Giuseppe Oliveri | 08/10/2015 | 2015 Wireless Innovation Forum European Conference on Communications Technologies and Software Defined Radio | Erlangen (Germany)

Advanced Low Power, High Speed Nonlinear Signal Processing

An Analog VLSI Example

Outline

- Introduction
  - Analog-Assisted Digital
  - The Equalizer Example
  - Transmission Model
- 4-Neurons Fully Adaptive Vector Equalizer
  - Recurrent Neural Networks
  - Time Domain Equalization
  - BER Surfaces and Curves
- Design
  - Schematic
  - Layout
  - Measurements
- Conclusions
  - Summary
  - Outlook

Introduction

Analog-Assisted Digital

Digital

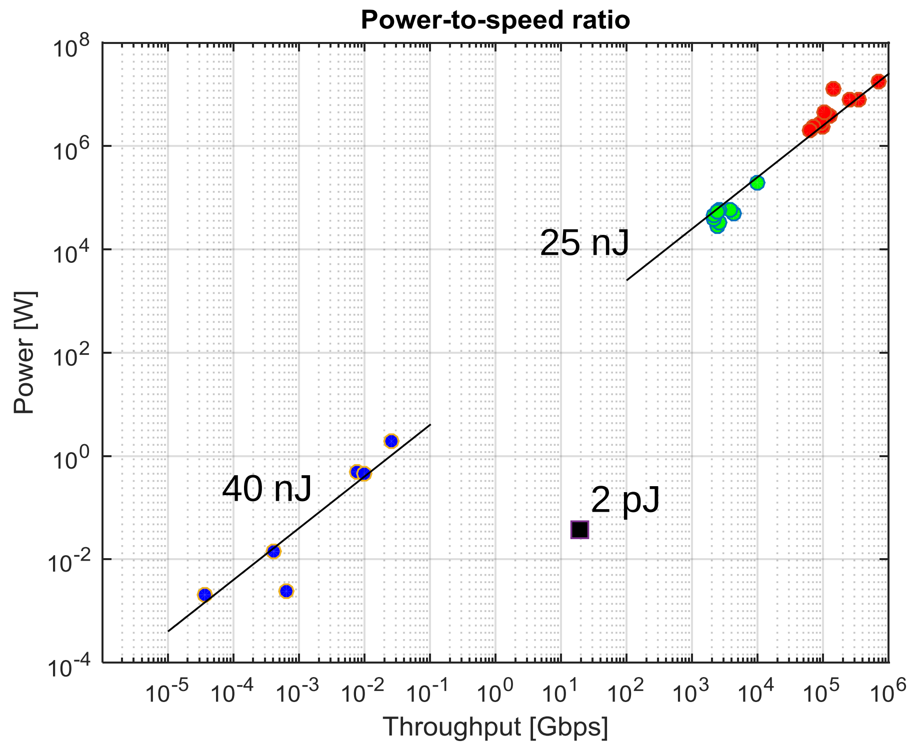
Flexibility;
Fast implementation of complex algorithms.

A/D – D/A needed;
Speed and power bottlenecks.

Analog

No A/D – D/A and no quantization error;
Computational speed;
Energy efficiency.

Effort to introduce adaptivity/scalability;
Noise, linearity, process variation;
Layout challenges.



Digital solutions
- Low-power microprocessors;
- Top-10 Green500 – June 2015;
- Top-10 Top500 – June 2015.

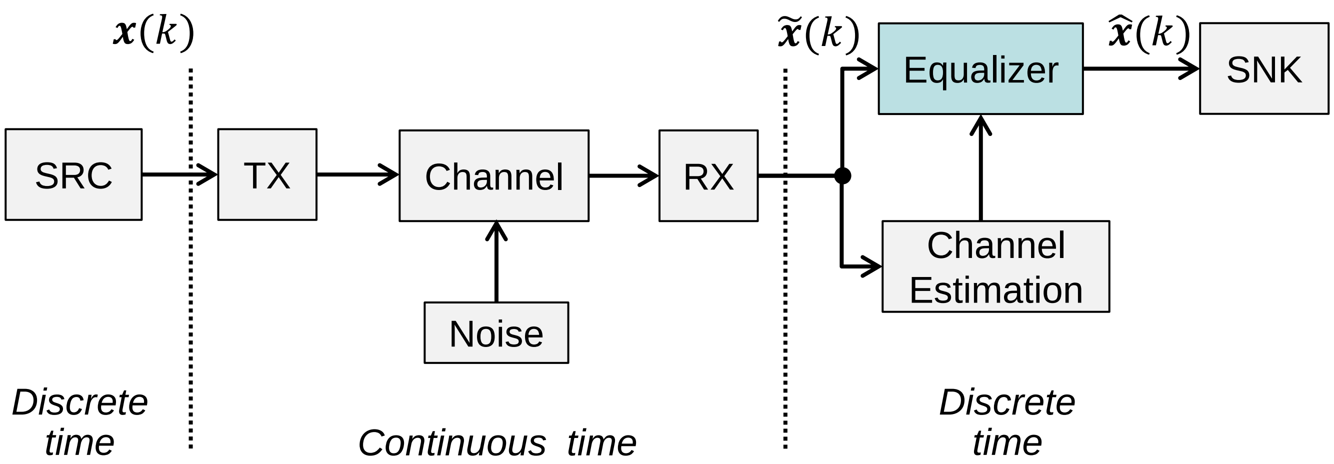
Analog solution
P_{stat} = 35 mW; Throughput = 16 Gbps.

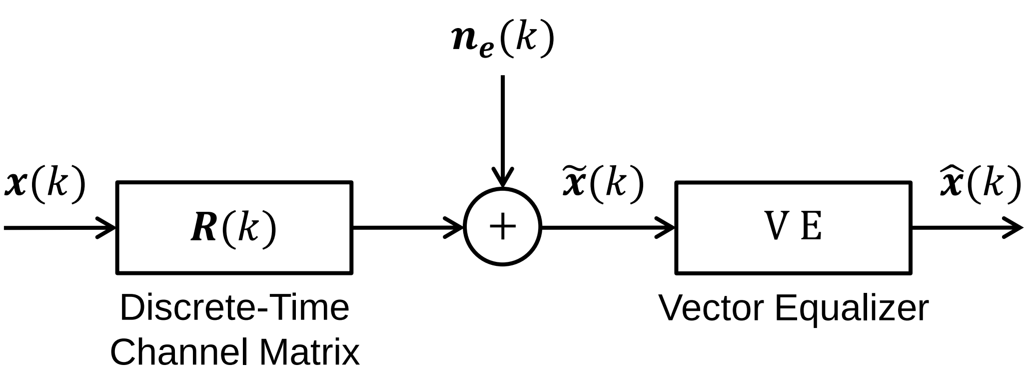
*Throughput calculation for an equalization:
10 Iterations, 8 Operations per iteration, 1 Clock per operation.

The Equalizer Example

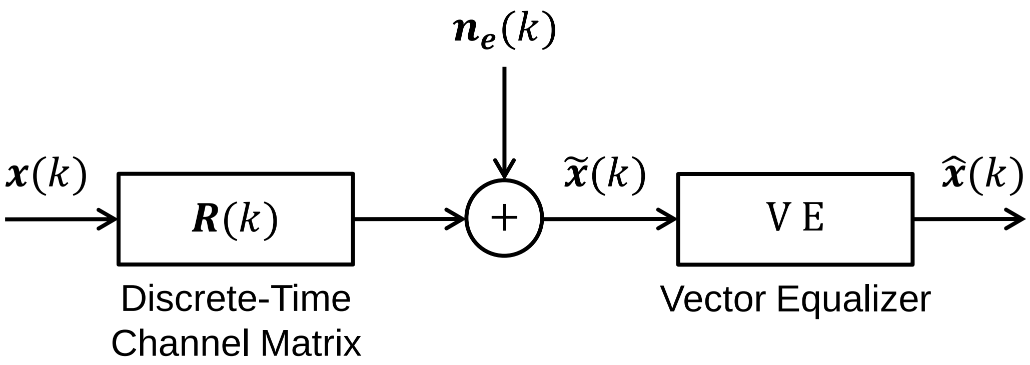
Reliable communication	Equalization	*Improve the performance in frequency selective channels.*
	Vector-valued transmission model	*Analyze all the different types of interference present in the system.*
Low power	Complexity	*Near-optimum, iterative schemes.*
	Analog VLSI technology	*No need for input quantization; Soft-exchange of information.*
High Speed	Recurrent neural networks	*Highly interconnected; Nonlinear, computationally complex.*
	Devices nonlinearities	*Profit from low-pass transfer function.*

Vector-Valued Transmission Model





Vector-Valued Transmission Model



- Valid model for linear modulation schemes as well as multiple access methods.
- Discrete-time channel matrix: whole link path between transmitted and received symbols.
- Communication link described by ***R*** (more details are not needed).

$$\widetilde{\boldsymbol{x}} = \boldsymbol{R} \cdot \boldsymbol{x} + \boldsymbol{n_e},$$

$$\widetilde{\boldsymbol{x}} = \underbrace{\boldsymbol{R}_d \cdot \boldsymbol{x}}_{\text{signal}} + \underbrace{\boldsymbol{R}_{\setminus d} \cdot \boldsymbol{x}}_{\text{interference}} + \underbrace{\boldsymbol{n_e}}_{\text{additive noise}},$$

$$\boldsymbol{R} = \underbrace{\boldsymbol{R}_d}_{\text{diagonal elements}} + \underbrace{\boldsymbol{R}_{\setminus d}}_{\text{non-diagonal elements}}.$$

4-Neurons Fully Adaptive Vector Equalizer

Recurrent Neural Networks

Continuous-time RNN

$$\boldsymbol{\gamma} \cdot \frac{\mathrm{d}\boldsymbol{u}(t)}{\mathrm{d}t} = -\boldsymbol{u}(t) + \boldsymbol{W} \cdot \boldsymbol{v}(t) + \boldsymbol{W_0} \cdot \boldsymbol{e},$$

$$\boldsymbol{v}(t) = \boldsymbol{\varphi}[\boldsymbol{u}(t)],$$

$$\hat{\boldsymbol{v}}(t) = \mathbf{HD}[\boldsymbol{v}(t)].$$

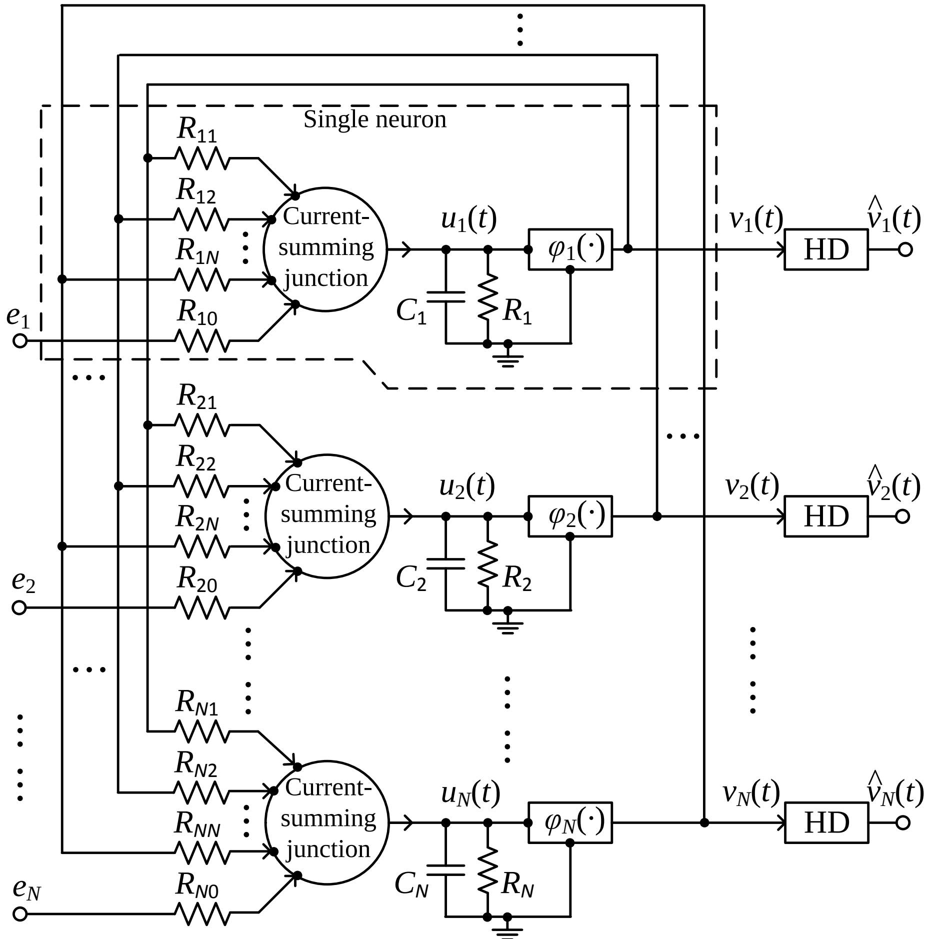
Legend

- t: time variable;
- N: number of neurons (vector lengths);
- $\boldsymbol{\varphi}(\cdot)$: activation function ($\mathbf{tanh}(\cdot)$);

- $\boldsymbol{u}(t)$; inner state vector;
- $\boldsymbol{v}(t)$: output of the activation function;
- $\hat{\boldsymbol{v}}(t)$: hard decision output vector;
- $\boldsymbol{e}(t)$: external input vector;

- $\boldsymbol{\gamma}$: diagonal matrix $(\gamma_{jj} = \tau_j = f[R_j, C_j])$;
- $\boldsymbol{W}$: channel matrix weights' choice;
- $\boldsymbol{W_0}$: external inputs weights' choice.

