J Sign Process Syst (2017) 89:163–180
DOI 10.1007/s11265-016-1171-0

Advanced Low Power High Speed Nonlinear Signal Processing: An Analog VLSI Example.

Giuseppe Oliveri[1] · **Mohamad Mostafa**[2] · **Werner G. Teich**[3] · **Jürgen Lindner**[3] · **Hermann Schumacher**[1]

Received: 15 January 2016 / Revised: 12 May 2016 / Accepted: 2 August 2016 / Published online: 20 August 2016

Abstract Despite the progress made in digital signal processing during the last decades, the constraints imposed by high data rate communications are becoming ever more stringent. Moreover mobile communications raised the importance of power consumption for sophisticated algorithms, such as channel equalization or decoding. The strong link existing between computational speed and power consumption suggests an investigation of signal processing with energy efficiency as a prominent design choice. In this work we revisit the topic of signal processing with analog circuits and its potential to increase the energy efficiency. Channel equalization is chosen as an application of nonlinear signal processing, and a vector equalizer based on a recurrent neural network structure is taken as an example to demonstrate what can be achieved with state of the art in VLSI design. We provide an analysis of the equalizer, including the analog circuit design, system-level simulations, and comparisons with the theoretical algorithm. First measurements of our analog VLSI circuit confirm the possibility to achieve an energy requirement of a few pJ/bit, which is an improvement factor of three to four orders of magnitude compared with today's most energy efficient digital circuits.

Keywords Nonlinear signal processing · Energy requirements · Analog VLSI · Vector equalization · Recurrent neural networks

✉ Giuseppe Oliveri
giuseppe.oliveri@uni-ulm.de

Mohamad Mostafa
mohamad.mostafa@dlr.de

Werner G. Teich
werner.teich@uni-ulm.de

Jürgen Lindner
juergen.lindner@uni-ulm.de

Hermann Schumacher
hermann.schumacher@uni-ulm.de

1 Institute of Electron Devices and Circuits, Ulm University, Ulm, Germany

2 Institute of Communications and Navigation, German Aerospace Center, Wessling, Germany

3 Institute of Communications Engineering, Ulm University, Ulm, Germany

1 Introduction

Energy efficiency became an essential aspect in recent years, especially for mobile devices. The most recent Green500 [2] and Top500 [21] rankings show that the most efficient heterogeneous supercomputers can reach peak performance of about five GFLOPS/Watt. Despite the effort to increase this value, the current trend shows an ever-growing difficulty in improving the energy efficiency of digital circuits [3]. As an example, in [14] the authors identify both physical and architectural limitations of modern processors, and predict that those barriers may severely hamper the reduction of the required energy per operation in the future.

We address here alternatives offered by analog circuits. Some authors of earlier work in the field of advanced analog processing concluded that analog systems have the potential to improve the efficiency substantially [12]. Moreover, depending on the application, there might be no need for additional A/D conversion [4]. Analog signal processing is already advantageously used for low-power low-frequency applications [22] and is gaining momentum for applications in the millimeter-wave frequency range [20].

Related to our work are activities in building circuits emulating functions of natural neural networks, e.g. the European "Human Brain" project [13]. In this context it is common to use the term "neuromorphic computing", see also the early work by Mead [15]. In contrast to neuromorphic computing, our focus is more on common signal processing algorithms, usually realized today with digital circuits or digital processors.

To demonstrate how far the energy efficiency can be increased with state of the art VLSI design, in this work, a vector equalizer (VE) [11] serves as an example of a functional block requiring nonlinear processing. The nonlinearity offers the chance to use analog circuits out of their conventional field of linear signal processing, with all its disadvantages, like accumulation of noise and inaccuracies of circuits elements. The corresponding algorithms achieve results as robust as their digital counterparts. This is no surprise, since "digital circuits" are in essence analog circuits with strong nonlinearities.

The paper is organized as follows: in Section 2 we explain the background of our application, while in Section 3 we analyze the structure of the algorithm. In Section 4 we detail the design steps and the components needed to implement a real-valued vector equalizer in SiGe BiCMOS technology. Section 5 expands the theory of the transmission model, of the algorithm, and of the design steps, in order to handle a complex-valued vector equalization. Sections 6 and 7 discuss simulation results of the vector equalizer. Included is the analysis of the penalty in the BER, introduced by a finite resolution of the equalizer's weights. Section 8 highlights the improvement in the energy requirement that is achievable with an analog circuit design, while Section 9 shows measurements on a real chip. Conclusions in Section 10 close the paper.

2 Background

The background for our application is an uncoded digital transmission over radio channels with multiple antennas (multiple-input-multiple-output, MIMO). We assume a linear modulation scheme. Figure 1 shows a model for such a transmission, which is a discrete-time model on a symbol basis. More about this model and its relation to the continuous-time (physical) transmission model can be found in [11]. For a real-valued transmission model the quantities in Fig. 1 are defined as follows:

- k is the discrete-time symbol interval variable;
- $\boldsymbol{x}(k)$ is the transmit symbol vector of length N at symbol interval k. We assume binary phase shift keying (BPSK) modulation, i.e. $x_i(k) \in \{-1, +1\}$ and the transmit symbol alphabet $A_{\boldsymbol{x}}$ contains 2^N possible transmit vectors of length N;
- $\boldsymbol{R}(k)$ is the discrete-time channel matrix on a symbol basis. Its size is $(N \times N)$, it is hermitian and positive semidefinite. We assume that the channel state is known, with this information used to appropriately configure the vector equalizer;
- $\boldsymbol{n}_e(k)$ is a sample function of an additive Gaussian noise vector process with zero mean and covariance matrix given by $\boldsymbol{\Phi}_{n_e n_e}(k) = \frac{N_0}{2} \cdot \boldsymbol{R}(k)$. N_0 is the single-sided noise power spectral density;
- $\tilde{\boldsymbol{x}}(k) = \boldsymbol{R}(k) * \boldsymbol{x}(k) + \boldsymbol{n}_e(k)$ is the received symbol vector. $*$ denotes matrix-vector convolution [11];
- $\hat{\boldsymbol{x}}(k) \in A_{\boldsymbol{x}}$ is the decided vector at the output of the vector equalizer (VE).

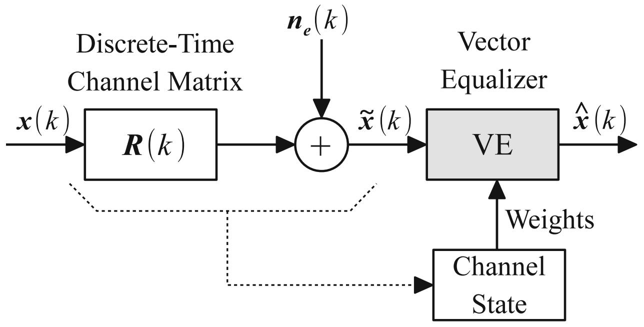


Figure 1 Discrete-time transmission model on a symbol basis for an uncoded transmission with linear modulation over MIMO channels.

$\boldsymbol{R}(k)$ includes the antennas at the transmit and receive sides, the transmit impulses, and the multipath propagation on the radio channel as well. In general it is a sequence of matrices with respect to the symbol interval variable k. Because we assume here no interference between symbol vectors (or "blocks"), k can be omitted and it is sufficient to consider a transmission of isolated vectors. The model in Fig. 1 can then be described mathematically as in Eq. 1. The non-diagonal elements $\boldsymbol{R}_{\backslash d}$ of the channel matrix lead to interference between the components of the transmitted vectors at the receive side. Refer to [11] for more details.

$$\begin{aligned}
\tilde{\boldsymbol{x}} &= \boldsymbol{R} \cdot \boldsymbol{x} + \boldsymbol{n}_e, \\
\tilde{\boldsymbol{x}} &= \underbrace{\boldsymbol{R}_d \cdot \boldsymbol{x}}_{\text{signal}} + \underbrace{\boldsymbol{R}_{\backslash d} \cdot \boldsymbol{x}}_{\text{interference}} + \underbrace{\boldsymbol{n}_e}_{\text{additive noise}}, \\
\boldsymbol{R} &= \underbrace{\boldsymbol{R}_d}_{\text{diagonal elements}} + \underbrace{\boldsymbol{R}_{\backslash d}}_{\text{non-diagonal elements}}.
\end{aligned} \qquad (1)$$

The computational complexity of the optimum vector equalization (i.e. maximum likelihood, ML) grows exponentially with N. Because this will result in an unrealistic number of operations per symbol vector, suboptimum schemes have to be used. Our approach is to use a recurrent neural network (RNN). The VE-RNN does not need a general training algorithm like backpropagation, the entries of $\boldsymbol{R}$ can be measured and directly related to the weights of the RNN.

The application of the RNN as a vector equalizer has been discussed first in the context of multiuser detection for code division multiple access (CDMA) transmission systems [8, 16, 24], see also [5, 23]. It can be shown that this RNN tries to maximize the likelihood function of the optimum VE. In general it converges to a local maximum, but in many cases this local maximum turns out to be close to or identical with the global maximum, see e.g. [6].

3 Continuous-Time Recurrent Neural Network

The VE-RNN discussed before relies on a discrete-time RNN. Analog circuit design requires continuous-time RNNs [17], which have also been known for long time. The dynamic behavior is described by a set of first order nonlinear differential equations as in Eq. 2, where:

- t is the continuous-time evolution time variable;
- $\boldsymbol{e}$ is the external input vector of length N, where N is the length of the transmit symbols, and corresponds to the number of neurons in the RNN;
- $\boldsymbol{T}$ is a diagonal matrix with time constants τ_i on its main diagonal;
- $\boldsymbol{W}$ is a $(N \times N)$ weights matrix with entries $w_{ii'}$;
- $\boldsymbol{W_0}$ is a diagonal matrix with entries w_{i0} on its main diagonal;
- $\boldsymbol{u}(t)$ is the state vector of length N;
- $\boldsymbol{v}(t)$ is the corresponding output vector;
- $\hat{\boldsymbol{v}}(t) = \mathbf{HD}[\boldsymbol{v}(t)]$ is the corresponding hard decision (HD) output vector;
- $\psi_i(\cdot)$ is the i^{th} element-wise activation function.

$$\begin{aligned}
\boldsymbol{T} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} &= -\boldsymbol{u}(t) + \boldsymbol{W} \cdot \boldsymbol{v}(t) + \boldsymbol{W}_0 \cdot \boldsymbol{e}, \\
\boldsymbol{v}(t) &= \boldsymbol{\psi}\,[\boldsymbol{u}(t)] \\
&= [\psi_1[u_1(t)], \psi_2[u_2(t)], \ldots, \psi_N[u_N(t)]]^T, \\
\hat{\boldsymbol{v}}(t) &= \mathbf{HD}\,[\boldsymbol{v}(t)] \\
&= [\mathrm{HD}_1[v_1(t)], \mathrm{HD}_2[v_2(t)], \ldots, \mathrm{HD}_N[v_N(t)]]^T .
\end{aligned} \tag{2}$$

Figure 2 shows a resistance-capacitance structure for a real-valued continuous-time RNN [7]. The stability of this RNN in the sense of Lyapunov has been intensively investigated, e.g. in [10]. $\tau_i = R_i \cdot C_i$ is the time constant of

