 0, 0, 0, 0}), or increases to maximum (i.e. {7, 7, 7, 7, 7, 7, 7, 7}). This is identified via a 7-stage shift register as shown. The C2 controller is identical to the one for C1, except that it reads Type 2 counter differences.

When there is a frequency offset between CK_{RX} and the clock embedded in the data, CK_{RX} may sample the data at the edges (rather than at the center). This may cause the slicers to underestimate eye height, sometimes leading the controller to increase gain erroneously. To compensate for this, we employ two mechanisms in the C1 and C2 controllers. First, the programmable error tolerance allows a count difference up to 50 before the controller increases the gain. In measurement, the design is able to tolerate as much as 25,000ppm of offset with the error tolerance set at 20. Second, due to inherently larger signal slopes at the edge (compared to the center), the convergence checker does not flag the convergence until CK_{RX} samples closer to the center of the eye.

The ΔV controller maximizes vertical eye opening by searching for the greatest ΔV that the C1 gain can compensate while avoiding bit errors on S2's output. It starts at the lowest setting (ΔV=1) and iteratively increments ΔV until the equalizer can no longer amplify the eye opening to ΔV (mV). The controller is based on C1 instead of C2 because the former compensates for higher signal attenuation. After all three coefficients converge, the controllers lock them to their final values.

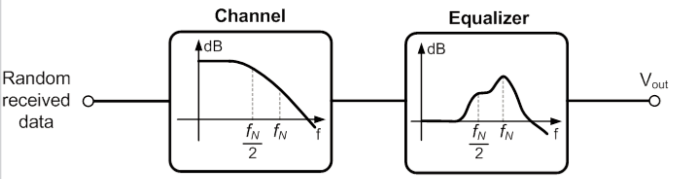
Figure 20.5.4 shows the implementation of the analog equalizer, which consists of three different paths in order to create the desired frequency response [5]. Two bandpass filters are followed by variable gain amplifiers (VGA) that allow independent gain control at f_N and $f_N/2$. The buffers provide a unity-gain path for low frequency data. The bandpass filter is implemented by a differential pair with an RLC load, including a varactor that allows tuning of the center frequency. The VGA is implemented using a differential pair with resistive degeneration controlled by one-hot-encoded switches.

Figures 20.5.5 and 20.5.6 show the measured results of the packaged test chip. Figure 20.5.5 shows the eye diagrams for a 6 Gb/s PRBS 2^7-1 input before and after equalization. The eyes on the left and the right correspond to channel attenuations of 13dB and 17dB, respectively, at 3GHz. The adapted C1 and C2 for the 13dB channel are 7 and 1, respectively. The corresponding C1 and C2 for the 17dB channel are 6 and 2, respectively. Figure 20.5.6 (left) shows the vertical and horizontal eye openings for all possible C1 and C2 levels for a channel with 13dB of attenuation. The adapted eye opening is within 0.2% of the optimal vertical eye opening and within 5.4% of optimal horizontal eye opening. Figure 20.5.6 (right) shows the same results for a 17dB channel with the adapted eye opening being within 2.6% of the optimal vertical eye opening and within 7.0% of the optimal horizontal eye opening. We have also confirmed through measurements that the starting value of the coefficients, C1 and C2, do not affect the final converged state. The equalizer coefficients adapt to their final values in under 400µs. Figure 20.5.7 shows the die photo prior to packaging. The area of the equalizer (marked A) is 0.104 mm^2 with a power consumption of 60mW. The digital adaptation implementation (marked C) occupies an area of 0.101 mm^2 with a power consumption of 16.8mW.