Resistance-capacitance model



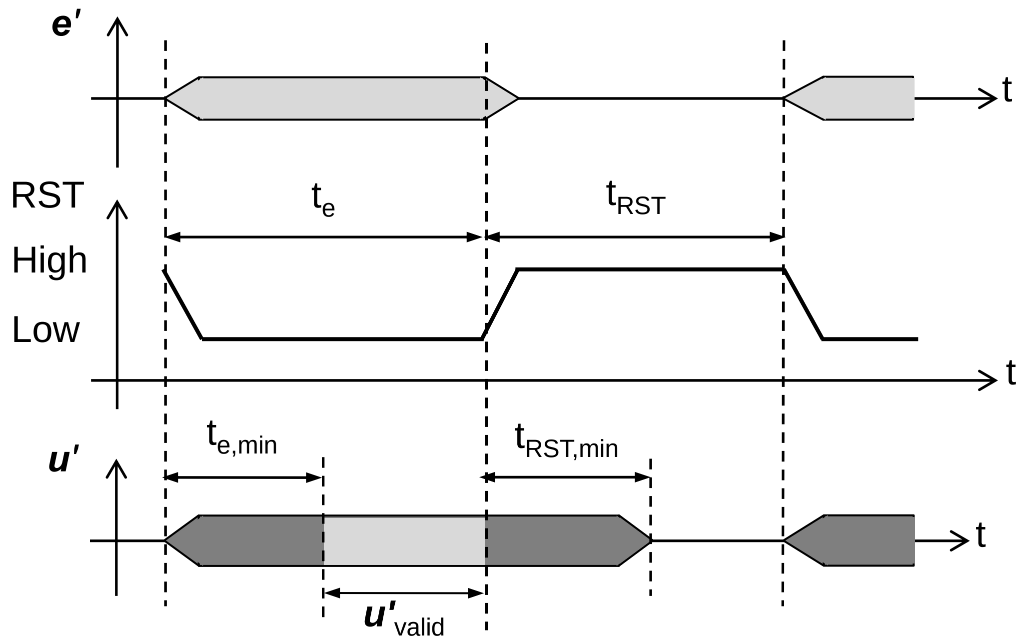
Time Domain Equalization

How the VE reaches stability, given a sequence of input vectors.

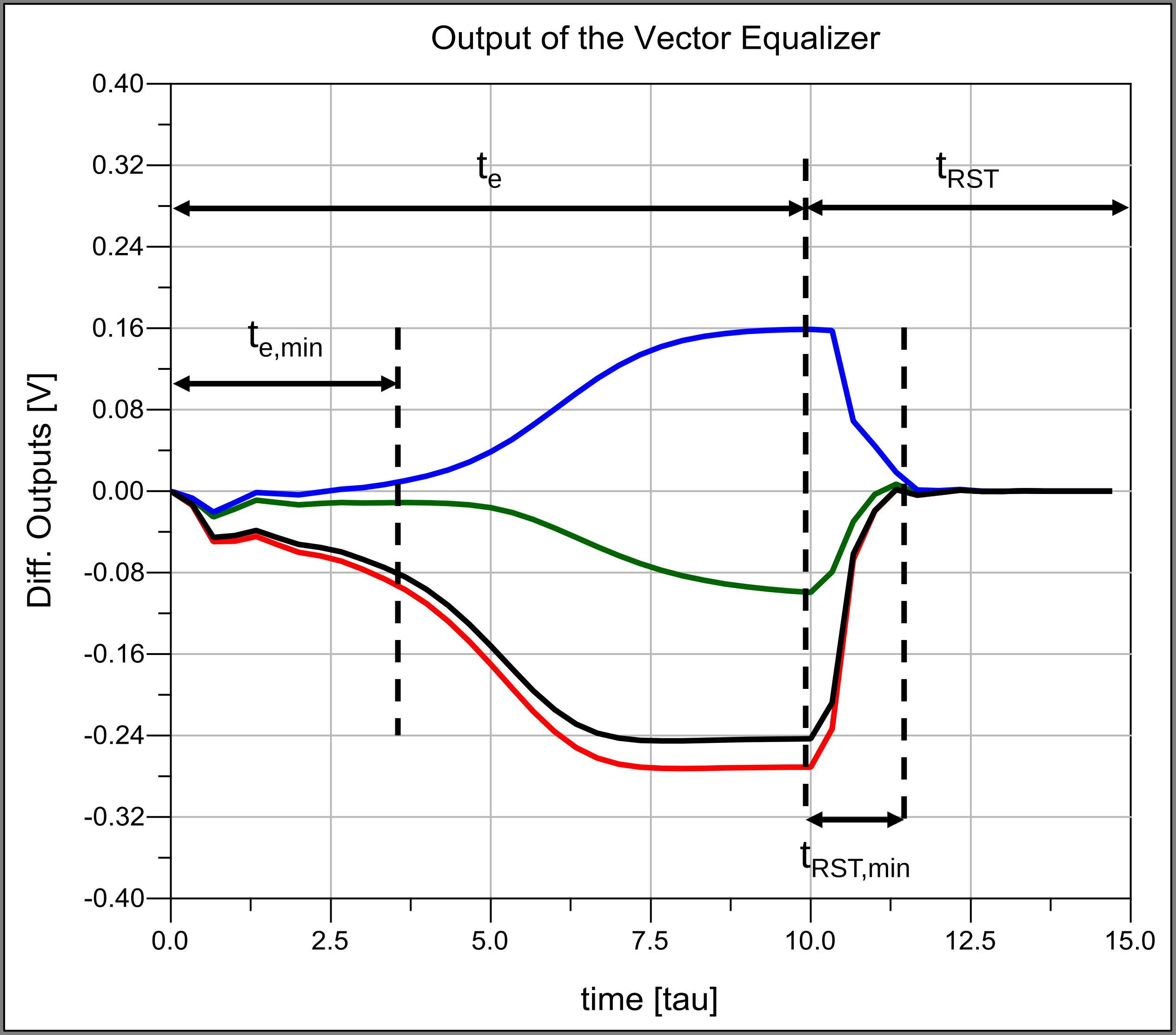
- $\boldsymbol{e}'$: scaled external input vector;
- RST: reset signal (acts as a sequencer);
- $\boldsymbol{u}'$: scaled inner state vector.

- $\mathrm{BER} = \mathrm{f}[\boldsymbol{R}, \mathrm{E_b/N_0}, \mathrm{t_e}, \mathrm{t_{RST}}]$;
- $\mathrm{t_e}$: equalization time slot;
- $\mathrm{t_{RST}}$: reset time slot.

- $[\mathrm{t_{e,min}}, \mathrm{t_{RST,min}}]$: statistical optimization for best performance.

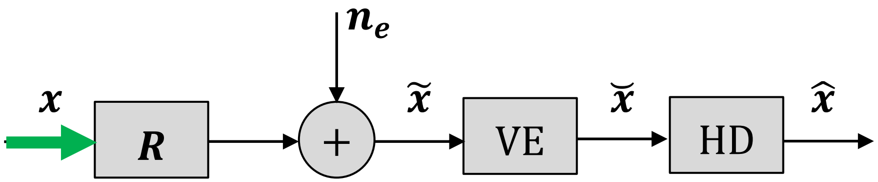


Time Domain Equalization



Time Domain Equalization

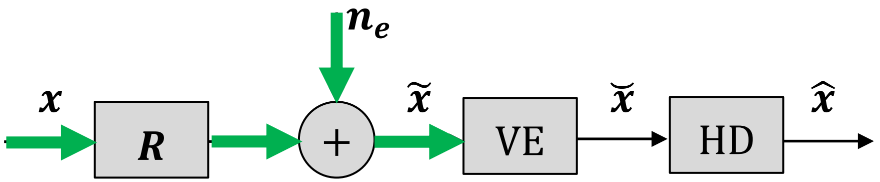
- Pseudo-random sequence (40 bits BPSK);
- $r_{ji} = 0.6, \forall\, i \neq j;$ $E_b/N_0 = 9$ dB;
- $\tau = 42$ ps; $t_e = 10 \cdot \tau;$ $t_{RST} = 5 \cdot \tau.$



-1+1+1-1	+1+1-1-1	-1+1-1+1	-1-1-1+1	+1+1+1+1	+1+1+1+1	-1-1-1+1	-1-1+1-1	-1-1-1+1	-1+1-1-1

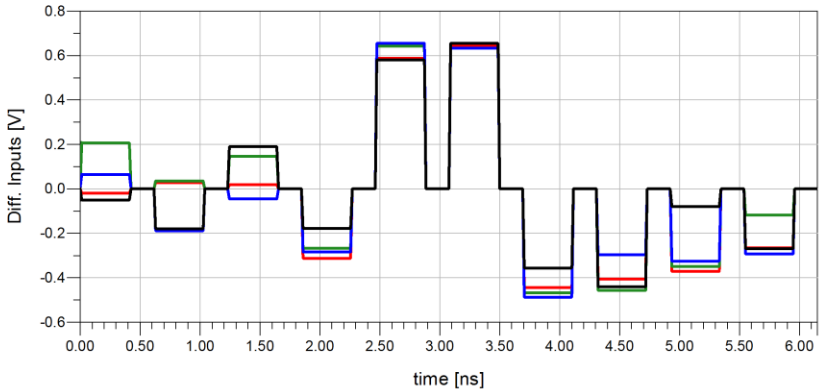
Time Domain Equalization

- Pseudo-random sequence (40 bits BPSK);
- $r_{ji} = 0.6, \forall\, i \neq j;$ $E_b/N_0 = 9$ dB;
- $\tau = 42$ ps; $t_e = 10 \cdot \tau;$ $t_{RST} = 5 \cdot \tau.$



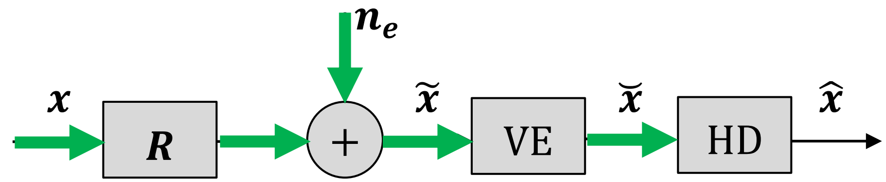
-1+1+1-1	+1+1-1-1	-1+1-1+1	-1-1-1+1	+1+1+1+1	+1+1+1+1	-1-1-1+1	-1-1+1-1	-1-1-1+1	-1+1-1-1