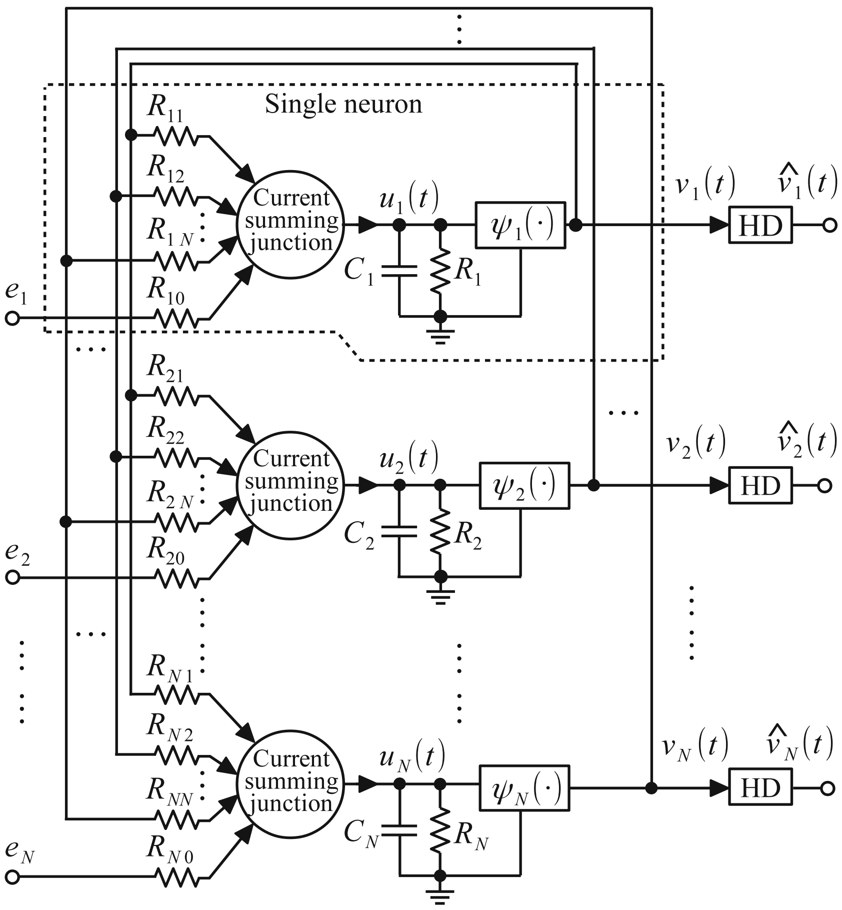


Figure 2 Resistance-capacitance structure of a real-valued recurrent neural network in continuous-time domain.

the i-th neuron. The weights in Eq. 2 above are related to the resistors in Fig. 2 by normalization as follows: $w_{ii'} = \frac{R_i}{R_{ii'}}$, $w_{i0} = \frac{R_i}{R_{i0}}$. To distinguish between resistors and channel matrix, the symbol for the channel matrix is bold.

3.1 Equalization based on Continuous-Time Recurrent Neural Networks

The vector equalizer discussed in Section 2 works on a symbol basis and is discrete in time. This means that the clock for the VE is kT_s, with k being the discrete time variable and T_s the symbol interval for the digital transmission. The RNN in Fig. 2 works equally with the parallel input of a symbol vector, but is continuous in time. In order to connect the vector equalizer of Fig. 1 with the network of Fig. 2, the following conditions must be fulfilled, cf. Eqs. 1, and 2:

- The continuous-time RNN requires a minimum interval of time to perform the equalization of a vector. This time slot is here defined as the total equalization time t_{equ}. It follows that the symbol interval T_s for the digital transmission is constrained by the equalization time, i.e. $T_s \geq t_{\mathrm{equ}}$;
- $\boldsymbol{e} = \tilde{\boldsymbol{x}}$: the external input vector of the RNN represents the received symbol vector;
- $\hat{\boldsymbol{v}}(t_{\mathrm{equ}}) = \hat{\boldsymbol{x}}$: the output vector of the RNN – after an equalization is performed and after the hard decision – is coincident with the decided vector of the discrete-time VE;
- $\boldsymbol{W}_0 = \boldsymbol{R}_d^{-1}$: the weights for the external inputs of the RNN are computed from the diagonal elements of the channel matrix. In the following we consider only normalized channel matrices, i.e. calling $\boldsymbol{I}$ the identity matrix, $\boldsymbol{R}_d^{-1} = \boldsymbol{R}_d = \boldsymbol{I}$;
- $\boldsymbol{W} = \boldsymbol{I} - \boldsymbol{R}_d^{-1} \cdot \boldsymbol{R}$: feedback paths between the different neurons are related to the intersymbol interference and are taken from the channel matrix. Under the hypothesis of normalized channel matrices, $\boldsymbol{W} = -\boldsymbol{R}_{\backslash d}$, i.e. the weight matrix corresponds to the additive inversion of the non-diagonal elements of the channel matrix, with zeros on the main diagonal (neurons with no self feedback);
- $\tau_1, \tau_2, \cdots, \tau_N = \tau$: all time constants have the same value;
- $\psi_1[(\cdot)_1], \psi_2[(\cdot)_2], \cdots \psi_N[(\cdot)_N] = \psi[(\cdot)_i]$: all neurons possess the same activation function. The activation function applied to a generic element of the state vector is defined as a hyperbolic tangent: $\psi[u_i(t)] = \alpha \cdot \tanh(\beta \cdot u_i(t))$. Here, $\alpha = 1$ V gives the dimension of Volts to the activation function, while β [V^{-1}] is a positive variable which must be optimized for achieving best performance. From our simulations, the condition to fulfill is $\beta \geq 3$ V^{-1};
- $\mathrm{HD}[v_i(t)] = \mathrm{sign}(v_i(t))$: the hard decision applied to a vector element has the codomain $\{-1, +1\}$.

With those assumptions, and applying the update rule of the first Euler method, Eq. 2 can be simulated on a digital computer:

$$\begin{aligned} \boldsymbol{u}(l+1) &= \left\{1 - \frac{\Delta t}{\tau}\right\} \boldsymbol{u}(l) + \frac{\Delta t}{\tau} \left\{\boldsymbol{W} \cdot \boldsymbol{v}(l) + \boldsymbol{e}\right\}, \\ \boldsymbol{v}(l) &= \boldsymbol{\psi}[\boldsymbol{u}(l)]. \end{aligned} \tag{3}$$

l is now a discrete time variable, connected to the temporal evolution of the network. Δt is the sampling step, which should be as small as possible. For our simulations we assume $\tau/\Delta t = 10$. Since the RNN is Lyapunov stable, $\hat{\boldsymbol{v}}(t)$ reaches an equilibrium state after the evolution time, i.e. for $l = t_{\mathrm{equ}}$. The above stated conditions are valid for BPSK, but can be generalized by combining the results of [10, 18, 19].

3.2 Scaling

The dynamic systems of Eqs. 2 and 3 must fit the limited voltage swings that an analog circuit can handle. It is thus convenient to introduce a dimensionless scaling factor S:

$$\boldsymbol{u}'(t) = S \cdot \boldsymbol{u}(t), \quad \boldsymbol{v}'(t) = S \cdot \boldsymbol{v}(t), \quad \boldsymbol{e}' = S \cdot \boldsymbol{e}. \tag{4}$$

The scaled set of equations, describing the dynamical behavior of the continuous-time RNN, can finally be written as:

$$\begin{aligned} \boldsymbol{T} \cdot \frac{\mathrm{d}\boldsymbol{u}'(t)}{\mathrm{d}t} &= -\boldsymbol{u}'(t) + \boldsymbol{W} \cdot \boldsymbol{v}'(t) + \boldsymbol{e}', \\ \boldsymbol{v}'(t) &= S \cdot \boldsymbol{\psi}\left[\frac{\boldsymbol{u}'(t)}{S}\right]. \end{aligned} \tag{5}$$

4 Real-Valued Equalizer

Potential implementations of a RNN cover a wide variety of solutions, from a discrete-time RNN implemented with field programmable gate arrays (FPGA) – as in [23] – to continuous-time analog hardware – as in [1] and [9]. Since here we focus on speed of operation and power efficiency, analog VLSI design and the continuous-time RNN will be the topic.

The resistance-capacitance model of Section 3 offers a very compact and descriptive view of a continuous-time RNN. It is useful for the stability analysis and for the algorithm definition, but is not of practical realization, the main issue being the presence of tunable resistors that must cover both a positive and a negative range with very fine resolution, according to the weights configuration.

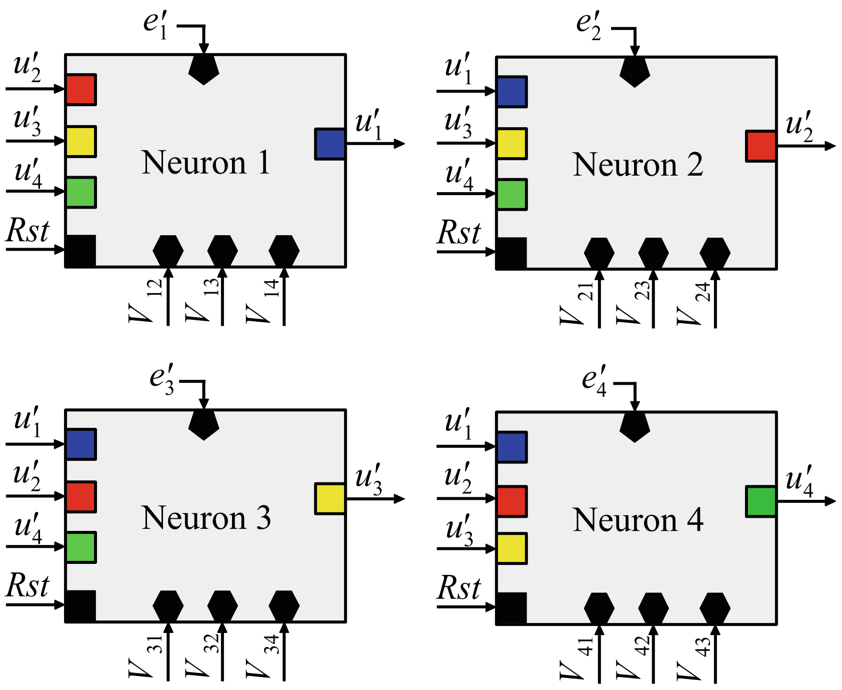


Figure 3 System-level view of a real-valued equalizer, composed by $N = 4$ neurons.

4.1 Circuit Design

The system-level view of the actual equalizer is presented in Fig. 3. It refers to a real-valued vector equalizer with $N = 4$ neurons, designed in IHP 0.25 µm SiGe BiCMOS technology (SG25H3). Figure 4 shows the functional view of one single neuron, with Fig. 5 finally presenting the schematic circuit. For each neuron the input/output ports are expressed in Volts. The circuit is fully differential and the bipolar junction transistors (BJTs) are assumed ideally matched.

The first set of inputs that the i^{th} neuron takes is represented by the feedback inner state elements u'_k from all other neurons in the RNN ($k \in [1, ..., N], k \neq i$). The activation function (φ in Fig. 4) is realized with a differential transconductance (TC) stage (transistors Q_1 and Q_2 in Fig. 5), biased with a tail current I_t, generated through a current mirror. This results in a large-signal output current as follows:

$$\begin{aligned} I_k &= \varphi\left[u'_k\right] \in [\text{-}I_t, I_t] \\ &\approx I_t \cdot \tanh\left(\frac{u'_k}{2 \cdot V_t}\right) \end{aligned} \tag{6}$$

Using a four quadrant analog multiplier (Gilbert cell), each feedback current I_k is multiplied by a weight w_{ik} in the range $[-1, +1]$. The value of w_{ik} is set to the corresponding entry of the channel matrix. The Gilbert cell (TC stage f in Fig. 4, quartet of transistors Q_3-Q_6 in Fig. 5) is controlled by the voltage V_{ik} and a constant reference voltage V_{ref}. An attenuator – in the form of a common emitter amplifier with gain lower than unity – allows

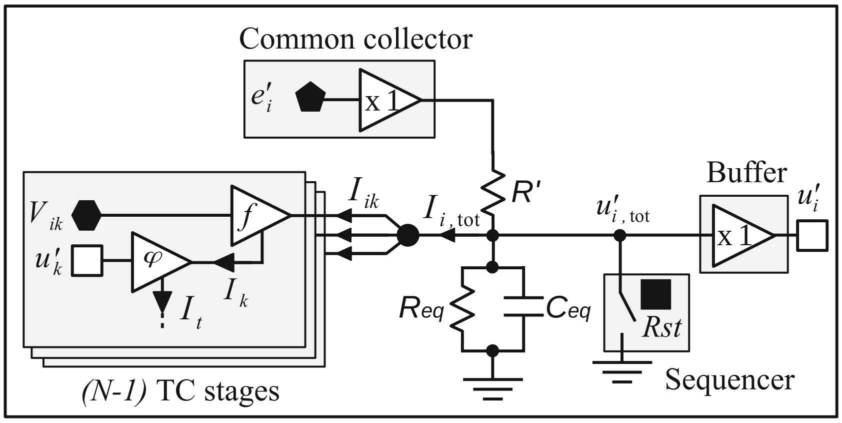


Figure 4 Functional blocks of the i^{th} neuron for a real-valued equalizer. R_{eq} and C_{eq} are an equivalent parasitic impedance between node $u'_{i,\text{tot}}$ and ground.

