

PAPER

A low-power 1 Gb/s line driver with configurable pre-emphasis for lossy transmission lines

To cite this article: N. St. John *et al* 2023 *JINST* **18** C04009

View the [article online](#) for updates and enhancements.

You may also like

- [Nonlinearity and pixel shifting effects in HXRG infrared detectors](#)
A.A. Plazas, C. Shapiro, R. Smith et al.
- [Dynamic defectoscopy with flat panel and CdTe Timepix X-ray detectors combined with an optical camera](#)
D Vavrik, A Fauler, M Fiederle et al.
- [Numerical studies of petawatt laser-driven proton generation from two-species targets using a two-dimensional particle-in-cell code](#)
J. Domaski, J. Badziak and S. Jabloski

TOPICAL WORKSHOP ON ELECTRONICS FOR PARTICLE PHYSICS
BERGEN, NORWAY
19–23 SEPTEMBER 2022

A low-power 1 Gb/s line driver with configurable pre-emphasis for lossy transmission lines

N. St. John,* S. Mandal, S. Miryala, P. Maj, G.W. Deptuch, E. Raguzin and S. Rescia

Brookhaven National Laboratory, Upton, NY, U.S.A.

E-mail: nstjohn@bnl.gov

Abstract: A line driver with configurable pre-emphasis is implemented in a 65 nm CMOS process. The driver utilizes a three-tap feed-forward equalization architecture. The relative delays between the taps are selectable in increments of 1/16th of the unit interval via an 8-stage delay-locked loop and digital interpolator. It is also possible to control the output amplitude and source impedance for each tap via a programmable array of eight source-series terminated drivers. The entire design consumes 9 mW from a 1.2 V supply at 1 Gb/s.

Keywords: Digital electronic circuits; Digital signal processing (DSP); VLSI circuits

ArXiv ePrint: [2210.11882](https://arxiv.org/abs/2210.11882)

*Corresponding author.

Contents

1	Introduction	1
2	Circuit design	2
3	Output waveform optimization	4
4	Test structure and results	5
5	Conclusion	6

1 Introduction

Line drivers in front-end ASICs must be optimized for the specific transmission line (i.e., cable) used by the system to ensure a low bit error rate (BER) to transmit digitized measurement data off-chip at minimal power budget. Furthermore, as data rates continue to increase, transmission line effects (signal attenuation and reflections) begin to hamper the performance of the data link. In particular, attenuation generally increases with frequency, resulting in a transfer function (TF) for the transmission line that is low-pass in nature, while reflections give rise to sharp maxima/minima at particular frequencies due to wave interference. In order for transmitters to counteract such frequency-dependent attenuation of the cable, which generates pulse distortion and inter-symbol interference (ISI) in the time domain, pre-emphasis or feed-forward equalization (FFE) is used [1].

Pre-emphasis is a technique in which the BER for a given data rate can be improved via controlled pre-distortion of the output data signal, in particular by adding overshoot (“emphasis”) around data transitions [2]. This process adds high-frequency components to the signal. Ideally, the output waveform should include the inverse of the cable in order to completely cancel the low-pass TF of the cable over a desired frequency range. Pre-emphasis can be implemented via an N -tap finite impulse response (FIR) filter as shown in figure 1(a). This example uses three taps ($N = 3$) with programmable inter-tap delays, τ_i , and tap weights, c_i . The resulting continuous-time transfer function is given by $H(f) = c_0 + \sum_{i=1}^N c_i \exp(-i2\pi f \sum_{j=1}^i \tau_j)$ and should have a high-pass response. Conventional FFE implementations constrain the delays to $\tau_i = 1/f_{\text{clk}} = 1 \times \text{UI}$ where f_{clk} is the clock frequency and $1/f_{\text{clk}}$ is the unit interval (UI) [3]. In this case, $H(f)$ is periodic with a period of f_{clk} , so the desired high-pass response is only obtained over a limited frequency range (up to $1/(2f_{\text{clk}})$). In addition, the output voltage swing is limited by power supply headroom, so the FFE coefficients must be normalized such that $\sum_{i=0}^N |c_i| = 1$. Thus, adding extra taps may not significantly improve performance, because the weights of the other taps must decrease to satisfy the headroom constraint, which limits the peak gain. Additionally, the shunt capacitance of the driver increases approximately linearly with the number of taps, thus reducing its output bandwidth. Thus, we use a three-tap design ($N = 3$). We also relax the constraint that all the

τ_i 's are equal to $1 \times \text{UI}$, thus allowing for greater flexibility in synthesizing the desired TF over a broad frequency range. This property is particularly useful in compensating for the attenuation of high-loss cables, such as radio-pure microstrip lines on flexible Taiflex™ substrates developed for the nEXO experiment.