Inputs to the vector equalizer



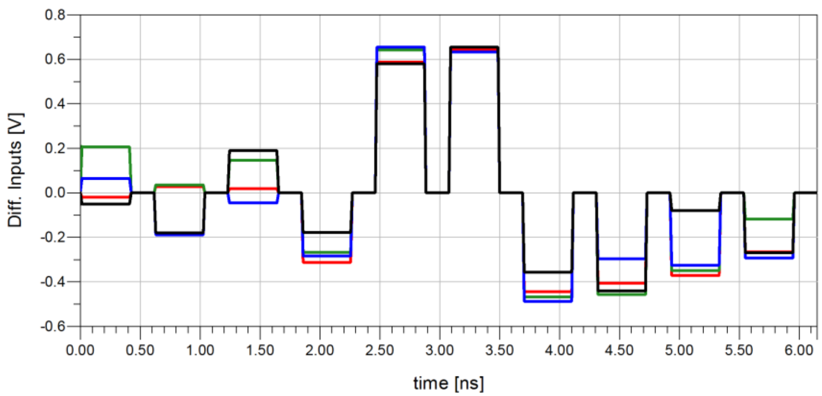
Time Domain Equalization

- Pseudo-random sequence (40 bits BPSK);
- $r_{ji} = 0.6, \forall\, i \neq j;$ $E_b/N_0 = 9$ dB;
- $\tau = 42$ ps; $t_e = 10 \cdot \tau;$ $t_{RST} = 5 \cdot \tau.$

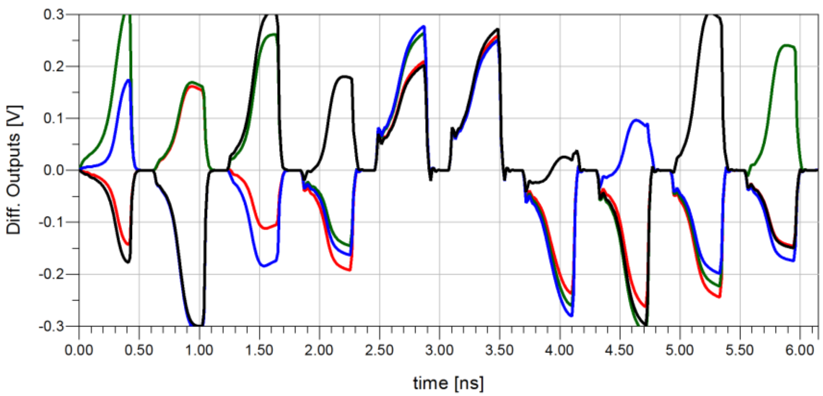


-1+1+1-1	+1+1-1-1	-1+1-1+1	-1-1-1+1	+1+1+1+1	+1+1+1+1	-1-1-1+1	-1-1+1-1	-1-1-1+1	-1+1-1-1

Inputs to the vector equalizer

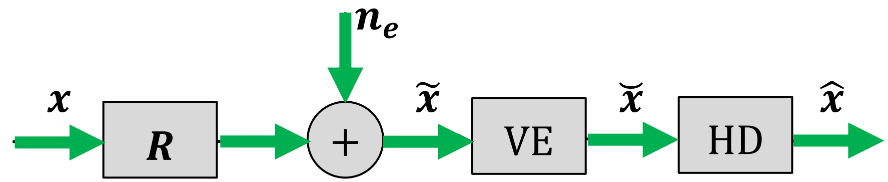


Outputs of the vector equalizer



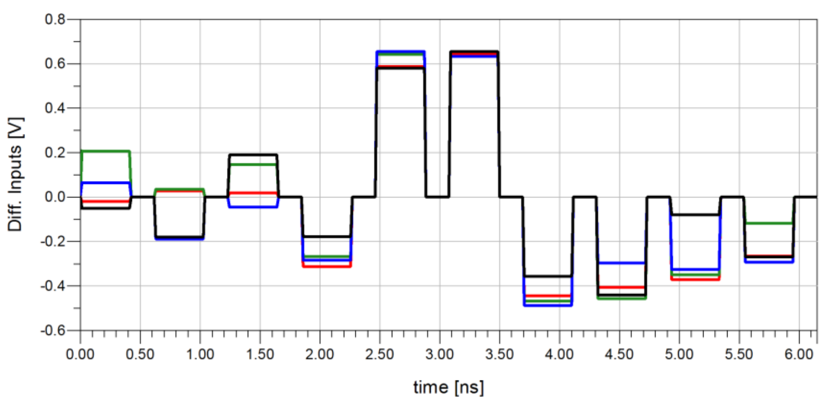
Time Domain Equalization

- Pseudo-random sequence (40 bits BPSK);
- $r_{ji} = 0.6, \forall\, i \neq j$; $E_b/N_0 = 9$ dB;
- $\tau = 42$ ps; $t_e = 10 \cdot \tau$; $t_{RST} = 5 \cdot \tau$.

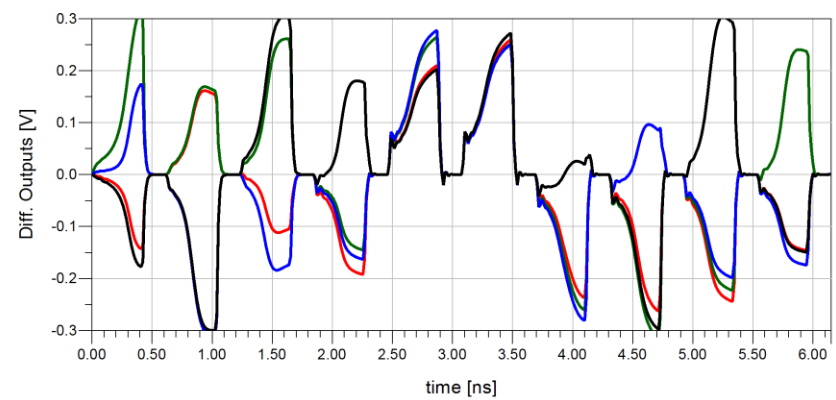


-1+1+1-1	+1+1-1-1	-1+1-1+1	-1-1-1+1	+1+1+1+1	+1+1+1+1	-1-1-1+1	-1-1+1-1	-1-1-1+1	-1+1-1-1

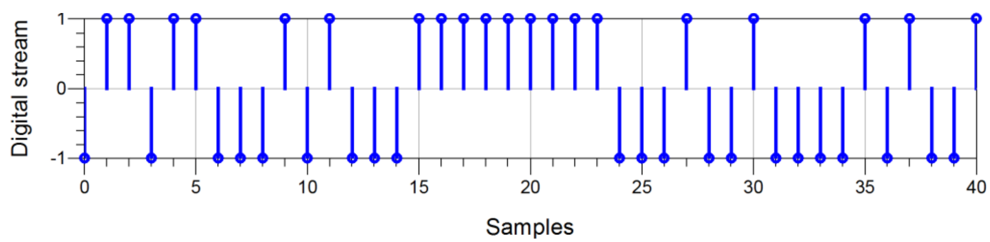
Inputs to the vector equalizer



Outputs of the vector equalizer



Recovered data stream after hard decision

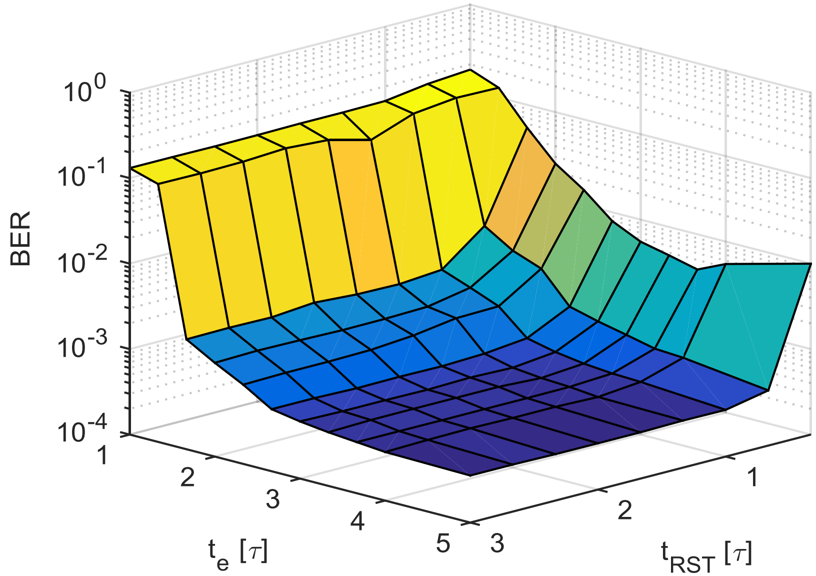


BER Surfaces and Curves

Moderate Interference

$$\underline{\underline{\mathbf{R}}}_{m} = \left\| \begin{matrix} 1 & +0.60 & +0.60 & +0.60 \\ +0.60 & 1 & +0.60 & +0.60 \\ +0.60 & +0.60 & 1 & +0.60 \\ +0.60 & +0.60 & +0.60 & 1 \end{matrix} \right\|$$

$$t_{e,\min} = 3.67\ \tau$$
$$t_{RST,\min} = 1.33\ \tau$$

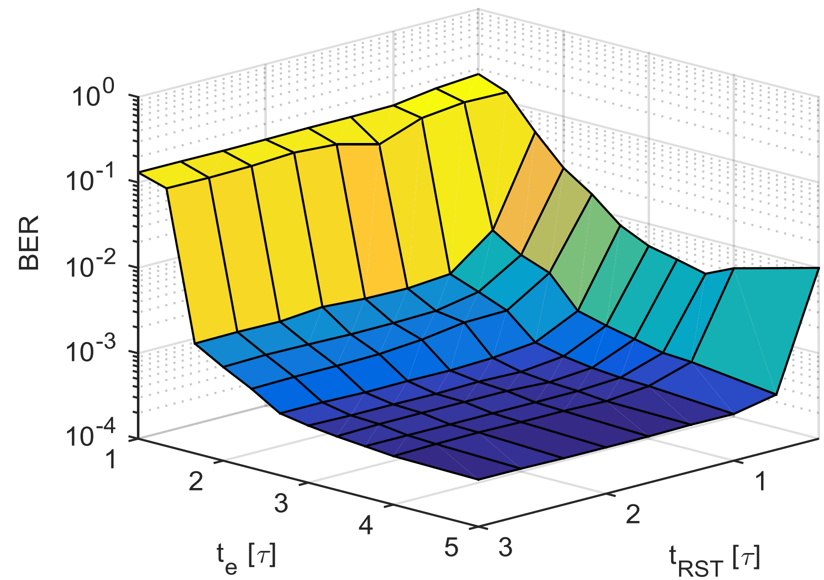


BER Surfaces and Curves

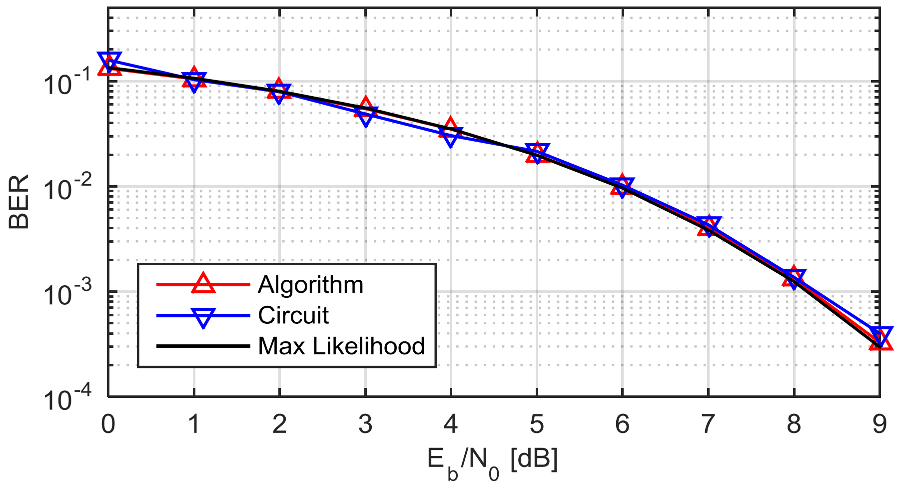
Moderate Interference

$$\underline{\underline{\mathbf{R}}}_m = \begin{Vmatrix} 1 & +0.60 & +0.60 & +0.60 \\ +0.60 & 1 & +0.60 & +0.60 \\ +0.60 & +0.60 & 1 & +0.60 \\ +0.60 & +0.60 & +0.60 & 1 \end{Vmatrix}$$