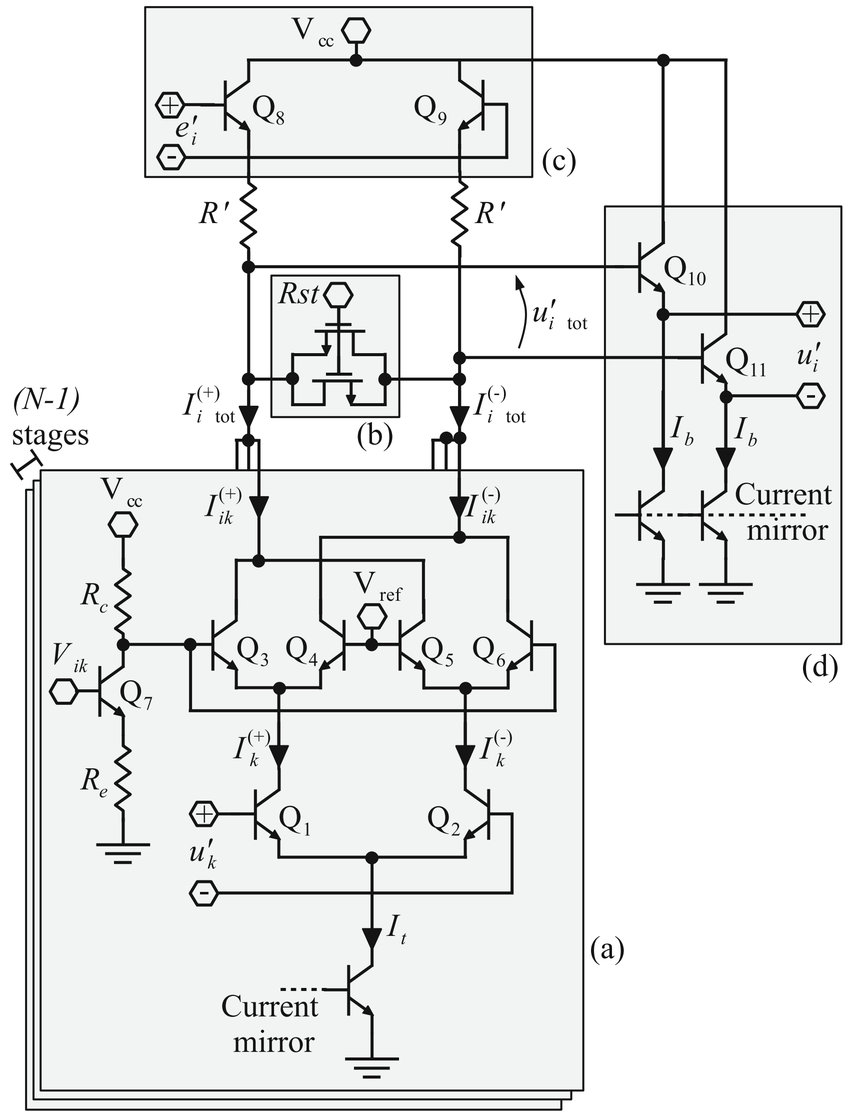


Figure 5 Circuit schematic for the i^{th} neuron: (a) Differential pair for the generation of the activation function $\varphi(\cdot)$, and Gilbert cell used as four quadrant analog multiplier; (b) MOSFET switch used as a sequencer; (c) Common collector stage for the external input; (d) Buffers.

each individual feedback current I_{ik} to be tuned with fine resolution:

$$\begin{aligned} I_{ik} &= f\,[V_{ik}] \in [\text{-}I_k, I_k] \\ &= w_{ik} \cdot I_k \end{aligned} \tag{7}$$

Connecting the output branches of the Gilbert cells, the total weighted feedback current for the i^{th} neuron $I_{i,\text{tot}}$ is obtained by applying Kirchhoff's current law:

$$I_{i,\text{tot}} = \sum_{\substack{k=1 \\ k \neq i}}^{N} w_{ik} \cdot I_{ik} \tag{8}$$

Two common collector transistors (Q_8, Q_9), biased by the same current $I_{i,\text{tot}}$ used for the summation of the feedback currents, create an additional differential voltage drop on $u'_{i,\text{tot}}$, proportional to the correspondent external input e'_i. Two additional buffer stages replicate the differential voltage $u'_{i,\text{tot}}$ into u'_i. The circuit is provided with an integrated metal-oxide-semiconductor field-effect transistor (MOSFET) switch. This switch acts as a sequencer. Its importance will be clarified in Section 4.2.

Considering the MOSFET switch in off state, we make the assumption of an equivalent low-pass behavior with time constant $\tau = R_{eq} \cdot C_{eq}$, where R_{eq} and C_{eq} mainly include the combination of the output impedance of the Gilbert cells, of the load resistor R', of the input impedance of the buffer stages loaded by the subsequent differential pairs, and of the parasitic capacitors of the MOSFET switch. Layout losses of the interconnections also play a role. In other words, to fully exploit the speed of the BJTs, in this architecture an equivalent low-pass filter, lumped at node u'_i, is used in lieu of an external low-pass filter. This allows for the minimization of the the time constant τ – that is the basis for scaling of the evolution time t. The validation of this hypothesis, both in a simulation environment and in a measurement setup on the real chip, is presented in Section 9.

The nodal analysis on $u_i'(t)$ gives:

$$\tau \cdot \frac{\mathrm{d}u_i'(t)}{\mathrm{d}t} = -u_i'(t) - R_{eq} \cdot \sum_{\substack{k=1 \\ k\neq i}}^{N} w_{ik} \cdot I_i + e_i' \tag{9}$$

$u_i'(t)$ represents the inner state that will be distributed to the other $N-1$ neurons in the network. Note also that – according to Eq. 2 – the sign of $\boldsymbol{u}$ coincides with the sign of $\boldsymbol{v}$, and can thus be used to perform a hard decision at the end of an equalization. Generalizing, the dynamics of the analog neural network can be finally written in vector form:

$$\boldsymbol{T} \cdot \frac{\mathrm{d}\boldsymbol{u}'(t)}{\mathrm{d}t} = -\boldsymbol{u}'(t) + \boldsymbol{W} \cdot \boldsymbol{v}'(t) + \boldsymbol{e}',$$
$$\boldsymbol{v}'(t) \approx R_{eq} \cdot I_t \cdot \mathbf{tanh}\left(\frac{\boldsymbol{u}'(t)}{2 \cdot V_t}\right). \tag{10}$$

The correspondence between the circuit model of Fig. 3 and the resistance-capacitance model of Fig. 2 is validated, if the following positions hold:

$$S = (R_{eq} \cdot I_t)/\alpha,$$
$$\beta = S/(2 \cdot V_t). \tag{11}$$

The scaling factor of the circuit depends on the tail current and on the equivalent resistive load. It also influences the slope of the hyperbolic tangent at the origin, i.e. for a null differential voltage $\boldsymbol{u}' = \boldsymbol{0}$. Using the values provided in Table 1, Eq. 10 is linked to Eq. 5, with scaling factor $S = 0.2$ and slope of the hyperbolic tangent $\beta = 3.87$ 1/V.

4.2 The Reset (Rst) Function

The VE-RNN is a dynamic system, where the network evolves from an initial state (a saddle equilibrium point) to a stable state, following a non-monotonic trajectory in the state-space according to the set of equations in Eq. 10. Given a sequence of input vectors $\boldsymbol{e}'$, Fig. 6 details how the VE reaches stability (and consequently when the output vector can be considered "valid") and how it is possible to discard the memory of a previous equalization.

The evolution time t_{ev} is defined as the time slot granted to the circuit, necessary to reach a stable state. External inputs are applied only during this time slot. Before the next input is applied, it is crucial that the network returns – and stays pinned – to a predefined initial state.

A reset time t_{Rst} can be defined as the time granted to the circuit to return to the initial state after a vector equalization. In our implementation the inner state $\boldsymbol{u}'$ is forced to return to zero, an unbiased starting point, equidistant from the 2^N possible stable states. From the circuit point of view this effect can be compared to a capacitor which must be fully discharged at the beginning of the equalization, in order to avoid a "memory" of the previous equalization.

Rst is the reset signal, indicating if either an equalization is running or the circuit is resetting. Rst acts on the gate port of a MOSFET switch (the sequencer, in Figs. 4 and 5). When high, Rst switches the two NMOS FETs into a low channel resistance state, short circuiting the differential internal state $\boldsymbol{u'}$. The width of the MOSFETs is chosen as a tradeoff between the parasitic capacitance seen with the switch in off state (to be minimized, since it strongly contributes to the increase of the equivalent τ) and the equivalent resistance seen in on-state (to be minimized, since it represents the "goodness" of the short circuit).

For best performance, i.e. highest throughput, both t_{ev} and t_{Rst} can be adjusted and minimized for each channel matrix. This is translated in the statistical optimization of the evolution time $t_{\mathrm{ev,min}}$ and of the reset time $t_{\mathrm{Rst,min}}$, as shown in Section 6.

Table 1 Summary of Main Circuit Parameters.

Parameter	Value	Unit	Note
R'	900	Ω	Load resistor
I_t	222	μA	Tail current
S	0.2		Scaling factor
β	3.87	V^{-1}	Hyperbolic tangent's slope at the origin
τ	42	ps	Equivalent time constant
A_{chip}	0.68	mm^2	Chip area
A_{act}	0.087	mm^2	Active area
Cnt	171		Transistor count
W_{st}	35	mW	VE power consumption

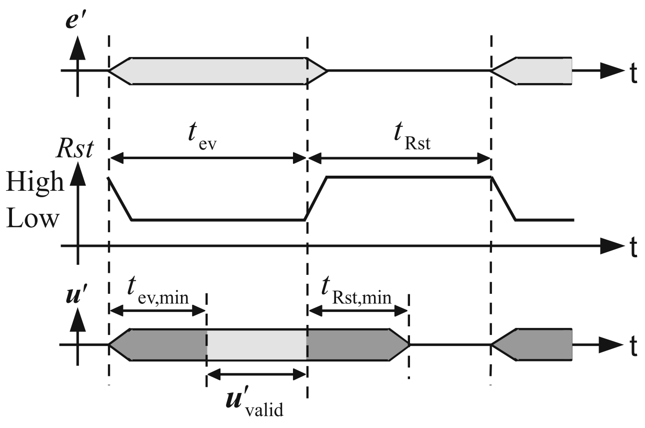


Figure 6 Time domain evolution of an equalization. Because of the iterative nature of the algorithm, the outputs are "valid" after a minimum evolution time. A minimum reset time is also necessary before a new equalization.