References:

[1] J. W. M. Bergmans, "Non-data-aided adaptive equalization," in *Digital Baseband Transmission and Recording*. Kluwer Academic, 1996.
[2] J. Lee, "A 20Gb/s Adaptive Equalizer in 0.13µm CMOS Technology," *ISSCC Dig. Tech. Papers*, pp 273-282, Feb. 2006.
[3] H. Joo, K. Ha, L. Kim, "A data pattern-tolerant adaptive equalizer using spectrum balancing method," *Symp. on VLSI Circuits*, pp 220-221, 2009.
[4] Y. Hidaka, W. Gai, T. Horie, et al., "A 4-channel 10.3Gb/s backplane transceiver macro with 35dB equalizer and sign-based zero-forcing adaptive control" *ISSCC Dig. Tech. Papers*, pp 188-189, Feb. 2009.
[5] C. Liao, S. Liu., "A 40 Gb/s CMOS Serial-Link Receiver With Adaptive Equalization and Clock/Data Recovery," *IEEE J. Solid-State Circuits*, vol. 43, no. 11, pp 2492-2502, Nov. 2008.

978-1-61284-302-5/11/$26.00 ©2011 IEEE



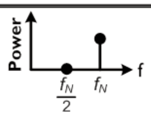
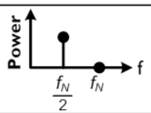
	Data Patterns	Power Spectrum	Equalization Guide
Type 1	1010, 0101		Count these to guide equalization at f_N
Type 2	0011, 0110, 1001,1100		Count these to guide equalization at $f_N/2$

Figure 20.5.1: Six 4-bit data patterns are used to guide equalization. The remaining ten 4-bit patterns are not used because they have identical signal power at $f_N/2$ & f_N.

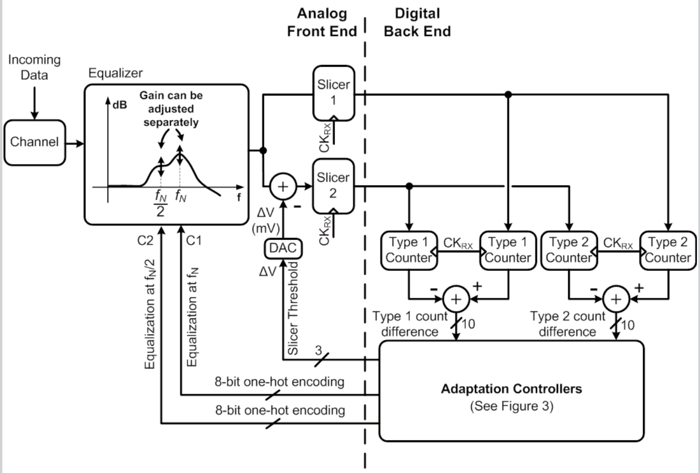


Figure 20.5.2: Block diagram of the proposed adaptive equalizer.

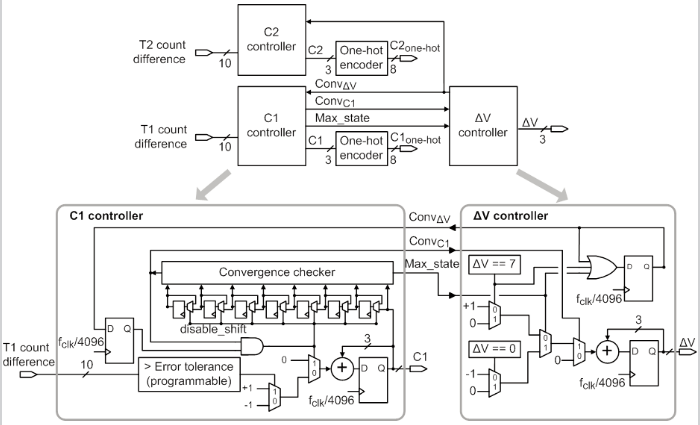


Figure 20.5.3: Implementation of digital adaptation controllers.