$$t_{e,min} = 3.67\ \tau$$
$$t_{RST,min} = 1.33\ \tau$$



Throughput: ~19 Gbps

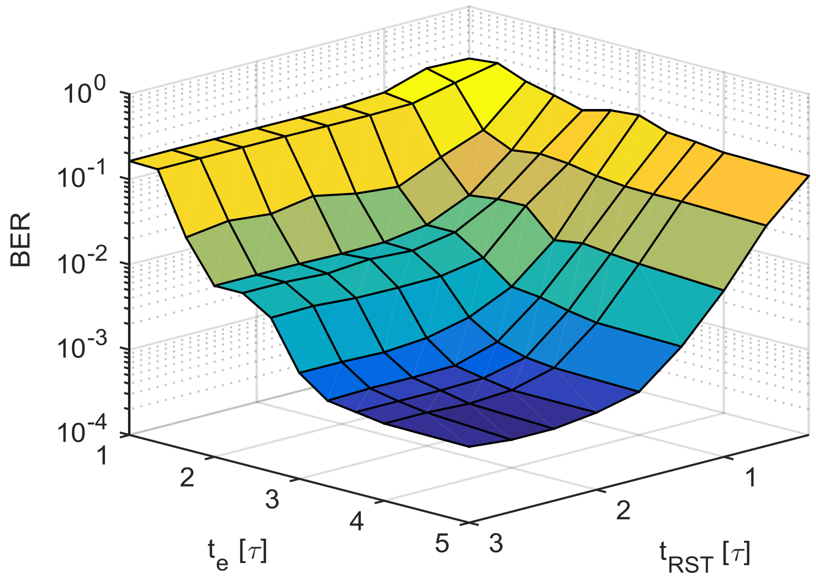


BER Surfaces and Curves

High Interference

$$\underline{\underline{\mathbf{R}}}_{h} = \left\| \begin{matrix} 1 & +0.85 & +0.66 & -0.67 \\ +0.85 & 1 & +0.85 & -0.79 \\ +0.66 & +0.85 & 1 & -0.89 \\ -0.67 & -0.79 & -0.89 & 1 \end{matrix} \right\|$$

$$t_{e,\mathrm{min}} = 4\ \tau$$
$$t_{\mathrm{RST,min}} = 2\ \tau$$

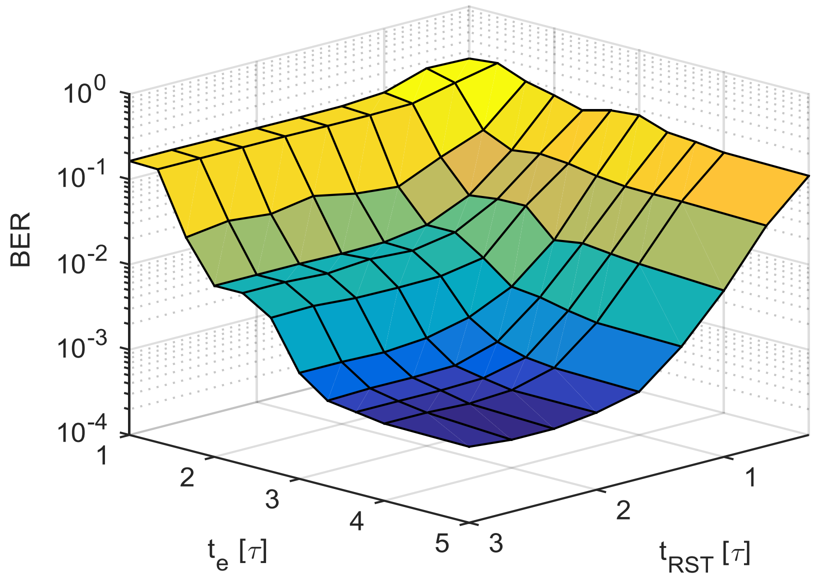


BER Surfaces and Curves

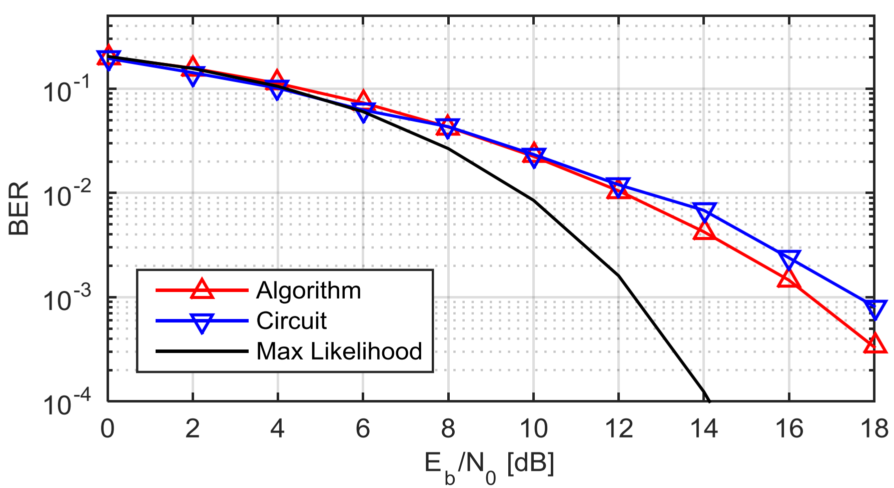
High Interference

$$\underline{\underline{\mathbf{R}}}_{\boldsymbol{h}} = \left\| \begin{matrix} 1 & +0.85 & +0.66 & -0.67 \\ +0.85 & 1 & +0.85 & -0.79 \\ +0.66 & +0.85 & 1 & -0.89 \\ -0.67 & -0.79 & -0.89 & 1 \end{matrix} \right\|$$

$$t_{e,min} = 4\,\tau$$
$$t_{RST,min} = 2\,\tau$$



Throughput: ~16 Gbps



Design

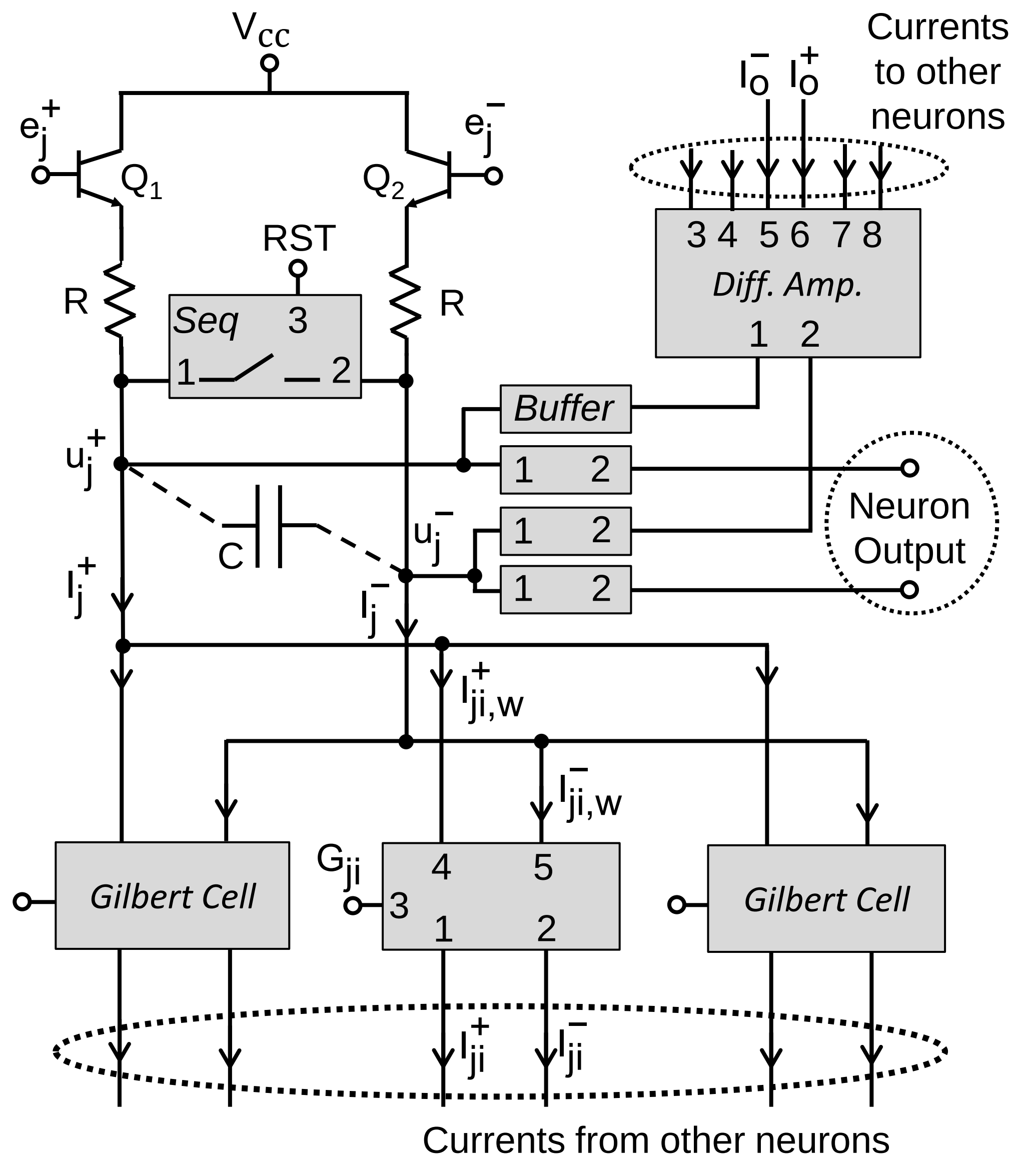
Schematic

Transistor-level hardware implementation:

- 0.25 µm IHP SiGe BiCMOS technology.

Novelty:

- No A/D Converters for external inputs;
- No external capacitors;
- Sequencer: NMOS FETs.



Schematic

Legend:

- u_j' : inner state of the j^{th} neuron;
- e_j' : external input applied to the j^{th} neuron;
- G_{ji} : weight configuration (from output of i^{th} to input of j^{th} neuron).

Fully-differential circuit:

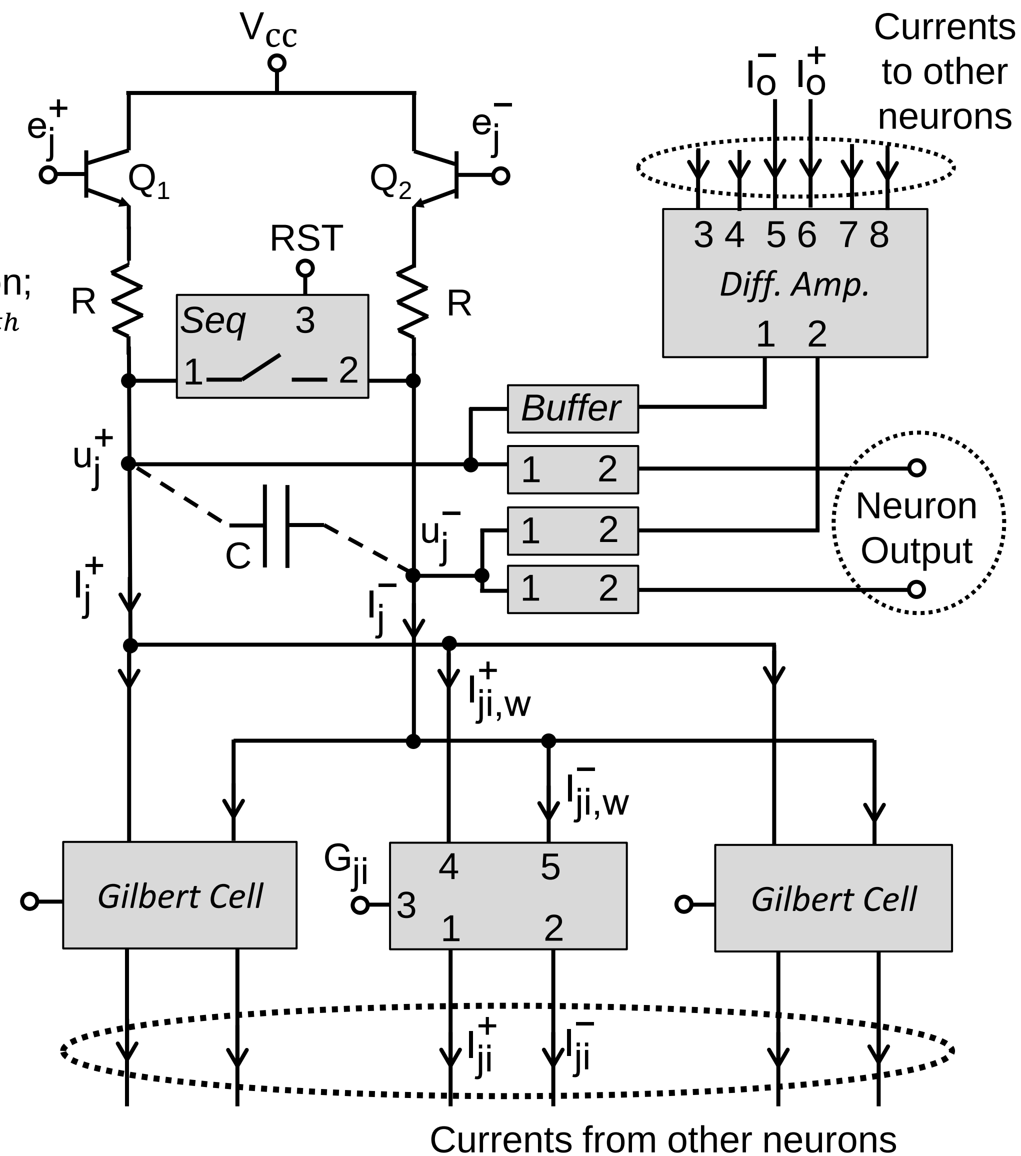
- $I_{ji} = I_{ji}^{+} - I_{ji}^{-}$; $I_{ji,w} = I_{ji,w}^{+} - I_{ji,w}^{-}$;
- $I_j = I_j^{+} - I_j^{-}$; $I_o = I_o^{+} - I_o^{-}$;
- $u_j' = u_j^{+} - u_j^{-}$; $e_j' = e_j^{+} - e_j^{-}$;