5 Complex-Valued Equalization

5.1 Theory of Operation

The background and the dynamical behavior of a VE-RNN can be extended to include quadrature phase shift keying (QPSK) modulation. We introduce subscripts "p" and "q" to refer to in-phase and quadrature components of the symbols, of the noise, and of the matrices. The discrete-time model of Fig. 1 and Eq. 1 still holds with the following assumptions:

- $\boldsymbol{x}_\mathrm{c} = \boldsymbol{x}_\mathrm{p} + j\boldsymbol{x}_\mathrm{q}$ is the complex-valued transmit symbol vector. $x_{\mathrm{c},i} \in \{\pm 1 \pm j\}$ and the transmit symbol alphabet $A_{\boldsymbol{x}_\mathrm{c}}$ contains 4^N possible transmit vectors. The same complex notation is applied to the received symbol vector $\tilde{\boldsymbol{x}}_\mathrm{c}$ and to the decided vector at the output of the equalizer $\hat{\boldsymbol{x}}_\mathrm{c}$;
- $\boldsymbol{R}_\mathrm{c} = \boldsymbol{R}_\mathrm{p} + j\boldsymbol{R}_\mathrm{q}$ is the complex-valued discrete-time channel matrix on symbol basis;
- $\boldsymbol{n}_{\mathrm{c},e}$ is the complex-valued additive Gaussian noise.

A complex-valued continuous-time RNN is still described by a set of first order nonlinear differential equations – cf. Eq. 2 – with the following modifications:

- $\boldsymbol{e}_\mathrm{c} = \boldsymbol{e}_\mathrm{p} + j\boldsymbol{e}_\mathrm{q}$ is the complex-valued external input vector. Using the same notation, $\boldsymbol{u}_\mathrm{c}(t)$, $\boldsymbol{v}_\mathrm{c}(t)$, and $\hat{\boldsymbol{v}}_\mathrm{c}(t)$ are the complex-valued state vector, output vector, and hard-decision vector, respectively.
- the weight matrix is now $\boldsymbol{W}_\mathrm{c}$ and has complex-valued entries $w_{\mathrm{c},ii'} = w_{\mathrm{p},ii'} + j w_{\mathrm{q},ii'}$;
- $\boldsymbol{\psi}_\mathrm{c}[\boldsymbol{u}_\mathrm{c}] = \boldsymbol{\psi}[\boldsymbol{u}_\mathrm{p}] + j\boldsymbol{\psi}[\boldsymbol{u}_\mathrm{q}]$: the complex-valued activation function is obtained by independently applying the real-valued activation function $\boldsymbol{\psi}$ to the in-phase and quadrature components of the state vector. The same procedure is valid for the complex-valued hard decision function on the output vector: $\mathbf{HD}_\mathrm{c}[\boldsymbol{v}_\mathrm{c}] = \mathbf{HD}[\boldsymbol{v}_\mathrm{p}] + j\mathbf{HD}[\boldsymbol{v}_\mathrm{q}]$.

The resistance-capacitance model of Fig. 2 must be extended to handle complex-valued quantities. If all the variables are expanded in terms of their real and imaginary part, and considering the scaling of the system, the set of equations in (5) can finally be separated as follows:

$$\begin{aligned}
\boldsymbol{T}\frac{\mathrm{d}\boldsymbol{u}'_\mathrm{p}(t)}{\mathrm{d}t} &= -\boldsymbol{u}'_\mathrm{p}(t) + \left[\boldsymbol{W}_\mathrm{p}\boldsymbol{v}'_\mathrm{p}(t) - \boldsymbol{W}_\mathrm{q}\boldsymbol{v}'_\mathrm{q}(t)\right] + \boldsymbol{e}'_\mathrm{p},\\
\boldsymbol{T}\frac{\mathrm{d}\boldsymbol{u}'_\mathrm{q}(t)}{\mathrm{d}t} &= -\boldsymbol{u}'_\mathrm{q}(t) + \left[\boldsymbol{W}_\mathrm{p}\boldsymbol{v}'_\mathrm{q}(t) + \boldsymbol{W}_\mathrm{q}\boldsymbol{v}'_\mathrm{p}(t)\right] + \boldsymbol{e}'_\mathrm{q},\\
\boldsymbol{v}'_\mathrm{p}(t) &= S\cdot\boldsymbol{\psi}\left[\frac{\boldsymbol{u}'_\mathrm{p}(t)}{S}\right],\\
\boldsymbol{v}'_\mathrm{q}(t) &= S\cdot\boldsymbol{\psi}\left[\frac{\boldsymbol{u}'_\mathrm{q}(t)}{S}\right].
\end{aligned} \tag{12}$$

As shown in Fig. 7, a complex-valued recurrent neural network with $N = 4$ neurons is equivalent to a real-valued recurrent neural network of $N = 8$ neurons, split in two sub-networks of $N = 4$ neurons. Each sub-network accepts in-phase and quadrature part of the received symbol, and will produce the in-phase and quadrature part of the decided vector, respectively. The N complex-valued feedback contributions are mapped into $2 \cdot N$ real feedback paths for each sub-network. This is the approach used

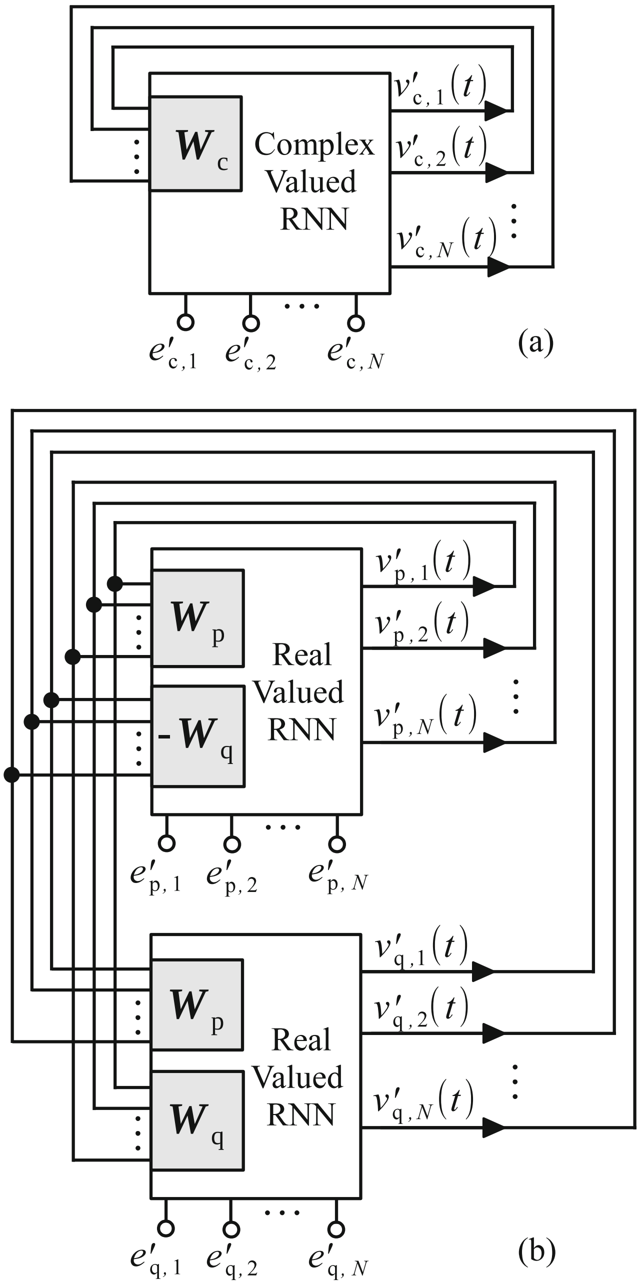


Figure 7 Equivalence between a complex-valued recurrent neural network with N neurons (a) and the interconnection of two real-valued neural sub-networks (b).

in this work to design the analog complex-valued vector equalizer.

5.2 Circuit Design

The system-level view of the complex equalizer is shown in Fig. 8. The i^{th} neuron takes the i^{th} complex-valued element of the external input vector $\boldsymbol{e}'_{\mathrm{c}}$ and outputs the i^{th} complex-valued element of the internal state vector $\boldsymbol{u}'_{\mathrm{c}}(t)$. Each neuron also accepts the complex-valued inner state elements coming from the other neurons in the network. The voltage $V_{\mathrm{p},ik}$ covers the real part of the interference from neuron k to neuron i. Correspondingly, $V_{\mathrm{q},ik}$ is used for the imaginary part of the interference. All the neurons possess the reset (Rst) input port.

The functional view of the single neuron is shown in Fig. 9, and the schematic is detailed in Fig. 10. The mode of operation is based on TC stages. According to the variable separation in Eq. 12, each neuron is formed by two twin subsystems, with each subsystem requiring $2 \cdot (N - 1)$ transconductance stages to generate the weighted feedback currents.

The first TC stage ("f" in Fig. 9) is formed by a differential pair (Q_2, Q_5, and the two resistors for the emitter degeneration in Fig. 10), biased with a tail current I_t. The tail current is generated through a current mirror, not shown in the figure. Transistors Q_3 and Q_4 represent the sequencer for the Rst function.

With respect to the differential pair Q_2-Q_5, the sequencer is in a "winner takes all" configuration. During the evolution time, Q_3 and Q_4 are biased with a base voltage lower than both Q_2 and Q_5. Therefore they are in off state, and the differential pair Q_2-Q_5 generates a differential current $I_{\mathrm{p},ik}$ function of the differential voltage $V_{\mathrm{p},ik}$. When $V_{\mathrm{q},ik}$ is concerned, the current is denoted as $I_{\mathrm{q},ik}$. With Q_3 and Q_4 off (evolution time), the input/output relations can be expressed as follows:

$$\begin{aligned} I_{\mathrm{p},ik} &= f\left[V_{\mathrm{p},ik}\right] \in [\text{-}I_t, I_t] \\ &= w_{\mathrm{p},ik} \cdot I_t \\ I_{\mathrm{q},ik} &= f\left[V_{\mathrm{q},ik}\right] \in [\text{-}I_t, I_t] \\ &= w_{\mathrm{q},ik} \cdot I_t \end{aligned} \tag{13}$$

During the reset time the base voltage of Q_3 and Q_4 is higher than the base of both Q_2 and Q_5. The bias

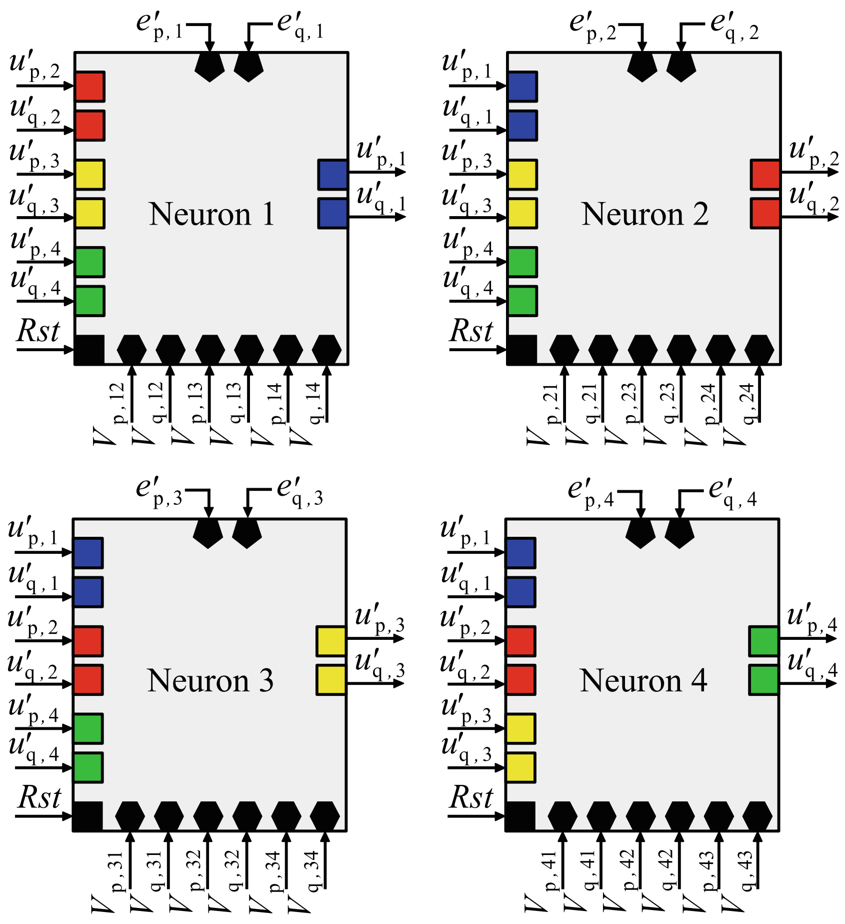


Figure 8 System-level view of a complex-valued equalizer composed by $N = 4$ neurons. The definitions of the external input, inner state, and voltages V_{ik} are kept unchanged. Subscripts "p" and "q" account for the real and imaginary parts of the values, respectively.