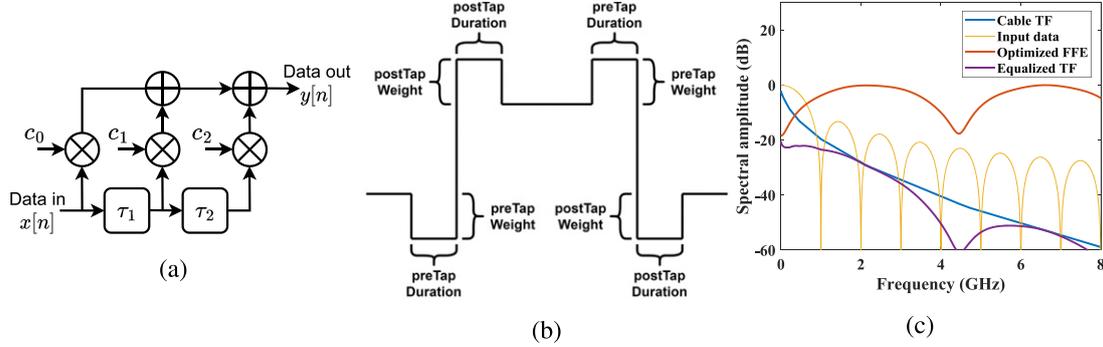


Figure 1. (a) Block diagram of the FFE architecture. (b) General pre-emphasized output waveform. (c) Simulated TF of a 3 m length of radio-pure planar transmission line on a flexible Taiflex™ substrate compared with the power spectrum of random digital data at 1 Gb/s. The optimized TF of a three-tap FFE and the resulting equalized TF of the channel are also shown.

The generic shape of the FFE output waveform is shown in figure 1(b). The weight and duration of the first tap (known as preTap) and the third tap (known as postTap) define the amount of pre-emphasis and are optimized to minimize BER (i.e., maximize the eye opening) for a given cable and data rate. As an example, we used electromagnetic simulations to extract a broadband S -parameter model of a 3 m long radio-pure cable. The results, which are shown in figure 1(c), reveal a loss of -7 dB/m at 1 GHz. The FFE was optimized to cancel this frequency-dependent channel loss over a bandwidth of 2 GHz, which is adequate for our assumed data rate of 1 Gb/s (i.e., $\text{UI} = 1$ ns). The equalized TF, which is also shown in the figure, is nearly flat over this bandwidth, which minimizes ISI as desired. By contrast, note that a conventional FFE (with delays limited to $1 \times \text{UI}$) would only be able to equalize the TF up to a frequency of $0.5/\text{UI} = 0.5$ GHz.

The rest of this paper is organized as follows. The architecture of the line driver is described in section 2. Section 3 discusses the scheme used to optimize the output waveform. Section 4 describes (1) a test chip designed in a 65 nm process, and (2) both simulated and experimental results of the optimized line driver design. Finally, section 5 concludes the paper.

2 Circuit design

A block diagram of the line driver is shown in figure 2. The design consists of two main parts, namely the tap duration control and tap weight control circuits. These work in tandem to generate the proposed fully-configurable output waveform and are discussed below.

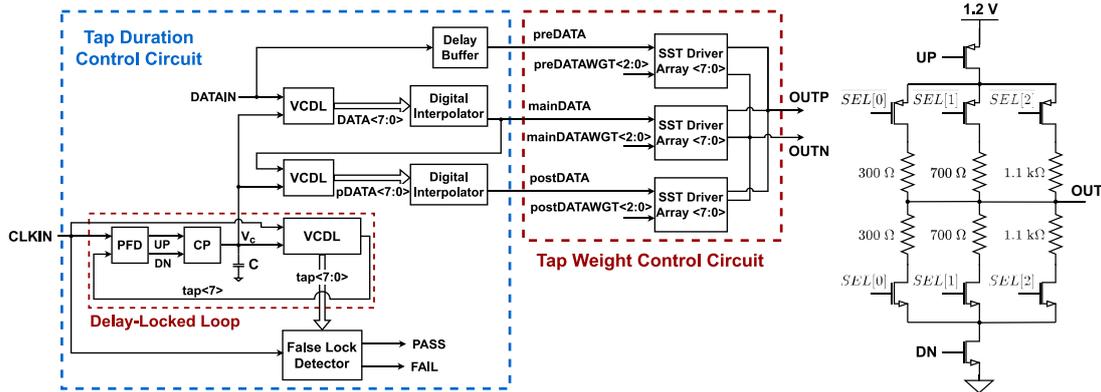


Figure 2. (a) Block diagram of the proposed line driver. (b) Schematic of an SST driver with configurable source resistance; three arrays of such drivers are used to generate the output waveform.