Scaling:

- $[\boldsymbol{u}'(t), \boldsymbol{v}'(t), \boldsymbol{e}'(t)] = S \cdot [\boldsymbol{u}(t), \boldsymbol{v}(t), \boldsymbol{e}(t)]$.

$$\boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{u}'(t)}{\mathrm{d}t} = -\boldsymbol{u}'(t) + \boldsymbol{W} \cdot \boldsymbol{v}'(t) + \boldsymbol{W}_0 \cdot \boldsymbol{e}',$$

$$\boldsymbol{v}'(t) = S \cdot \alpha \cdot \tanh\left(\frac{\beta \cdot \boldsymbol{u}'(t)}{S}\right).$$

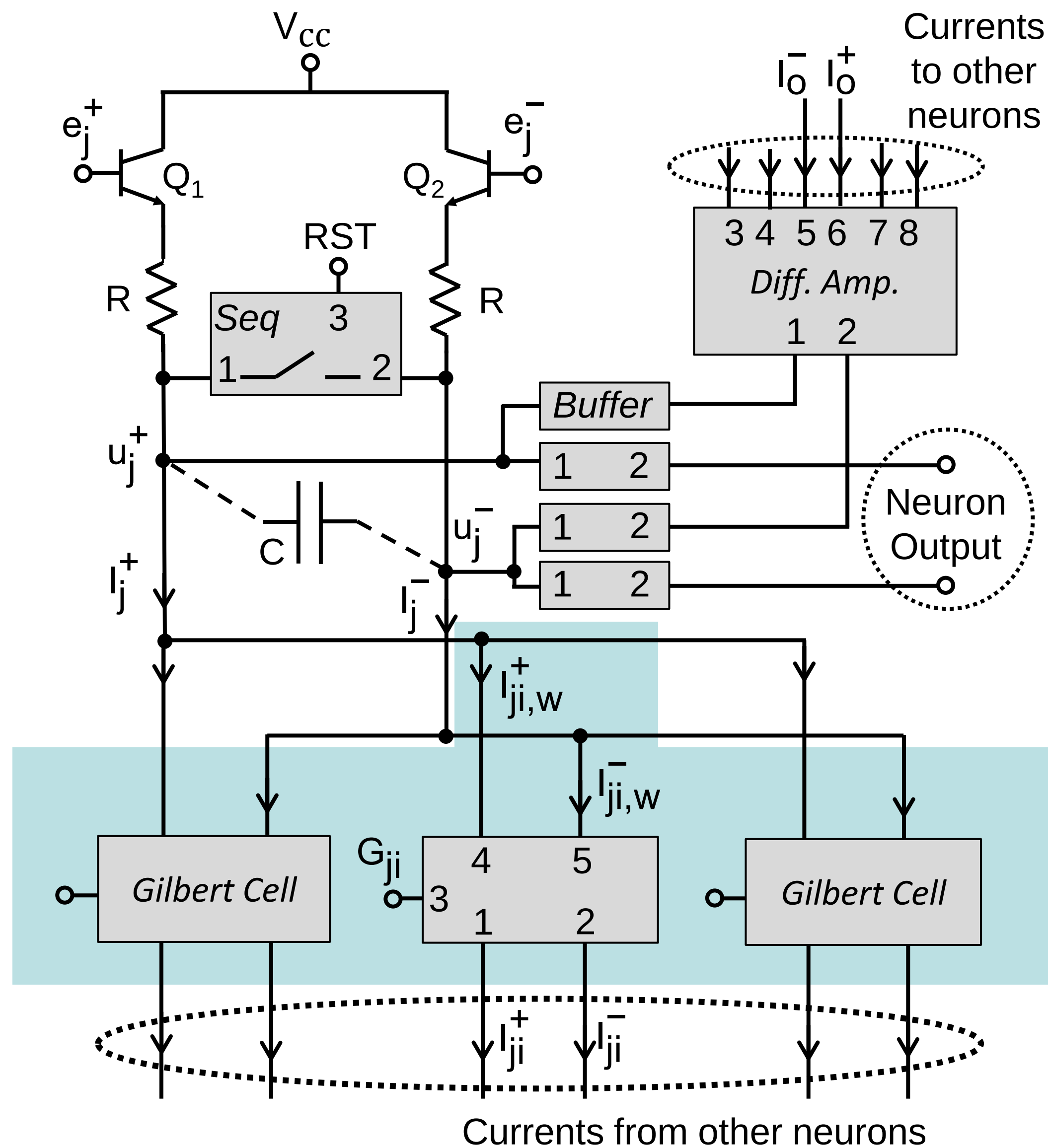


Schematic

Attenuator and Gilbert cell:

$$w_{ji} = f\left[G_{ji}\right] \in [-1, +1];$$

$$I_{ji,w}(t) = w_{ji} \cdot I_{ji}(t).$$



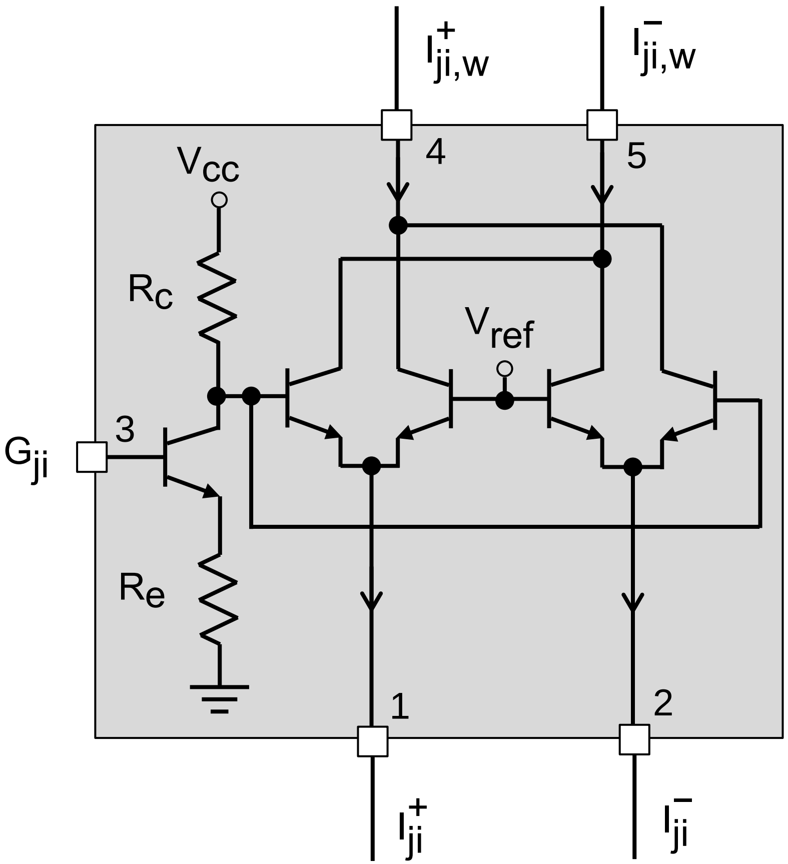
Schematic

Attenuator:

$$w_{ji} = f\left[G_{ji}\right] \in [-1, +1]$$

Gilbert cell:

$$I_{ji,w}(t) = w_{ji} \cdot I_{ji}(t)$$



Schematic

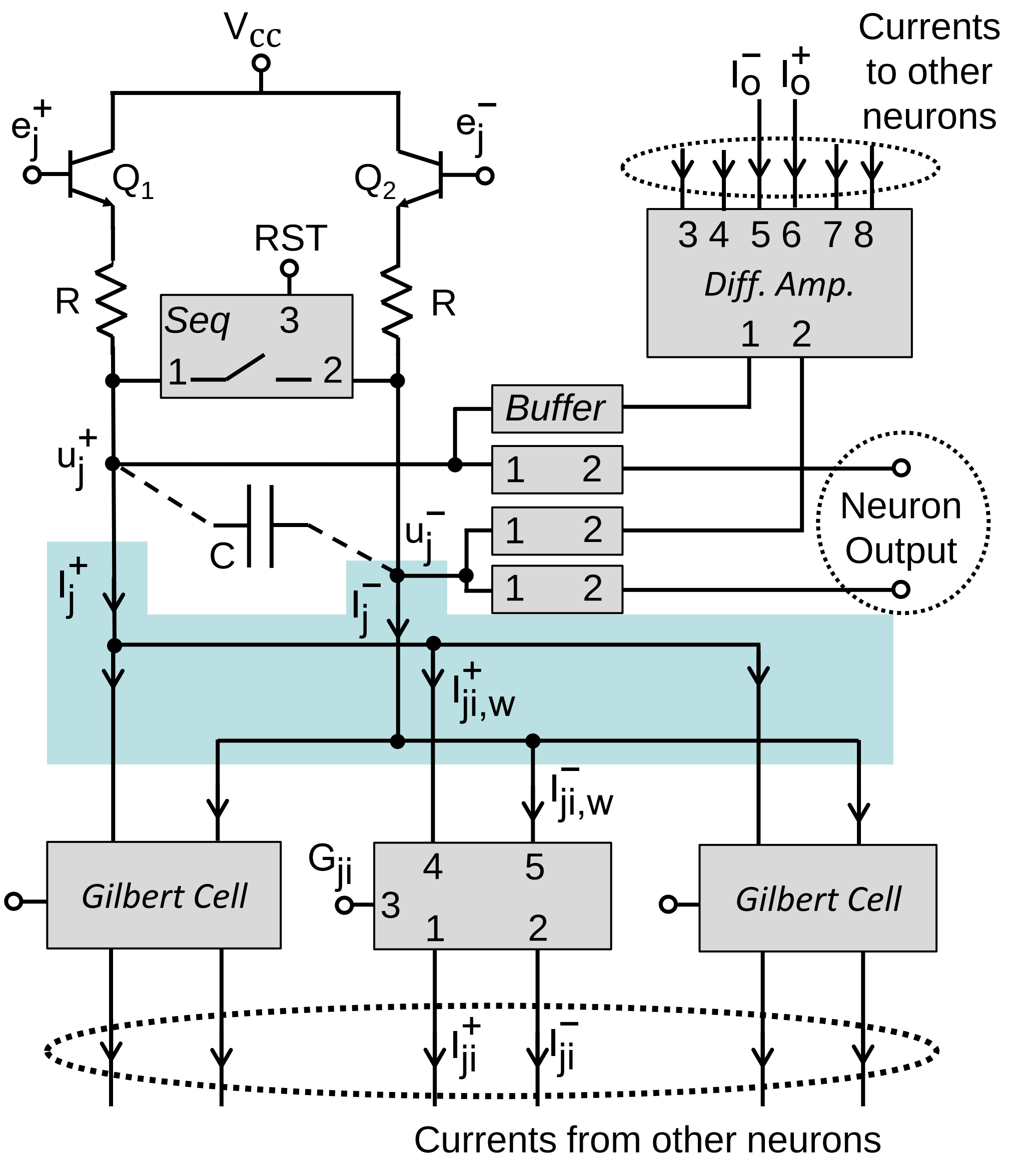
Attenuator and Gilbert cell:

$$w_{ji} = f\left[G_{ji}\right] \in [-1,+1];$$

$$I_{ji,w}(t) = w_{ji} \cdot I_{ji}(t).$$

Summation nodes:

$$I_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^{N} w_{ji} \cdot I_{ji}(t)$$