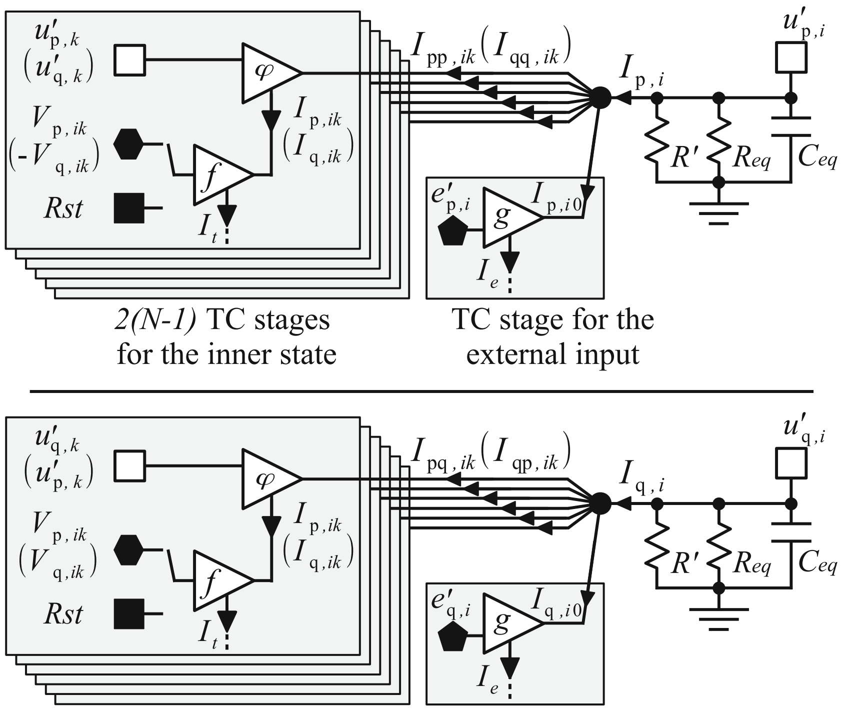


Figure 9 Functional blocks of a neuron, as part of a $N = 4$ complex-valued equalizer. The topology includes two twin subsystems. R_{eq} and C_{eq} represent an equivalent parasitic low-pass filter connected to the node $u'_{p,i}$ (or $u'_{q,i}$).

current I_t flows only through Q_3 and Q_4, equally split, and independent of $V_{p,ik}(V_{q,ik})$. With Q_3 and Q_4 on, the differential currents $I_{p,ik}$ and $I_{q,ik}$ become zero:

$$I_{p,ik} = I_{q,ik} = 0, \ \ \forall (V_{p,ik}, V_{q,ik}) \ \ \text{(Reset time)}$$

The differential current $I_{p,ik}$ $(I_{q,ik})$ biases a second TC stage (φ in Fig. 9), formed by two differential pairs in Gilbert cell configuration (Q_6, Q_7, Q_8, Q_9 in Fig. 10). The large signal output current is a four-quadrant multiplication of the inner state $u'_{p,k}$ $(u'_{q,k})$. Depending on the

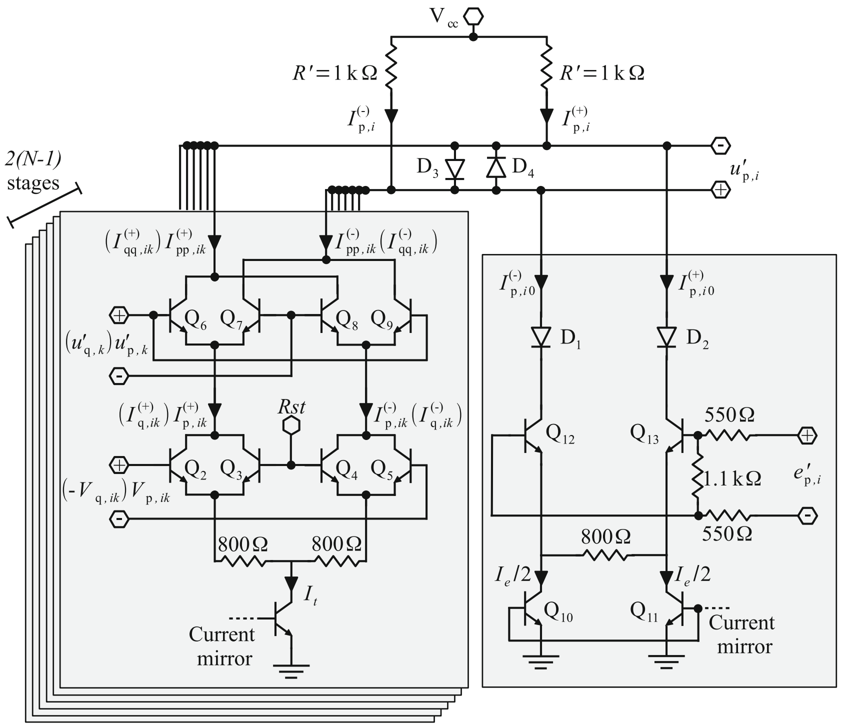


Figure 10 Circuit schematic of a fully-differential subsystem, as part of a complex-valued vector equalizer.

subsystem under consideration, the input/output relations can be expressed as follows:

$$\begin{cases} I_{\mathrm{pp},ik} = \varphi\left[V_{\mathrm{p},ik}, u'_{\mathrm{p},k}\right] \in \left[-I_{\mathrm{p},ik}, I_{\mathrm{p},ik}\right] \\ \quad = w_{\mathrm{p},ik} \cdot I_t \cdot \tanh\left(\frac{u'_{\mathrm{p},k}}{2 \cdot V_t}\right) \\ I_{\mathrm{qq},ik} = \varphi\left[V_{\mathrm{q},ik}, u'_{\mathrm{q},k}\right] \in \left[-I_{\mathrm{q},ik}, I_{\mathrm{q},ik}\right] \\ \quad = w_{\mathrm{q},ik} \cdot I_t \cdot \tanh\left(\frac{u'_{\mathrm{q},k}}{2 \cdot V_t}\right) \end{cases} \tag{14}$$

$$\begin{cases} I_{\mathrm{pq},ik} = \varphi\left[V_{\mathrm{p},ik}, u'_{\mathrm{q},k}\right] \in \left[-I_{\mathrm{p},ik}, I_{\mathrm{p},ik}\right] \\ \quad = w_{\mathrm{p},ik} \cdot I_t \cdot \tanh\left(\frac{u'_{\mathrm{q},k}}{2 \cdot V_t}\right) \\ I_{\mathrm{qp},ik} = \varphi\left[V_{\mathrm{q},ik}, u'_{\mathrm{p},k}\right] \in \left[-I_{\mathrm{q},ik}, I_{\mathrm{q},ik}\right] \\ \quad = w_{\mathrm{q},ik} \cdot I_t \cdot \tanh\left(\frac{u'_{\mathrm{p},k}}{2 \cdot V_t}\right) \end{cases} \tag{15}$$

An additional transconductance stage (g in Fig. 9, Q_{12} and Q_{13} in Fig. 10) is used to generate a current $I_{p,i0}$ (or $I_{q,i0}$), proportional to the in-phase (or quadrature) i^{th} element of the external input $\boldsymbol{e}'_{\mathrm{c}}$. This stage is optimized to provide a linear large signal output characteristic (constant transconductance G) among the range of interest:

$$\begin{aligned} I_{\mathrm{p},i0} &= g\left[e'_{\mathrm{p},i}\right] \in [-I_e, I_e] \\ &= G \cdot e'_{\mathrm{p},i} \\ I_{\mathrm{q},i0} &= g\left[e'_{\mathrm{q},i}\right] \in [-I_e, I_e] \\ &= G \cdot e'_{\mathrm{q},i} \end{aligned} \tag{16}$$

Connecting the output branches of the Gilbert cells, the total in-phase differential currents ($I_{\mathrm{p},i}$) for the i^{th} neuron can finally be computed as in Eq. 17. For the twin subsystem, the total quadrature differential current of the i^{th} neuron ($I_{\mathrm{q},i}$) is given in Eq. 18.

$$\begin{aligned} I_{\mathrm{p},i} = & I_t \cdot \sum_{\substack{k=1 \\ k \neq i}}^{N} \left[w_{\mathrm{p},ik} \cdot \tanh\left(\frac{u'_{\mathrm{p},k}}{2 \cdot V_t}\right)\right] \\ & -I_t \cdot \sum_{\substack{k=1 \\ k \neq i}}^{N} \left[w_{\mathrm{q},ik} \cdot \tanh\left(\frac{u'_{\mathrm{q},k}}{2 \cdot V_t}\right)\right] \\ & +G \cdot e'_{\mathrm{p},i} \end{aligned} \tag{17}$$

$$\begin{aligned} I_{\mathrm{q},i} = & I_t \cdot \sum_{\substack{k=1 \\ k \neq i}}^{N} \left[w_{\mathrm{p},ik} \cdot \tanh\left(\frac{u'_{\mathrm{q},k}}{2 \cdot V_t}\right)\right] \\ & +I_t \cdot \sum_{\substack{k=1 \\ k \neq i}}^{N} \left[w_{\mathrm{q},ik} \cdot \tanh\left(\frac{u'_{\mathrm{p},k}}{2 \cdot V_t}\right)\right] \\ & +G \cdot e'_{\mathrm{q},i} \end{aligned} \tag{18}$$

As for the real-valued equalizer, an equivalent parasitic low-pass filter can be defined, mainly composed of a physical load resistor R', and the combination of (i) the output impedance of the Gilbert cells connected to the node $u'_{\mathrm{p},i}$ (or $u'_{\mathrm{q},i}$), and (ii) of the input impedance of the transconductance stages, driven by $u'_{\mathrm{p},i}$ (or $u'_{\mathrm{q},i}$). Defining $\tau \equiv R_{eq} \cdot C_{eq}$, and choosing $G = 1/R_{eq}$, the nodal analysis on nodes $u'_{\mathrm{p},i}$ and $u'_{\mathrm{q},i}$ respectively gives:

$$\begin{aligned} \tau \cdot \frac{\mathrm{d}u'_{\mathrm{p},i}(t)}{\mathrm{d}t} &= -u'_{\mathrm{p},i}(t) - R_{eq} \cdot I_{\mathrm{p},i}(t) + e'_{\mathrm{p},i}, \\ \tau \cdot \frac{\mathrm{d}u'_{\mathrm{q},i}(t)}{\mathrm{d}t} &= -u'_{\mathrm{q},i}(t) - R_{eq} \cdot I_{\mathrm{q},i}(t) + e'_{\mathrm{q},i}, \end{aligned} \tag{19}$$

When written in vector form, the set of equations in (19) corresponds to Eq. 12, if $S \cdot \alpha = R_{eq} \cdot I_t$ and $\beta = S/(2 \cdot V_t)$. Finally, the diodes D_1 and D_2 in Fig. 10 are used as voltage shifters, while the diodes D_3 and D_4 are voltage limiting circuits.