Tap duration control circuitry: We utilize a multi-tap delay-locked loop (DLL) to generate user-configurable tap durations that are robust to process, voltage, and temperature (PVT) variations. Specifically, we use an 8-stage DLL with a standard architecture consisting of a phase-frequency detector (PFD), charge pump (CP), and voltage-controlled delay line (VCDL). A capacitive load at the output of the CP creates a control voltage, V_c , that stabilizes upon locking of the DLL. A novel, single-test false lock detector, which we will discuss later, monitors the DLL to ensure proper operation. The control voltage V_c then feeds two standalone 8-stage replica VCDLs that generate the delays of the main and post-cursor taps. Each creates eight delayed copies of its input waveform with relative delays equal to those of the stages in the main VCDL. The delayed data waveforms then feed a single-stage digital interpolator (DI) [4] that allows the user to interpolate between adjacent taps, thus increasing the timing resolution by $2\times$. Hence, there are 16 uniformly-spaced relative delay options for each tap, resulting in a resolution of $UI/16$. Finally, a buffer, accounting for the delay of the DI, inputs the system data and outputs the precursor tap. This concept of delaying the data rather than re-timing it via flip-flops is ideal for high-speed design as it does not require power-hungry high-speed flip-flops. Furthermore, the use of replica VCDLs allows designers to easily add additional taps by daisy-chaining them, as we have done here.

False lock detector: The DLL includes a false lock detector (FLD) circuit that can determine if the final tap is locked to a delay that is not equal to $1 \times UI$; this is crucial since an unlocked DLL will degrade the granularity of relative delays available to the user. An ideal FLD should only run once to verify correct locking, but earlier FLD designs wasted power by running continuously [5]. Thus, we developed the novel one-shot FLD algorithm shown in figure 3(a). When run on our 8-stage VCDL, this algorithm compares the relative delays of taps 1–4 and tap 8. Two checks are performed during the test: check #1 detects locking at an integer multiple of the UI (>1), while check #2 detects locking at a non-integer multiple of the UI. Check #1 compares taps 1–4. If the DLL locks correctly, tap 4 has the maximum relative delay of this set, so the delay between tap output and input clock increases from tap 1 to tap 4. However if the DLL locks at an integer multiple of the UI >1 , these delays do not increase monotonically. Therefore, we compare the

delays of $\text{tap}[i]$ and $\text{tap}[i + 1]$ for $i = \{1, 2, 3\}$. The test fails if any of these comparisons result in $\text{tap}[i]$ having a greater rising-edge delay than $\text{tap}[i + 1]$. If all tests pass, check #2 runs. Recall that the delay between the final tap and input clock (modulo $1 \times \text{UI}$) is ideally zero after locking. However, if the DLL locks on a non-integer value, the delay will be greatest at the final tap. Thus, check #2 compares taps 4 and 8: the test fails if tap 8 has greater delay. After detecting a pass or fail, the FLD block shuts down unless initiated to run again by the system, thus conserving power. At the hardware level, the FLD was implemented using a set of modified phase detector (PD) and CP circuits and static logic.