Schematic

Attenuator and Gilbert cell:

$$w_{ji} = f\left[G_{ji}\right] \in [-1, +1];$$

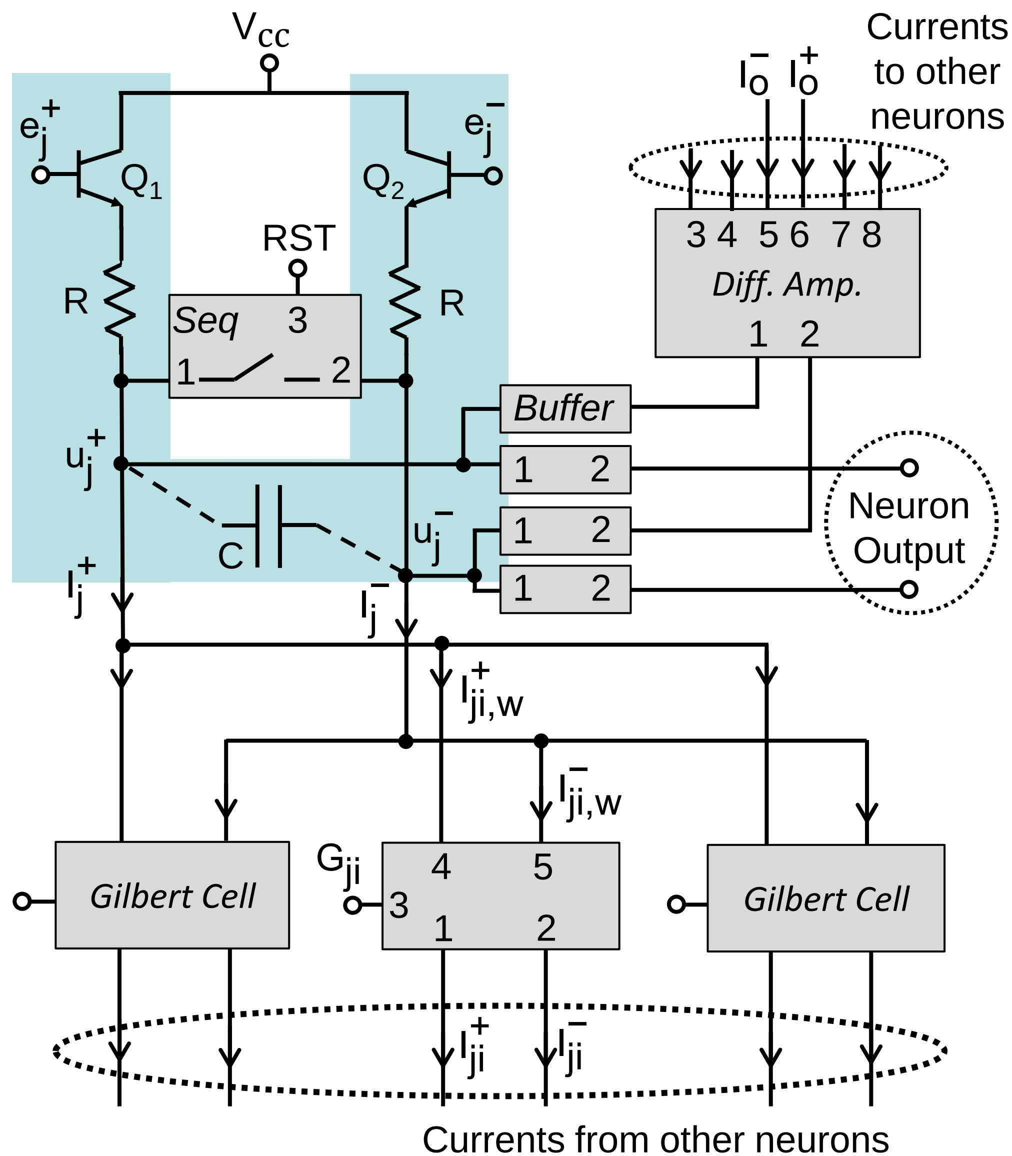
$$I_{ji,w}(t) = w_{ji} \cdot I_{ji}(t).$$

Summation nodes:

$$I_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^{N} w_{ji} \cdot I_{ji}(t)$$

Inner state:

$$\tau \cdot \frac{du_j'(t)}{dt} = -u_j'(t) - R \cdot \sum_{\substack{i=1 \\ i \neq j}}^{N} w_{ji} \cdot I_{ji}(t) + e_j'$$



Schematic

Attenuator and Gilbert cell:

$$w_{ji} = f\left[G_{ji}\right] \in [-1, +1];$$

$$I_{ji,w}(t) = w_{ji} \cdot I_{ji}(t).$$

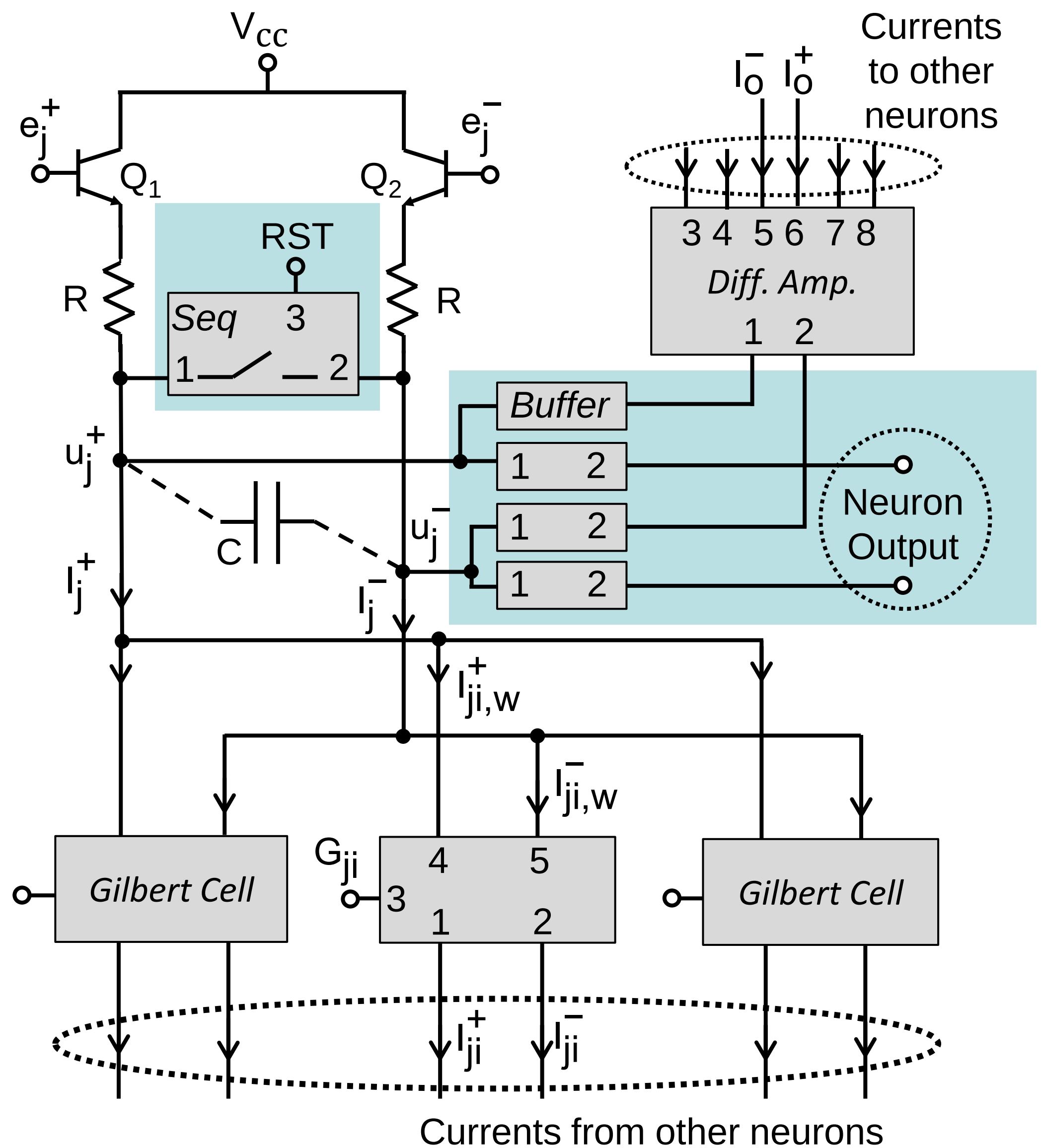
Summation nodes:

$$I_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^{N} w_{ji} \cdot I_{ji}(t)$$

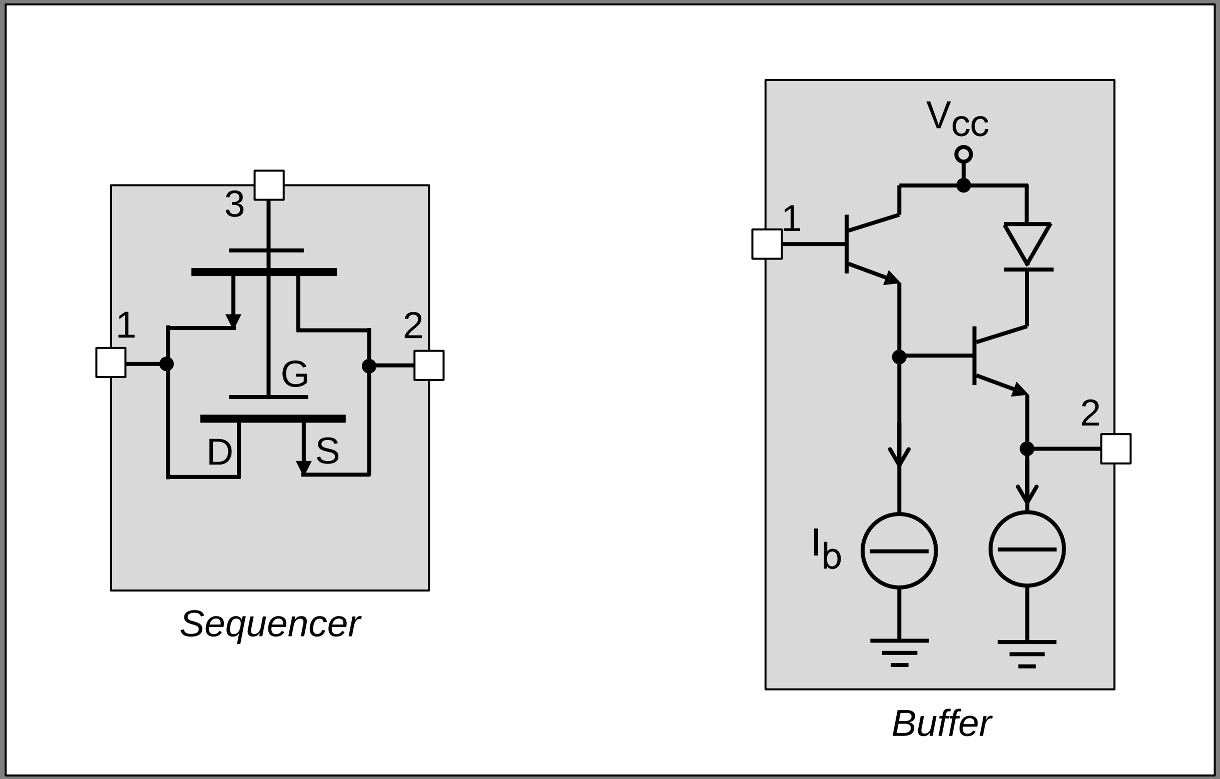
Inner state:

$$\tau \cdot \frac{du_j'(t)}{dt} = -u_j'(t) - R \cdot \sum_{\substack{i=1 \\ i \neq j}}^{N} w_{ji} \cdot I_{ji}(t) + e_j'$$