6 Simulations Results

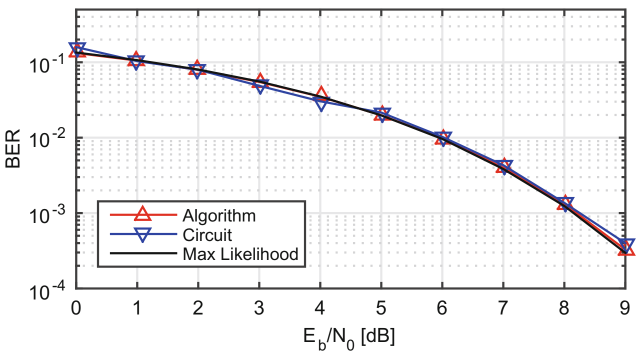
In this section two types of simulations, run on general-purpose computers, of the continuous-time RNN equalizer are compared and shortly discussed: one represents Eq. 3 simulated in Matlab, and labeled in the following as "algorithm". The second is a circuit-based simulation, performed in Keysight ADS, and labeled as "circuit". The modulation is BPSK, and the number of neurons is four. Here results are presented for two channel matrices:

$$\boldsymbol{R}_m = \begin{bmatrix} 1 & +0.60 & +0.60 & +0.60 \\ +0.60 & 1 & +0.60 & +0.60 \\ +0.60 & +0.60 & 1 & +0.60 \\ +0.60 & +0.60 & +0.60 & 1 \end{bmatrix}$$

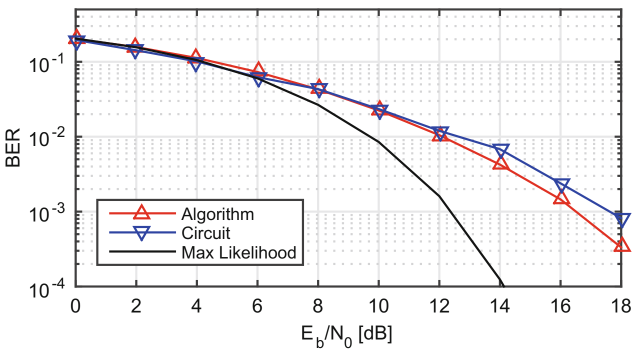
$$\boldsymbol{R}_h = \begin{bmatrix} 1 & +0.85 & +0.66 & -0.67 \\ +0.85 & 1 & +0.85 & -0.79 \\ +0.66 & +0.85 & 1 & -0.89 \\ -0.67 & -0.79 & -0.89 & 1 \end{bmatrix}$$

They are representative of channels with moderate ($\boldsymbol{R}_m$) and high ($\boldsymbol{R}_h$) crosstalk (interference between vector components), respectively. A pseudo-random sequence of symbol vectors was generated and multiplied with one of these matrices. Gaussian noise vectors according to the E_b/N_0 signal-to-noise ratio were then added. E_b is the average energy per bit. For the circuit simulations all the applied signals have a rise/fall time of $t_{r/f} = \tau/3$.

Figure 11 shows the good agreement of the bit error rate (BER) curves between the algorithm and the circuit simulation. Since the vector equalization based on RNNs is a suboptimum scheme, the Maximum Likelihood curves are also shown for reference.



(a) BER vs. E_b/N_0 for moderate crosstalk $\boldsymbol{R}_m$



(b) BER vs. E_b/N_0 for high crosstalk $\boldsymbol{R}_h$

Figure 11 BER evaluation of the continuous-time RNN equalizer, including circuit and algorithm simulations. Maximum Likelihood algorithm shown for reference.

Because of the iterative nature of the RNN algorithm, the BER is – additionally to E_b/N_0 – a function of the evolution (t_{ev}) and reset (t_{Rst}) time. Given a channel matrix, a BER surface is obtained by sweeping the evolution and reset time, and keeping the signal-to-noise ratio constant, as shown in Fig. 12 for a circuit-based simulation with interference given by $\boldsymbol{R_h}$ and $E_b/N_0 = 18$ dB. Following this optimization procedure, and considering the region in which

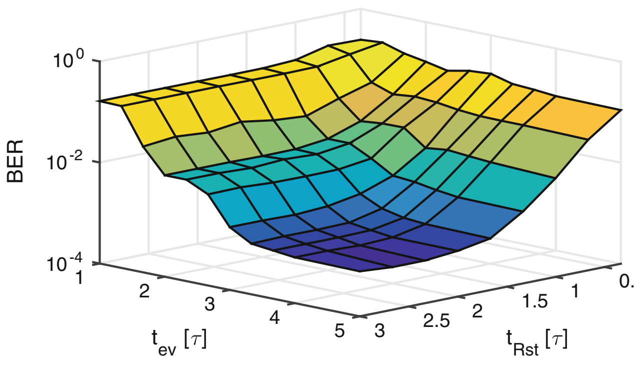


Figure 12 $\boldsymbol{R}_h$ BER surface for different evaluation and reset times, and constant signal-to-noise ratio. Flat performance indicates that (i) the circuit reaches a proper stable equilibrium point, and (ii) does not possess memory of a previous equalization. Note: circuit-based simulation performed in Keysight ADS.

the BER performance becomes flat, values for the minimum equalization ($t_{\mathrm{ev,min}}$) and reset ($t_{\mathrm{Rst,min}}$) time can be found.

$$\begin{cases} \boldsymbol{R}_m : [t_{\mathrm{ev,min}},\ t_{\mathrm{Rst,min}}] = [3.67,\ 1.33]\tau \\ \boldsymbol{R}_h : [t_{\mathrm{ev,min}},\ t_{\mathrm{Rst,min}}] = [4,\ 2]\tau \end{cases}$$

$t_{\mathrm{equ}} = t_{\mathrm{ev,min}} + t_{\mathrm{Rst,min}}$ is the total equalization time, i.e. the minimum relative time between two successive symbol vectors. t_{equ} must be equal or smaller than the symbol interval T_s of the digital transmission. With the numbers from before and $\tau = 42$ ps (see Section 9) we get T_s for the worst case channel $\boldsymbol{R}_h$:

$$T_s \geq (4 + 2) \cdot 42 \text{ ps} = 252 \text{ ps},$$

corresponding to a throughput of four GSymbol/s (16 Gbit/s). For the BER simulations of Fig. 11, the minimum values for T_s were taken.

7 Weights Discretization

For both a real-valued (cf. Section 4.1) and a complex-valued (cf. Section 5.2) vector equalizer the differential transconductance stages f provide the multiplication of a differential current signal, as a function of an analog voltage. All the weights for the equalizer can be in principle configured to assume any value in a range between $[-1, +1]$ with any precision (see also Section 9). As shown in Fig. 13 (a), this section is concerned with the resolution D of the weights, such that the BER of the equalizer with finely spaced discrete weights approaches the BER of an equalizer, driven by precise analog values. Results of this study are presented in Fig. 13 (b) for a QPSK modulation with the complex-valued channel matrix in Eq. 20.

$$\boldsymbol{R}_{cx} = \begin{bmatrix} 1 & \text{0.25-j0.10} & \text{-0.15+j0.15} & \text{+0.15+j0.20} \\ R_{12}^* & 1 & \text{-0.10-j0.35} & \text{+0.10+j0.15} \\ R_{13}^* & R_{23}^* & 1 & \text{-0.35+j0.00} \\ R_{14}^* & R_{24}^* & R_{34}^* & 1 \end{bmatrix} \quad (20)$$

With $D = 1$ bit the equalizer does not work correctly, and the BER presents an error floor. With a resolution $D = 2$ bits the vector equalizer shows a SNR loss of ≈ 3 dB (with respect to an equalizer, driven by precise values) at a BER $= 10^{-2}$. The SNR loss decreases to approximately 0.6 and 0.3 dB, with resolutions $D = 3$ and 4 bits, respectively. The SNR loss falls to a value of ≈ 0.01 dB for $D = 6$ bits, and a similar behavior is observed for different matrices. We conclude that a digital-to-analog converter (DAC), covering the whole range of weights $[-1, +1]$ with resolution $D = 6$ bits, is sufficient to mimic the performance of an equalizer without discretization error.

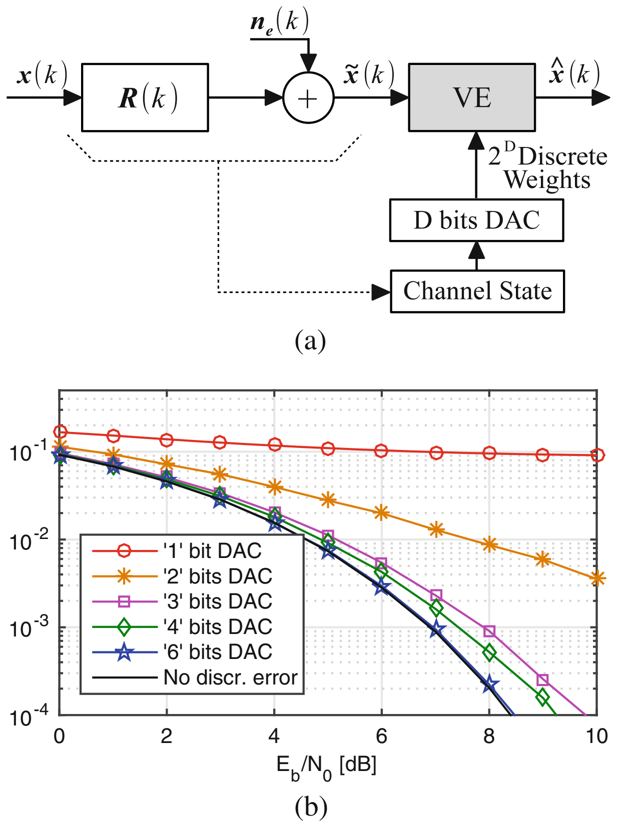


Figure 13 Weights discretization error: (a) Discrete-time model on symbol basis with discrete weights control; (b) BER vs. E_b/N_0 for crosstalk $\boldsymbol{R}_{cx}$, parameterized for different weights resolutions. Setup: 2^{23} input bits, $t_{\mathrm{ev}} = t_{\mathrm{Rst}} = 5\,\tau$.

8 Energy Requirement

Digital and analog signal processing rely on highly diverse theories of operation. A common denominator between the two domains can be found in the energy requirement (ER), here defined as the ratio between the power requirements of an architecture [Watt] and its bit rate, i.e. the number of bits per second the architecture is able to equalize. All other aspects, e.g. the area requirement, are excluded from the comparison. ER has dimensions of [J/bit], so the the smaller ER the more energy efficient the system is. This definition of energy requirement allows to compare very diverse architectures, overcoming the problem of an analysis solely relying on performance, i.e. a pure benchmark.

$$\mathrm{ER}\ [\mathrm{J/bit}] = \frac{\mathrm{Power}\ [\mathrm{W}]}{\mathrm{Bit\ rate}\ [\mathrm{bit/s}]} \tag{21}$$

Digital solutions included in this comparison are ranked in terms of floating point operations per second (FLOPS) and of the related power consumption. The conversion between FLOPS and bit rate is achieved by considering (i) the algorithm complexity (how many floating point operations are required by the algorithm per vector equalization), and (ii) the degree of parallelization (how many bits are produced in parallel after an equalization). The algorithm complexity is computed as follows (cf. Table 2): for a real-valued equalization, each neuron requires three multiplications, four sums, and one hyperbolic tangent computation per iteration. We assume that each operation corresponds to one FLOP, and that ten iterations are sufficient to equalize a vector. This results in an algorithmic complexity of 320 floating point operations per equalization. The output parallelization is equal to N.