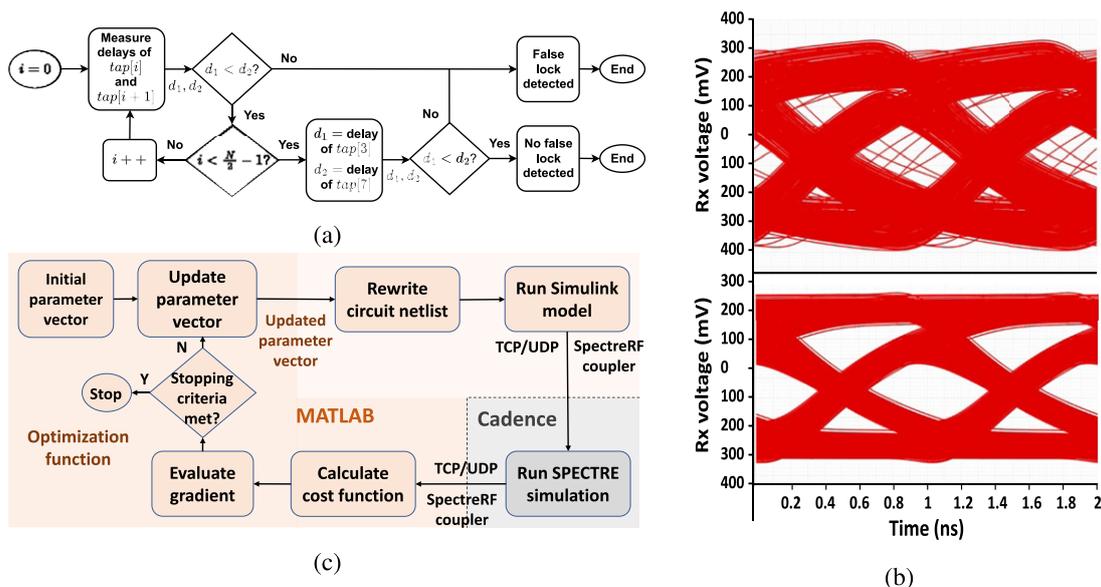


Figure 3. (a) Flowchart of the single-shot false-lock detection (FLD) algorithm. (b) Simulated eye diagrams for data transmission at 1 Gb/s through 3 m of a radio-pure cable. Top: with default parameter values; bottom: with optimized parameter values. (c) Block diagram of the MATLAB-Cadence co-simulation developed for automated optimization of the line driver parameters.

Tap weight control circuitry: Configurable tap weights require control over the drive strength of each tap output. Voltage-mode source-series terminated (SST) driver arrays [6] are used for this purpose. SST drivers are chosen instead of current-mode logic (CML) drivers as they consume 2–4 \times less power for the same output swing. Eight parallel SST slices are provided for each of the three taps, and the user can enable any number of slices per tap. Lastly, three different source resistance options are provided, namely 300, 700, and 1,100 Ω . This additional flexibility allows one to either improve output impedance matching or maximize/minimize output swing as needed. The schematic of a single SST slice with configurable source resistance is shown in figure 2(b).

3 Output waveform optimization

An automated process was developed for finding the optimum pre-emphasis waveform for a given cable (in this case, the same 3 m long radio-pure cable discussed in section 1). As shown in

figure 3(b), the eye diagram can be significantly improved (relative to a conventional FFE with all tap delays equal to $1 \times \text{UI}$) by also optimizing the tap durations. While this optimum was found via an exhaustive search, a more efficient procedure is essential due to the large amount of available parameter permutations. To this end, we developed a Cadence test bench that contains a Verilog-A model of the line driver and a transistor-level model of the SST arrays driving the chosen cable. This test bench outputs results into MATLAB via a Simulink model and a SpectreRF coupler block that enables bidirectional communication between MATLAB and Cadence using TCP/UDP packets. These results are then fed to a MATLAB optimization function that minimizes a cost function given by $\text{FOM} = \sqrt{I_{\text{av}}}/(e_{\text{H}} \times e_{\text{W}})$, where I_{av} is the average current consumption of the driver and e_{H} and e_{W} are the height and width of the eye opening with random input data, respectively. These parameters are combined into a single Figure of Merit (FOM) that is designed to favor a reasonable compromise between power consumption (proportional to I_{av}) and BER (strongly decreasing with e_{H} and e_{W}). Figure 3(c) illustrates the entire co-simulation and optimization flow.

4 Test structure and results

Figure 4(a) shows a die photograph of the fabricated test chip, which integrates the line driver with various clocking and data generation options and a two-wire I^2C control interface. The line driver core occupies $355 \times 363 \mu\text{m}$. The packaged chip was assembled on a PCB for testing. Total power consumption varied from 4–9 mW, lower than earlier work [7–10], while the maximum data rate is limited to 1.1 Gb/s by the locking range of the DLL. Figure 4(b) shows measured output waveforms at 0.8 Gb/s for two different pre-emphasis settings; these are in good agreement with simulations. Figure 4(c) shows the measured eye diagram at 1.0 Gb/s after 0.5 m of RG-142 coaxial cable. The eye is open despite the use of a high-jitter on-chip ring oscillator as the clock source for the tests.