Sequencer and buffers



Schematic



Schematic

Attenuator and Gilbert cell:

$$w_{ji} = f\left[G_{ji}\right] \in [-1, +1];$$

$$I_{ji,w}(t) = w_{ji} \cdot I_{ji}(t).$$

Summation nodes:

$$I_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^{N} w_{ji} \cdot I_{ji}(t)$$

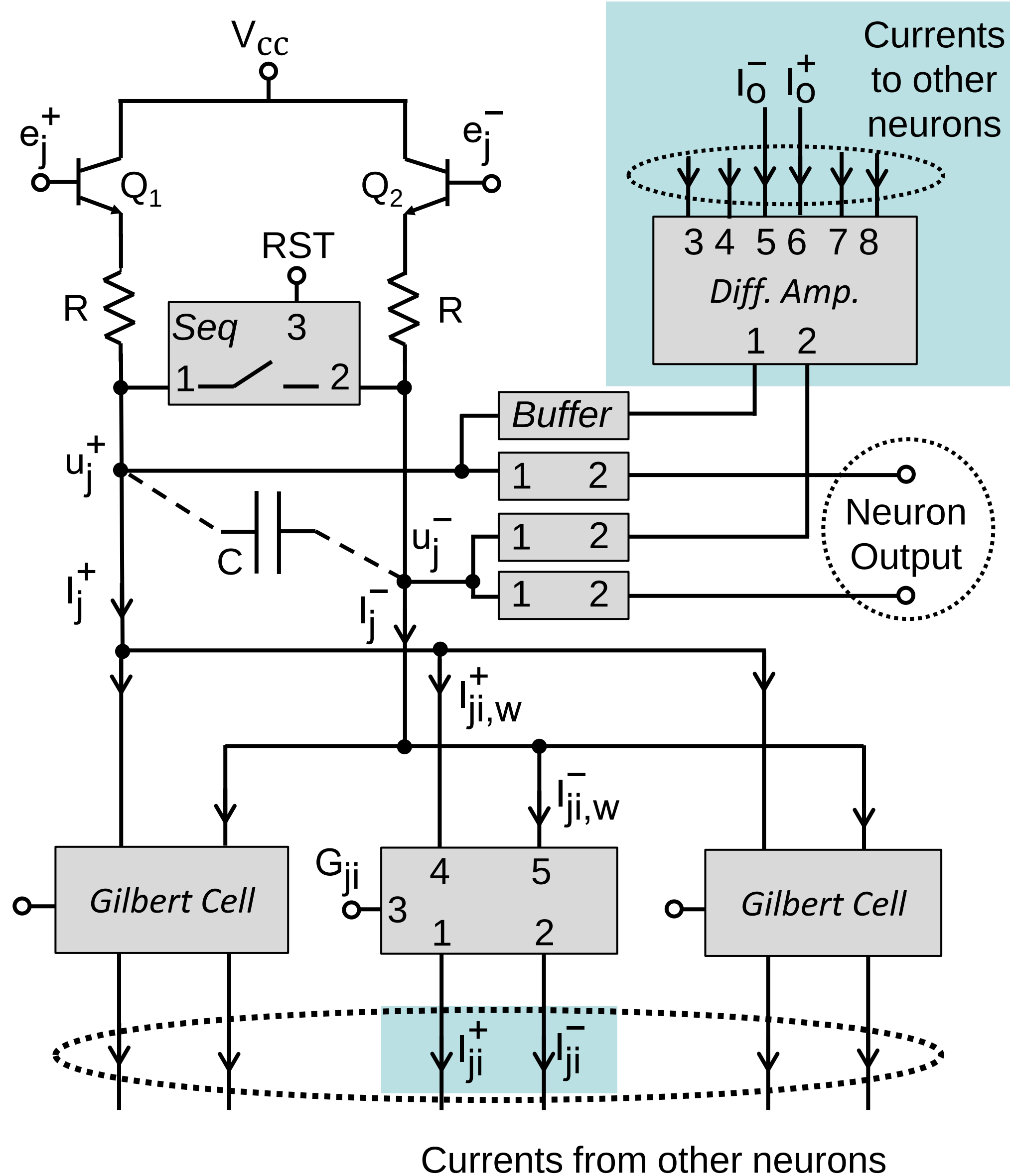
Inner state:

$$\tau \cdot \frac{du_j'(t)}{dt} = -u_j'(t) - R \cdot \sum_{\substack{i=1 \\ i \neq j}}^{N} w_{ji} \cdot I_{ji}(t) + e_j'$$

Sequencer and buffers

Differential pairs:

$$I_o(t) = -\frac{I_t}{(N-1)} \cdot \tanh\left(\frac{u_j'(t)}{2 \cdot V_t}\right)$$

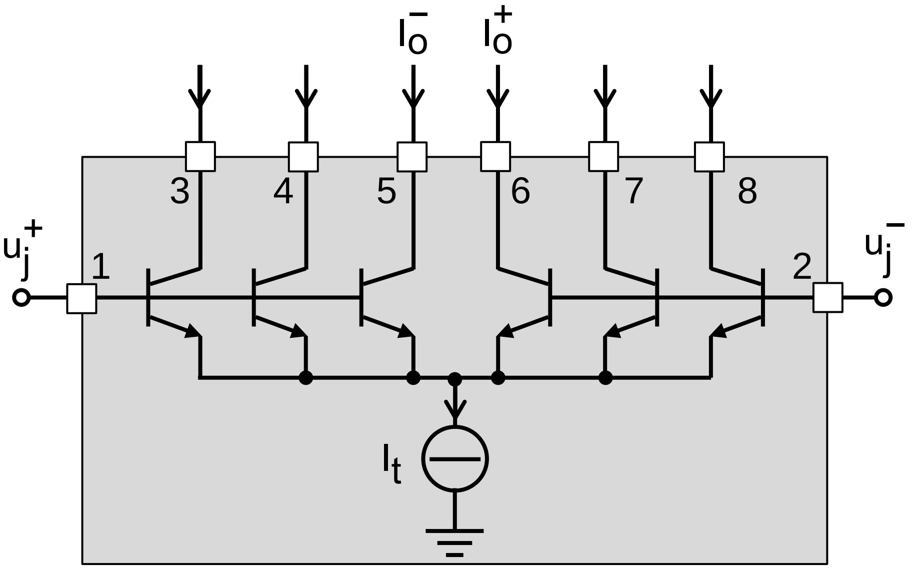


Schematic

Current from j^{th} neuron, directed the i^{th} neuron:

$$I_o(t) = -\frac{I_t}{(N-1)} \cdot \tanh\left(\frac{u_j'(t)}{2 \cdot V_t}\right)$$
$$= I_{ij}(t)$$

Inverting the indexes, the feedback current $I_{ji}(t)$ to the j^{th} neuron, coming from the i^{th} neuron can be found.



Differential pairs

Schematic

What we obtain:

$$\boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{u}'(t)}{\mathrm{d}t} = -\boldsymbol{u}'(t) + \boldsymbol{W} \cdot \boldsymbol{v}'(t) + \boldsymbol{e}',$$

$$\boldsymbol{v}'(t) = \frac{R \cdot I_t}{N-1} \cdot \tanh\left(\frac{\boldsymbol{u}'(t)}{2 \cdot V_t}\right).$$

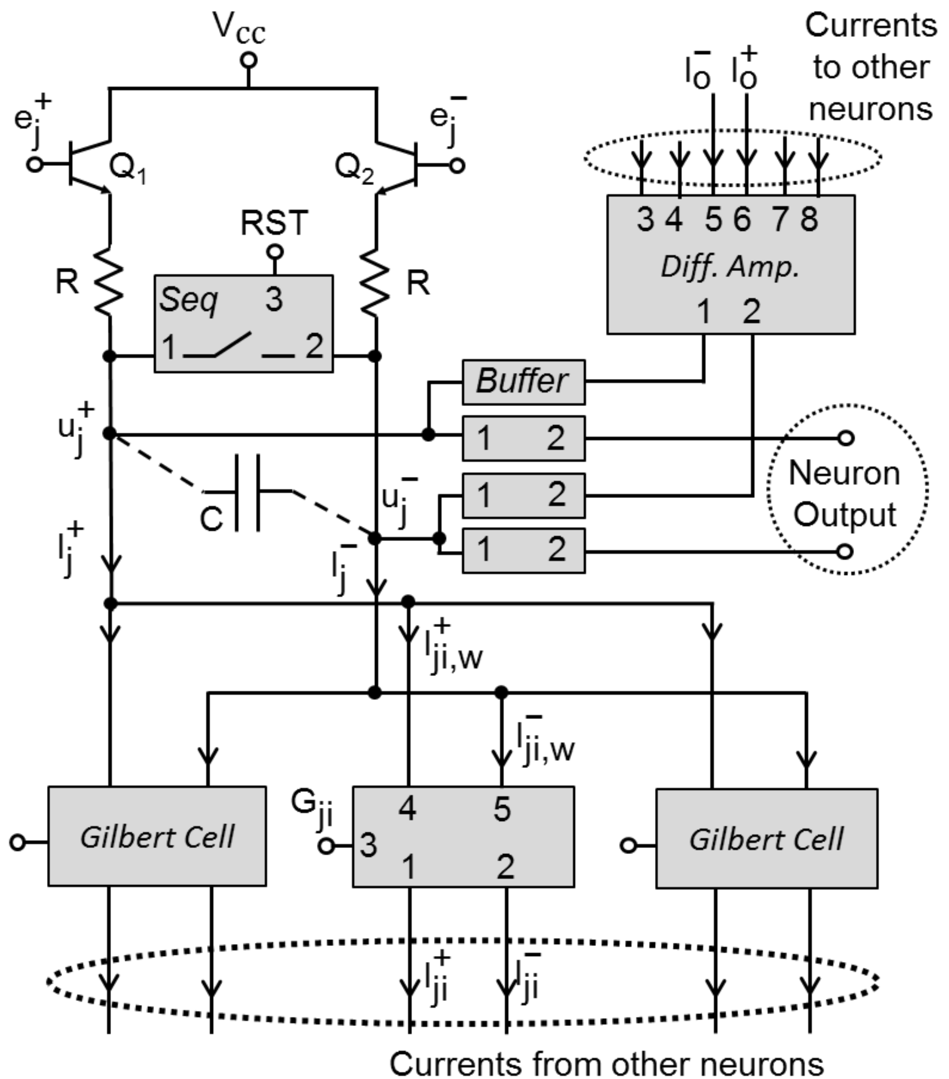
Dynamical behavior of the RNN:

$$\boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{u}'(t)}{\mathrm{d}t} = -\boldsymbol{u}'(t) + \boldsymbol{W} \cdot \boldsymbol{v}'(t) + \boldsymbol{W}_0 \cdot \boldsymbol{e}',$$

$$\boldsymbol{v}'(t) = S \cdot \alpha \cdot \tanh\left(\frac{\beta \cdot \boldsymbol{u}'(t)}{S}\right).$$

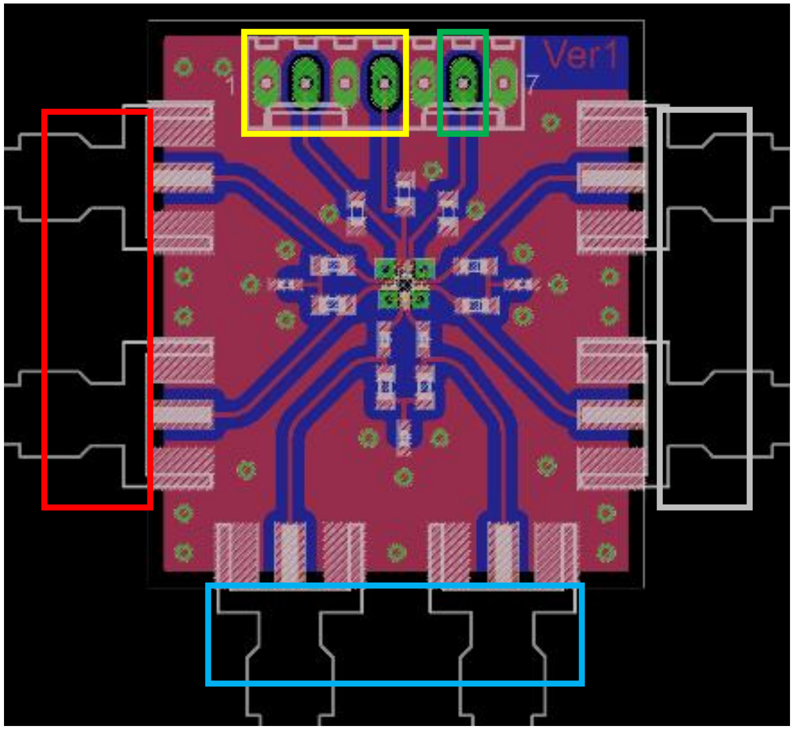
Equivalence:

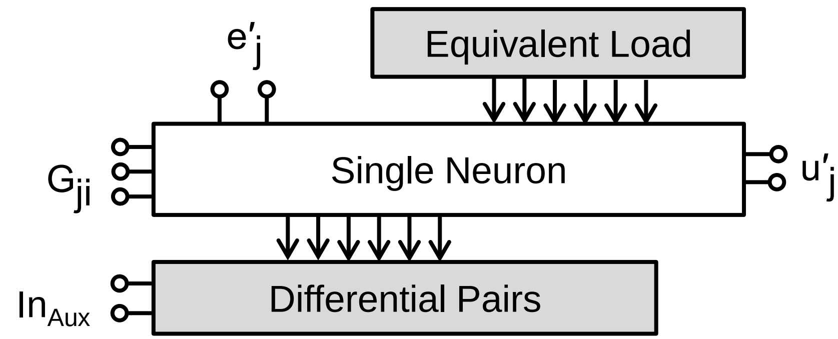
- $\boldsymbol{W}_0 = \boldsymbol{I}$;
- $\alpha = 1\ \mathrm{V}$;
- $S \cdot \alpha = (R \cdot I_t)/(N-1)$;
- $\beta = S/(2 \cdot V_t)$.