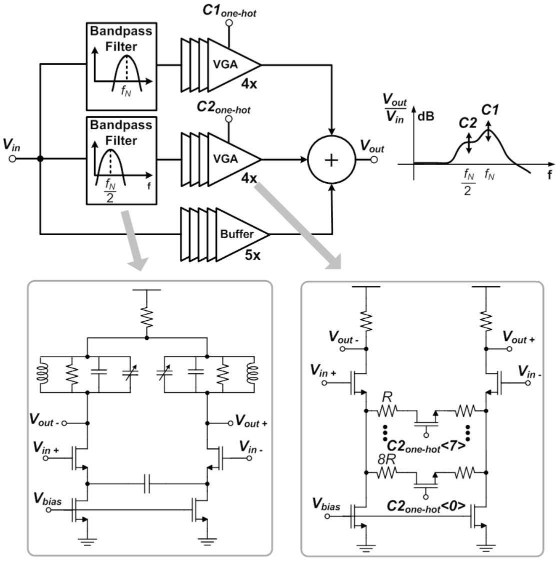


Figure 20.5.4: Equalizer implementation.

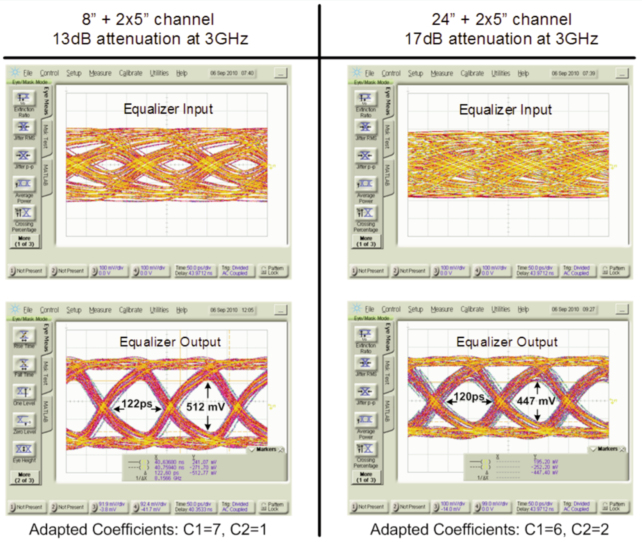


Figure 20.5.5: Measured equalizer input and output eye diagrams for a data rate of 6 Gb/s and a receiver clock with a frequency offset of 25,000ppm.

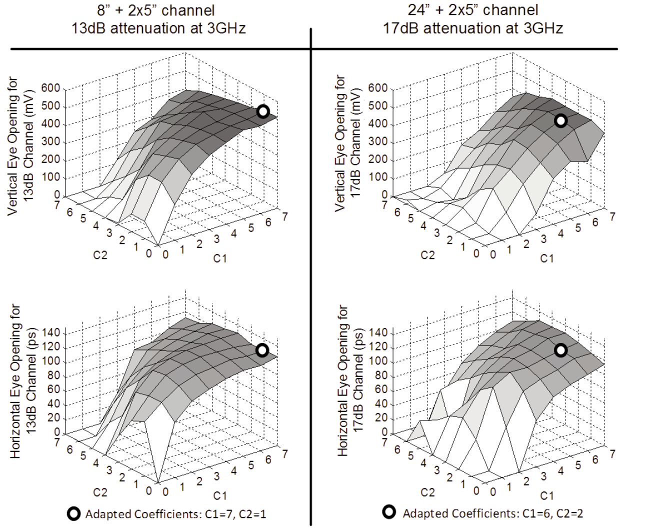
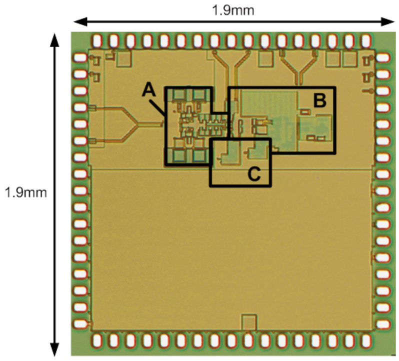


Figure 20.5.6: Measured vertical and horizontal eye openings for all possible gain coefficients for the 13dB and 17dB channels.



Process	65nm CMOS
Data Rate	6Gb/s
Supply	1.2V

		Block description	Area (mm^2)
Analog front end	A	Equalizer	0.104
	B	CDR & output buffers	0.201
Digital back end	C	Digital counters/controllers	0.101
		Total	0.406

Figure 20.5.7: Die photo.