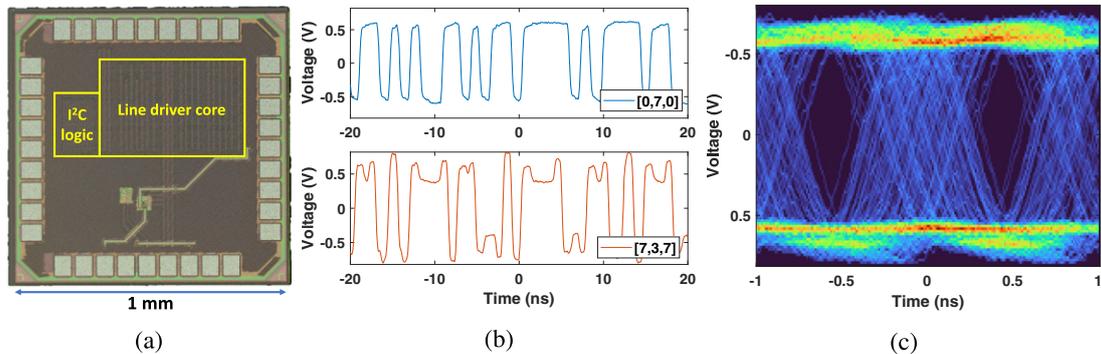


Figure 4. (a) Labeled die photograph of the test chip (total area = 1 mm^2). (b) Measured driver waveforms (PRBS-31 data) at 0.8 Gb/s after 0.5 m of coaxial cable for two different pre-emphasis settings (quantified by the weight vector $w = [w_{\text{pre}}, w_{\text{main}}, w_{\text{post}}]$): $w_{\text{pre}} = w_{\text{post}} = 0$, resulting in no pre-emphasis (top); and $(w_{\text{pre}} = w_{\text{post}}) > w_{\text{main}}$, resulting in significant pre-emphasis (bottom). Both pre- and post-tap time delays were set to $(3/16) \times \text{UI}$. (c) Measured eye diagram (PRBS-31) at 1.0 Gb/s using the same cable, weight vector [5, 5, 7], and time delays of $(3/16) \times \text{UI}$.

5 Conclusion

We have described a highly-configurable line driver circuit in 65 nm CMOS that is particularly suitable for driving high-loss channels. The circuit uses a 3-tap FFE with fine-grained control over both tap weights and time delays, thus allowing the user to adapt the output waveform for a variety of cable types and data rates. Future work will focus on (i) higher data-rate implementations, (ii) replacing the existing DLL with an all-digital version, and (iii) implementing self-monitoring algorithms to enable the driver to autonomously optimize its performance over time.

References

- [1] J.F. Bulzacchelli, *Equalization for electrical links: current design techniques and future directions*, *IEEE Solid-State Circuits Mag.* **7** (2015) 23.
- [2] J.L. Casall, *Driver pre-emphasis for data transmission*, Ph.D. Thesis, Texas Tech University (2001).
- [3] Y. Fu, Z. Wen, & L. Chen, *A 3.75 Gb/s CML output driver with configurable pre-emphasis in 65 nm CMOS technology*, in proceedings of the *2nd International Conference on Communication and Information Processing, ICCIP '16*, Association for Computing Machinery, New York, NY, USA (2016), pp. 200–205.
- [4] B.W. Garlepp, K.S. Donnelly, J. Kim, P.S. Chau, J.L. Zerbe, C. Huang et al., *A portable digital DLL for high-speed CMOS interface circuits*, *IEEE J. Solid-State Circuits* **34** (1999) 632.
- [5] B. Razavi, *The delay-locked loop [a circuit for all seasons]*, *IEEE Solid-State Circuits Mag.* **10** (2018) 9.
- [6] C. Menolfi, T. Toifl, P. Buchmann, M. Kossel, T. Morf, J. Weiss et al., *A 16 Gb/s source-series terminated transmitter in 65 nm CMOS SOI*, in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, (IEEE) (2007), pp. 446–614.
- [7] V. Gromov, V. Zivkovic, M. van Beuzekom, X. Llopart, T. Poikela, J. Buytaert et al., *Development of a low power 5.12 Gbps data serializer and wireline transmitter circuit for the VeloPix chip*, *2015 JINST* **10** C01054.
- [8] K. Iniewski, V. Axelrad, A. Shibkov, A. Balasinski, S. Magierowski, R. Dlugosz et al., *3.125 Gb/s power efficient line driver with 2-level pre-emphasis and 2 kV HBM ESD protection*, in *2005 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE (2005), pp. 1154–1157.
- [9] K. Huang, Z. Wang, X. Zheng, C. Zhang, and Z. Wang, *A 80 mW 40 Gb/s transmitter with automatic serializing time window search and 2-tap pre-emphasis in 65 nm CMOS technology*, *IEEE Trans. Circuits Syst. I* **62** (2015) 1441.
- [10] P. Moreira, S. Baron, S. Biereigel, J. Carvalho, B. Faes, M. Firlej et al., *lpGBT documentation: release*, CERN (2022), <https://cds.cern.ch/record/2809058>.