Summarizing, the energy requirement for the digital solution ($\mathrm{ER_{dig}}$) can be written as:

$$\mathrm{ER_{dig}} = \frac{\mathrm{Power}}{\mathrm{Outputs} \cdot (\mathrm{FLOPS/Algorithm})} \tag{22}$$

The analog solution represents an application specific integrated circuit, i.e. the algorithm complexity is hardwired in the circuit design. The bit rate is computed from the equalization time t_{equ}, function of the time constant τ. With regards to the power consumption, in our design static power is the dominant parameter involved. The energy requirement for the analog implementation can then be written as:

$$\mathrm{ER_{an}} = \frac{\mathrm{Power}}{\mathrm{Outputs} \cdot (1/\mathrm{Equalization\ time})} \tag{23}$$

Figure 14 shows the energy requirement comparison for the case of a $N = 4$ neurons real-valued equalizer. Cyan squares represent the ten fastest architectures, as ranked in the Top500 list in June 2015 [21]. The rate of execution lies between 50 and 400 Tbit/s, but the power requested to achieve such performance is between 1 and 20 MW. Those architectures show on average an $\mathrm{ER_{dig}} \approx 50$ nJ/bit. Green triangles are representative of the ten most efficient architectures, as ranked in the Green500 list in June 2015 [2]. Those system can perform an equalization with bit rates approximately ranging from 2 to 10 Tbit/s, with power consumptions in the range of 30-200 kW. The average energy requirement is $\mathrm{ER_{dig}} \approx 18$ nJ/bit. Located at the bottom left corner of the picture, red circles show the performance of five commercial general purpose processors[1]. Those single-processor architectures cover bit rates between 300 Mbit/s

Table 2 Algorithm complexity.

	Equalization complexity per neuron	
	Real valued	Complex valued
Mult	$N-1$	$4 \cdot N - 4$
Sums	N	$4 \cdot N - 2$
$\tanh(\cdot)$	1	2

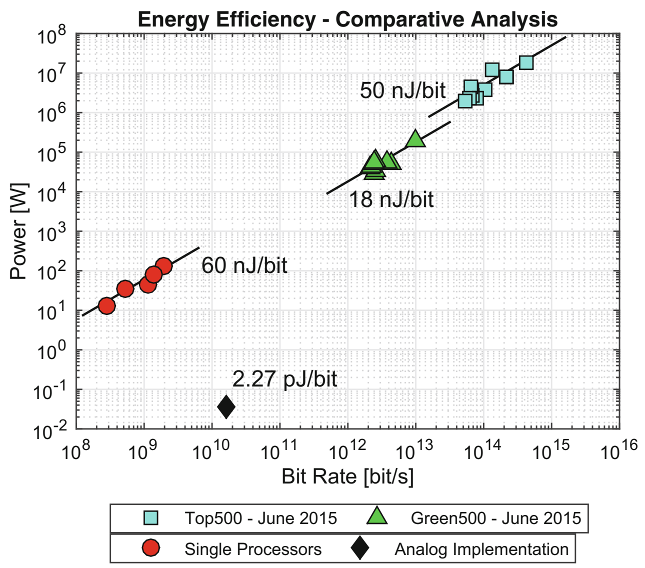


Figure 14 Energy requirement comparison between digital and analog real-valued vector equalizers, with $N = 4$ neurons.

and 2 Gbit/s, with power requirements approximately spanning from 10 to 100 W. The average energy requirement is $\text{ER}_{\text{dig}} \approx 60$ nJ/bit.[1]

Finally, the analog solution presented in this work allows for the equalization of 16 Gbit/s, already outperforming single-processors architectures in terms of pure performance. Most important, the analog vector equalizer requires a static power of only 35 mW (measured on the real chip, cf. Fig. 18 in Section 9). With an energy requirement $\text{ER}_{\text{an}} \approx$ 2 pJ/bit, we can conclude that our dedicated hardware shows an efficiency improvement between three and four orders of magnitude over the digital counterparts.

The advantage of the analog solution is maintained in the case of a complex-valued equalizer. The power consumption of a digital architecture – as well as its performance in FLOPS – is assumed as constant. The output parallelization doubles (from N parallel bits for a real-valued equalization to $2 \cdot N$ bits for a complex-valued one). As derived from Eq. 12 and listed in Table 2, the algorithmic complexity increases to 1120 floating point operations per equalization, for a complex-valued equalization with $N = 4$ neurons.

The complex-valued analog equalizer of Section 5 is designed with a power consumption $P \approx 85$ mW, and a time constant $\tau \approx 160$ ps. Assuming a sufficient equalization time $t_{\text{equ}} = 6\,\tau$, also the complex equalizer shows an energy efficiency improvement of three orders of magnitude.

[1]Microprocessors included in the comparison [Name, declared peak performance, TDP]: (1) Intel i7-3930k, 153 GFLOPS, 130 W; (2) Intel i7-3840QM, 89.6 GFLOPS, 45 W; (3) Intel i5-3570, 108.8 GFLOPS, 77 W; (4) Intel i5-3610ME, 43 GFLOPS, 35 W; (5) Intel i3-3229Y, 22.4 GFLOPS, 13 W.

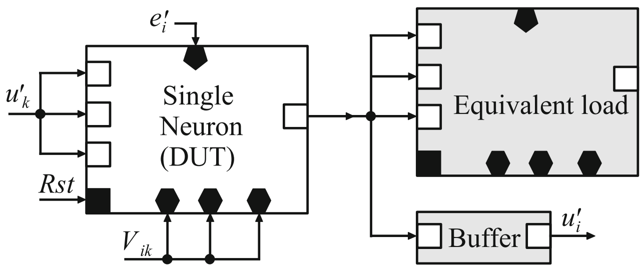


Figure 15 Single neuron characterization: realized test structure, including the single neuron (cf. Fig. 3), an equivalent load and additional buffers to facilitate the measurements. The inner state u_k' is accessible and used to generate the feedbacks for the i^{th} neuron.

9 Measurement Results

Our first measurements focused on the functional validation of the single neuron for real-valued equalizations: weighted multiplication, β of the activation function, and cutoff frequency, i.e. the equivalent time constant τ. For this purpose the circuit of Fig. 15 was realized using a 250 nm SiGe BiCMOS fabrication process by IHP. The test structure was bonded and mounted on a Rogers RO4003 printed circuit board (PCB).

The feedback states u_k' for the i^{th} neuron, coming from the other neurons ($k \in [1, ..., N], k \neq i$), are here externally generated and directly applied. Provided that the neuron under test also drives an identical load ($N - 1$ transconductance stages) as in the full vector equalizer, the characterization of this elementary cell remains valid at system level.

Figure 16 shows the gain variation w_{ik} as a function of the voltage V_{ik} applied. The values are computed by applying a sinusoidal excitation to u_k' and by measuring magnitude and phase of u_i' before the corner frequency given by τ, at a frequency of 0.1 GHz. The attenuator – cf. Figs. 4 and 5 – allows the weights to be fine-tuned, within a span of 1.2 V. The measured curve presents a shift

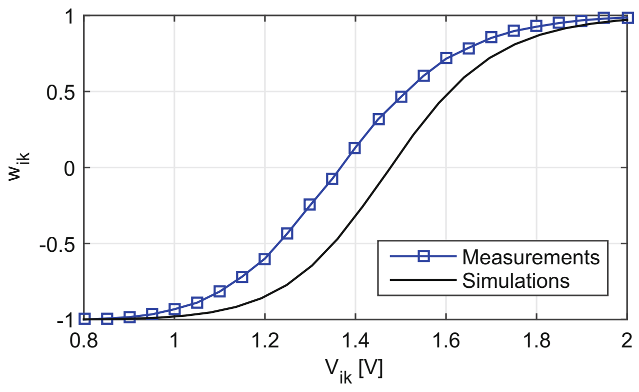


Figure 16 Voltage mapping $w_{ik} = f[V_{ik}] \in [-1, 1]$. Blue squares represent measured values.

$\Delta V_{ik} \approx 0.1$ V with respect to simulations. This shift can however be easily calibrated in the measurement setup and has not any impact on neuron's performance.

The slope of the activation function β at the origin – cf. Eqs. 2 and 5 – is a free parameter that can be optimized. From our simulations, the condition to fulfill for best performance is $\beta \geq 3$ V^{-1}. Measurements performed at 0.1 GHz resulted in a value of 3.47, slightly smaller than the simulated one $\beta = 3.87$ V^{-1}. Reasons can probably be imputed to small losses in the measurement setup.

The equivalent τ for time scaling can be measured by applying a sinusoidal excitation to the external input e'_i and measuring the frequency response at the neuron output u'_i. Fig. 17 (a) shows the simulated transfer function $|u'_i/e'_i|$ and a comparison with an ideal RC low pass filter with cutoff frequency of 3.79 GHz ($\tau = 42$ ps). The hypothesis of a frequency response which resembles an ideal RC behavior is confirmed by Fig. 17 (b), showing the single-input single-output $|u^+_i/e^+_i|$ measurement and the comparison with the expected curve.

Having the single neuron validated by measurement data, a full vector equalizer has been fabricated (Fig. 18). The chip area of 0.68 mm^2 is dominated by the several pads needed for measurements. The pin configuration is the following: four differential external inputs (pads 1, 2, 3, 4, 5, 6, 7, 8), four differential outputs (pads 9, 10, 11, 12, 23, 24, 25, 26), six pins for the weights configuration (pads 13, 14, 18, 19, 20, 21), reset (pad 15), voltage supplies (pads 16, 17, 22) and grounds (square pads). The active area is approximately 0.09 mm^2, with a transistor count CNT = 171 for four neurons. The power consumption of 35 mW was measured, confirming simulation results.

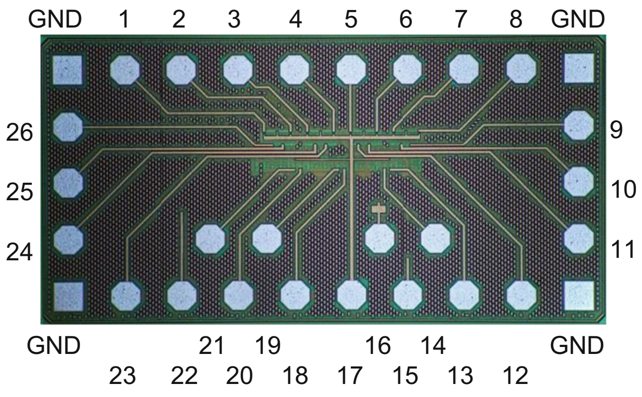
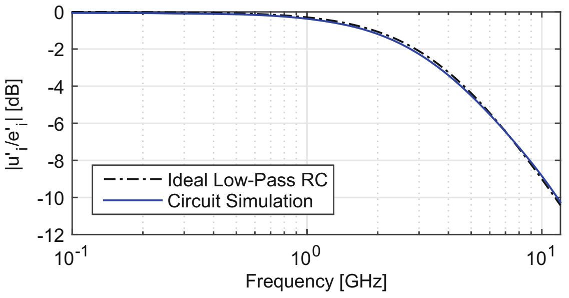


Figure 18 Layout of the vector equalizer and pin configuration.

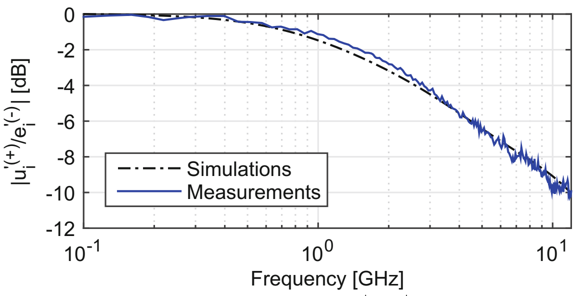
A descriptive test to check the functionality of the equalizer connections is shown in Fig. 19. The equalizer is tested with a set of input vectors $\boldsymbol{e}'$ with equal elements ($e'_i = e', i \in [1, ..., N]$) and the steady state outputs $\boldsymbol{u}'$ are measured. If the weights are set equally for all the neruons ($w_{ik} = w, k \in [1, ..., N], k \neq i$), the expected transfer characteristic $\boldsymbol{u}' = f(\boldsymbol{e}')$ complies with the following transcendental scalar equation:

$$u' - w \cdot (S\alpha) \cdot (N-1) \cdot \tanh\left(\frac{u'}{2 \cdot V_t}\right) = e' \qquad (24)$$

The numerical solution of Eq. 24 with $w = 1$ shows a hysteresis loop, described by the model in Eq. 25: b_{neg} and b_{pos} are two switching boundaries. When the input is outside the boundaries, one unique solution exists for Eq. 24. When the input is within the boundaries, the output presents two stable solutions. The choice of the "plus" or "minus" sign in Eq. 25 then depends on the last crossed boundary. If the



(a) $|u'_i/e'_i|$ frequency response simulation, and comparison with an ideal low pass RC filter with $\tau = 42$ ps.



(b) Single input, single output $|u^+_i/e^+_i|$ frequency response measurement, and comparison with simulation results.

Figure 17 Equivalent τ of a single neuron. Measurements confirm the hypothesis of a first-order low-pass filter comparable to an ideal low-pass RC filter, lumped between u^+_i and u^-_i.

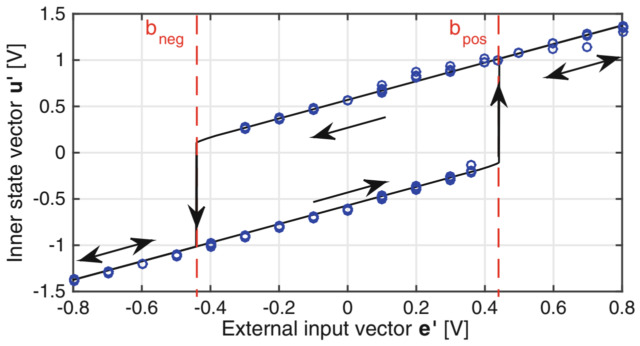


Figure 19 Hysteresis curve resulting from a numerical solution of Eq. 24: comparison between simulations (black line) and measured data (blue circles).

input last crossed b_{pos}, the numerical solution is with the plus sign. Otherwise, the solution with the "minus" sign is the correct one.

$$u' = \begin{cases} e' + (S\alpha)(N-1), \ \forall\, e' \geq b_{\text{pos}} \\ e' - (S\alpha)(N-1), \ \forall\, e' \leq b_{\text{neg}} \\ e' \pm (S\alpha)(N-1), \ \forall\, b_{\text{neg}} < e' < b_{\text{pos}} \end{cases} \tag{25}$$

In other words, because of the neurons' strong nonlinearities, as the external input increases and reaches the boundary b_{pos}, the internal state flips from a negative to a positive value. As the external input decreases and reaches the boundary b_{neg}, the inner state switches from a positive to a negative value. The good agreement between simulations and measurements is confirmed by Fig. 19, where all the four differential outputs are measured for each differential external input vector $\boldsymbol{e}'$.

10 Conclusions

Given the current trend of wireless and mobile communications, implementing complex algorithms, achieving high data rates, and at the same time minimizing the power consumption of a digital signal processing system is becoming extremely challenging. And the situation is not likely to be reversed in the near future, by scaling the minimum feature size of transistors. Our intention is to turn this challenge into an opportunity to revitalize the topic of analog signal processing, i.e. implementing algorithms with efficient dedicated analog circuits.

As an application of analog nonlinear signal processing we presented a vector equalizer for MIMO transmissions, realized in SiGe BiCMOS technology. The equalizer can handle vectors of length $N = 4$ for either BPSK or QPSK modulation schemes.

Bit error rate performance comparisons showed virtually the same or similar behavior for the common digital signal processing and the analog VLSI circuit version. The reason for the comparable robustness – the input is noisy – is that both types of processing use equilibrium states of nonlinear dynamical systems to get the outputs, rather than simple amplitude levels.

The throughput of the vector equalizer is influenced by the evolution time the analog RNN needs to reach the equilibrium state. This time in turn depends on the equivalent time constant τ. In our circuit design the throughput was maximized by exploiting the low-pass behavior, given by parasitic capacitances of bipolar transistors and MOSFETs. Furthermore, an on-chip switch gives the possibility to reset the internal states of the equalizer – a fundamental prerequisite to handle a sequence of vectors.

The analog vector equalizer does not need an analog to digital conversion of the inputs, but needs to be configured with proper weights, representing the channel state. We showed that the optimum interface requires a DAC with a minimum resolution of six bits.

The set of measured data confirmed the expected characteristics of a single neuron. Also the equalizer was tested with a predefined set of input-output vectors, and always confirmed the simulation results. In comparison with common digital signal processing we conclude that the energy efficiency can be improved by some orders of magnitude. This confirms earlier conjectures, stating a huge potential for nonlinear signal processing with analog circuits.

Acknowledgments Financial support by the German research foundation DFG (Deutsche Forschungsgemeinschaft) is gratefully acknowledged. A note of thanks goes also to IHP GmbH for the Si/SiGe foundry processes needed to realize the circuit.

References

1. Cauwenberghs, G. (1996). An analog VLSI recurrent neural network learning a continuous-time trajectory. *IEEE Transactions on Neural Network*, *2*, 346–361.
2. CompuGreen-LLC: The green500 list (2015). http://www.green500.org [Accessed on December 2015].
3. Degnan, B., Marr, B., & Hasler, J. (2016). Assessing trends in performance per watt for signal processing applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *24*(1), 58–66.
4. Draghici, S. (2000). Neural networks in analog hardware - design and implementation issues. *International Journal of Neural Systems*, *10*(1), 19–42.
5. Engelhart, A. (2003). Vector detection techniques with moderate complexity. Ph.D. thesis, Ulm University Institute of Information Technology.
6. Engelhart, A., Teich, W.G., Lindner, J., Jeney, G., Imre, S., & Pap, L. (2002). A survey of multiuser/multisubchannel detection schemes based on recurrent neural networks. *Wireless Communications and Mobile Computing, Special Issue on Advances in 3G Wireless Networks*, *2*(3), 269–284.
7. Haykin, S. (1994). *Neural networks: A comprehensive foundation*. USA: Macmillan college publishing company, Inc.
8. Kechriotis, G.I., & Manolakos, E.S. (1996). Hopfield neural network implementation for optimum CDMA multiuser detector. *IEEE Transactions on Neural Networks*, *7*(1), 131–141.
9. Kothapalli, G. (2005). An analogue recurrent neural network for trajectory learning and other industrial applications. In *3Rd IEEE international conference on industrial informatics (INDIN)* (pp. 462-466). Western Australia: Pert.
10. Kuroe, Y., Hashimoto, N., & Mori, T. (2002). On energy function for complex-valued neural networks and its applications. In *Proc. of the 9th international conference on neural information processing ICONIP'02*, (Vol. 3 pp. 1079–1083).
11. Lindner, J. (1999). MC-CDMA in the context of general multiuser/ multisubchannel transmission methods. *European Transactions on Telecommunications*, *10*(4), 351–367.
12. Loeliger, H.A. (1999). Decoding in analog VLSI. *IEEE Communications Magazine*, *37*(4), 99–101.

13. Markram, H. (2012). The human brain project - a report to the european commission. Tech. rep., The HBP-PS Consortium.
14. Marr, B., Degnan, B., Hasler, P., & Anderson, D. (2013). Scaling energy per operation via an asynchronous pipeline. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *21*(1), 147–151.
15. Mead, C. (1989). Analog VLSI and neural systems Addison-Wesley.
16. Miyajima, T., Hasegawa, T., & Haneishi, M. (1993). On the multiuser detection using a neural network in code-division multiple-access communications. *IEICE Trans. on Communications E76-B*, 961–968.
17. Mostafa, M. (2014). Equalization and decoding: a continuous-time dynamical approach. Ph.D. thesis, Ulm University Institute of Communications Engineering.
18. Mostafa, M., Teich, W.G., & Lindner, J. (2012). Vector equalization based on continuous-time recurrent neural networks. In *6Th IEEE international conference on signal processing and communication systems* (pp. 1-7). Australia: Gold Coast.
19. Mostafa, M., Teich, W.G., & Lindner, J. (2014). Approximation of activation functions for vector equalization based on recurrent neural networks. In *6Th international symposium on turbo codes and iterative information processing* (pp. 52-56). Germany: Bremen.
20. Parlak, M., Matsuo, M., & Buckwalter, J.F. (2012). Analog signal processing for pulse compression radar in 90-nm CMOS. *IEEE Transactions on Microwave Theory and Techniques*, *60*(12), 3810–3822.
21. Prometheus-GmbH: Top500 list (2015). http://www.top500.org [Accessed on December 2015].
22. Schlottmann, C.R., & Hasler, J. (2014). High-level modeling of analog computational elements for signal processing applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *22*(9), 1945–1953.
23. Teich, W.G., Engelhart, A., Schlecker, W., Gessler, R., & Pfleiderer, H.J. (2000). Towards an efficient hardware implementation of recurrent neural network based multiuser detection. In *IEEE 6Th international symposium on spread spectrum techniques and applications* (pp. 662–665). New Jersey, USA: NJIT.
24. Teich, W.G., & Seidl, M. (1996). Code division multiple access communications: multiuser detection based on a recurrent neural network structure. *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications*, *3*, 979–984.

Giuseppe Oliveri received his Bachelor and Master degrees in Electronics Engineering at the University of Palermo, Italy, in 2007 and 2010, respectively. In January 2011 he joined the Electron Devices and Circuits Group at the University of Ulm, Germany, as a doctoral candidate. His research is mainly focused on signal processing algorithms, realized with analog circuits. His interests extend to high frequency microsystems and monolithic microwave integrated circuits, among others.

Mohamad Mostafa received the BEng and Diploma degrees in Electronics Engineering from Aleppo University, Syria in 2004 and 2006, respectively and PhD degree in Communications Engineering from the University of Ulm, Germany in 2014. Since 2013 he is a senior research assistant at the German Aerospace Center (DLR). His research interests include physical layer design, dynamical systems and artificial neural networks, among others.

Werner G. Teich graduated with a M.Sc. in Physics from Oregon State University, Corvallis, Oregon, in 1984. He received the Dipl.-Phys. and the Dr. rer. nat. degree in Physics from the University of Stuttgart in 1985 and 1989, respectively. In 1991 he joined the Department of Information Technology, Ulm University, Germany. Currently, he is Senior Lecturer in Digital Communications at the Institute of Communications Engineering, Ulm University. His research interests are in the general field of digital communications. Specific areas of interest include the application of iterative methods in wireless communications.

Jürgen Lindner is Professor Emeritus and former head of the Institute of Communication Engineering at Ulm University. He received the Dipl.-Ing. and Dr.-Ing. degrees in Electrical Engineering from RWTH Aachen University in 1972 and 1977, respectively. After some years in industrial research he was appointed Full Professor at Ulm University with a chair in Communications Engineering in 1991. His research interests are in wireless digital communications, including broadcast, mobile indoor and outdoor communications. One special field of his research interest is the application of artificial neural networks for low power signal processing with analog VLSI.

Hermann Schumacher obtained his doctorate in engineering from RWTH Aachen University in 1986. He was a member of technical staff at Bellcore, Red Bank, NJ, USA from 1986 until 1990. Since 1990, he has been a Professor with Ulm University. Since 2010, he is the director of the Institute of Electron Devices and Circuits, and since 2011 director of the University' School of Advanced Professional Studies. His research areas are in monolithic IC design for millimeter-wave applications, and high speed analog signal processing, applied to both communications and sensor systems.