Single neuron: PCB layout

PCB: (32 x 36) mm^2 on Roger © 3003

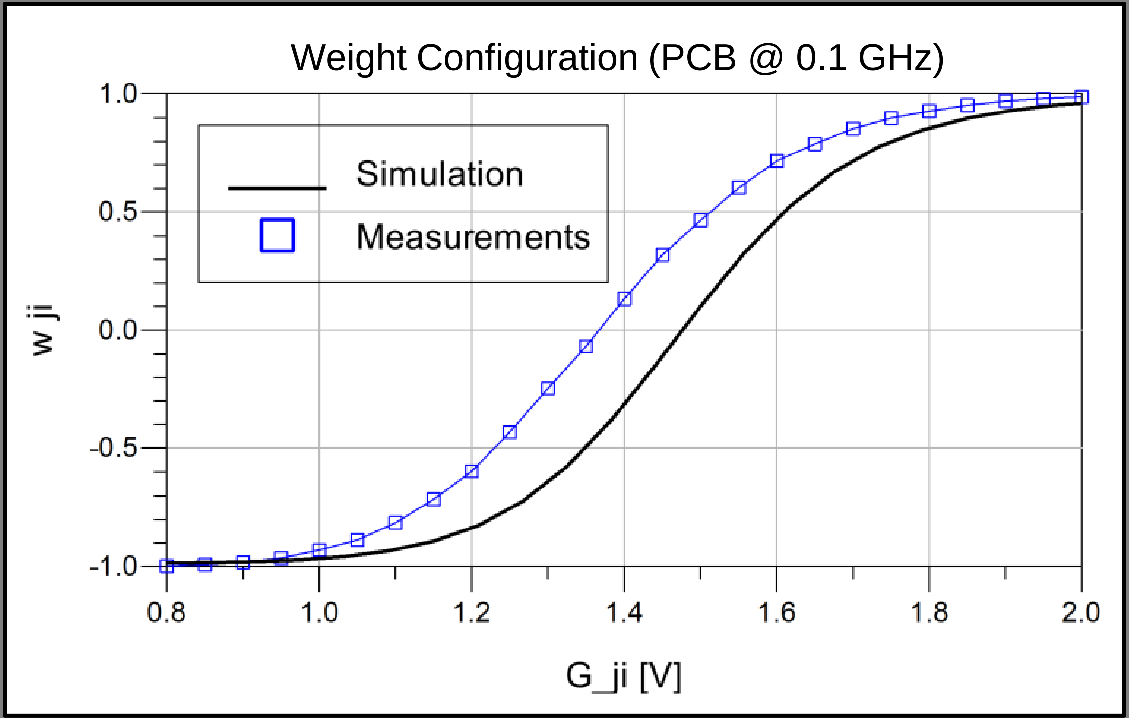




Embedding:
- Ideal lumped components;
- Transmission lines;
- Capacitors to ground at the Pads.

- Buffered inner state u'_j
- Weight configuration G_{ji}
- Auxiliary input In_{Aux}
- Power supplies
- External input e'_j

Single Neuron: Measurements

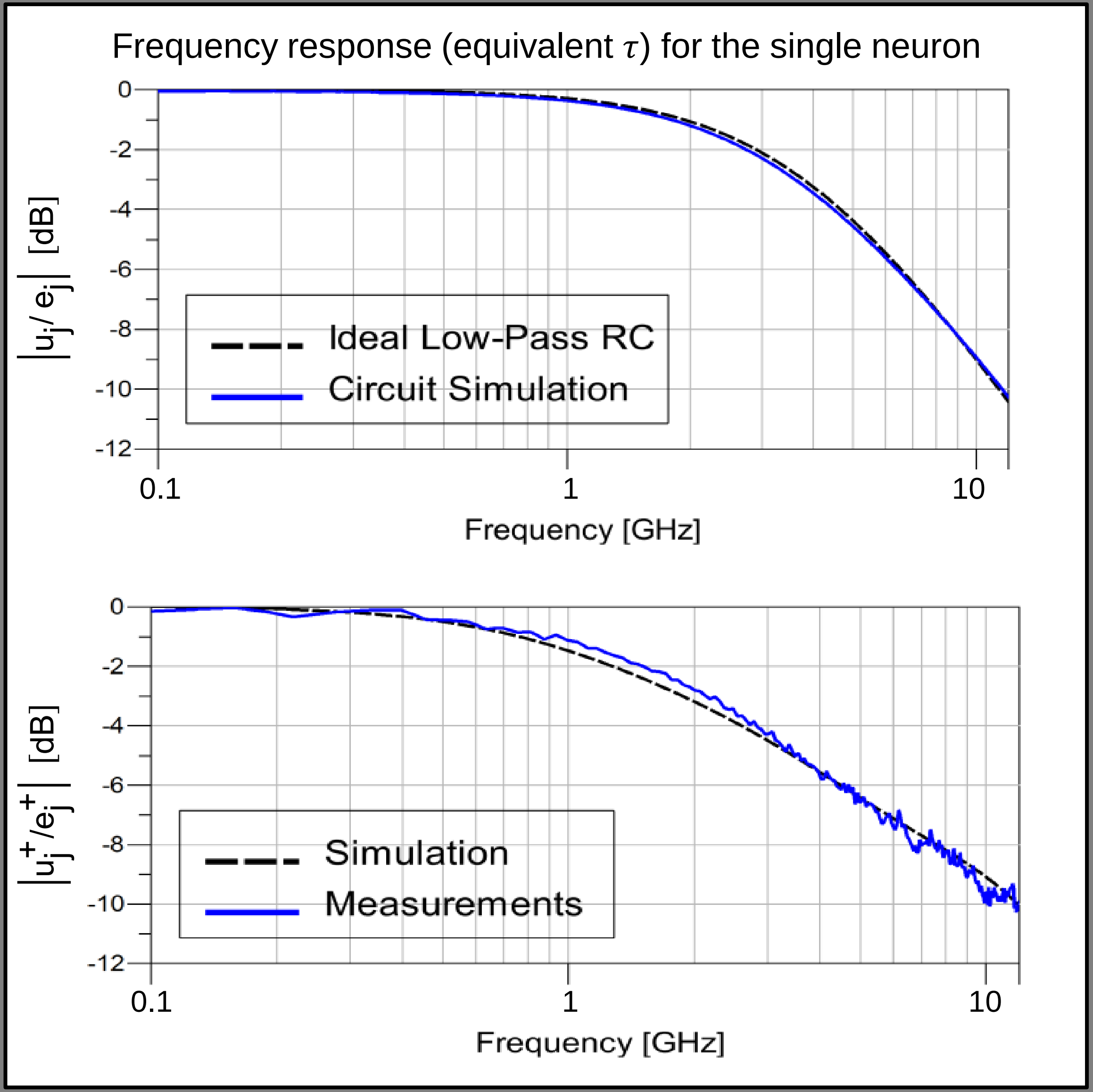


Hyperbolic tangent slope

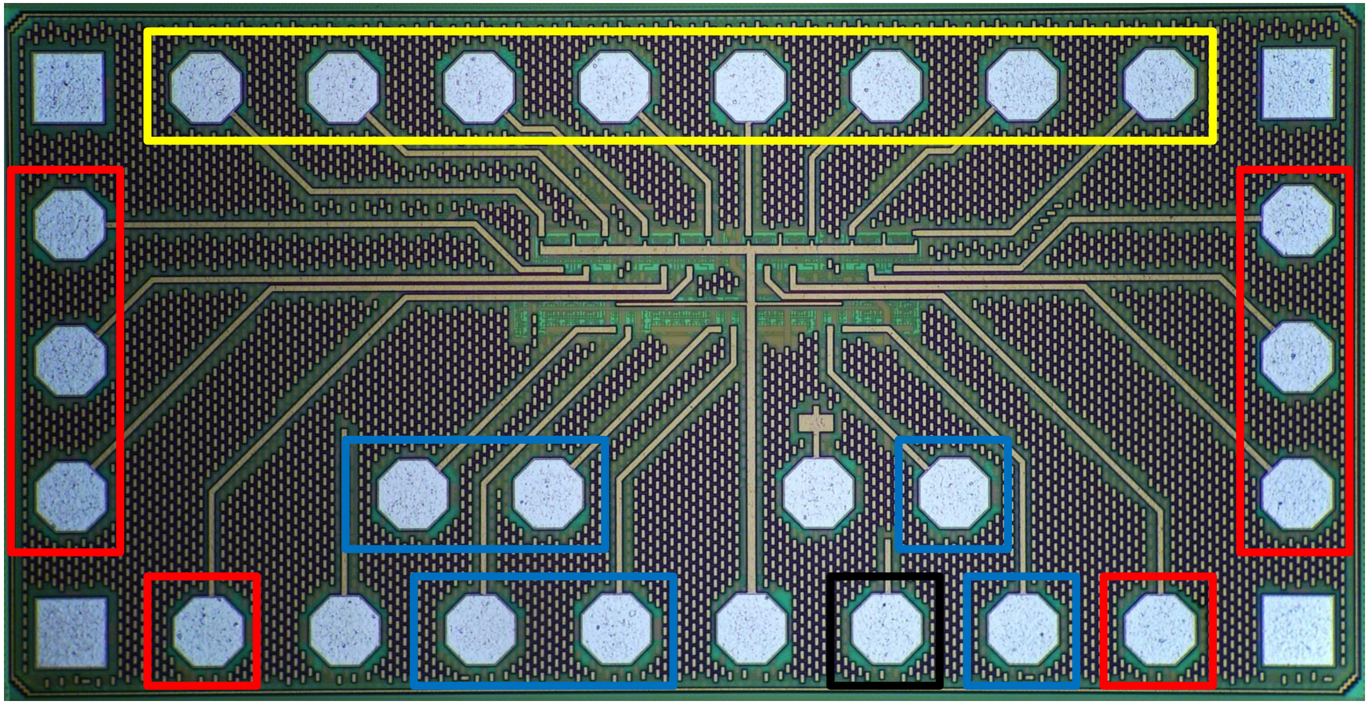
$$\varphi[\boldsymbol{u}(t)] = \alpha \cdot \tanh\big(\beta \cdot \boldsymbol{u}(t)\big)$$

$\beta_{\mathrm{opt}} \geq 3\ \mathrm{V}^{-1}$ $\beta_{\mathrm{sim}} = 3.87\ \mathrm{V}^{-1}$ $\beta_{\mathrm{meas}} = 3.47\ \mathrm{V}^{-1}$

Single Neuron: Measurements



Equalizer: Chip Layout



- Chip area: (1.49 x 0.46) mm^2
- 30 pads (4 grounds)
- Power consumption: 35 mW

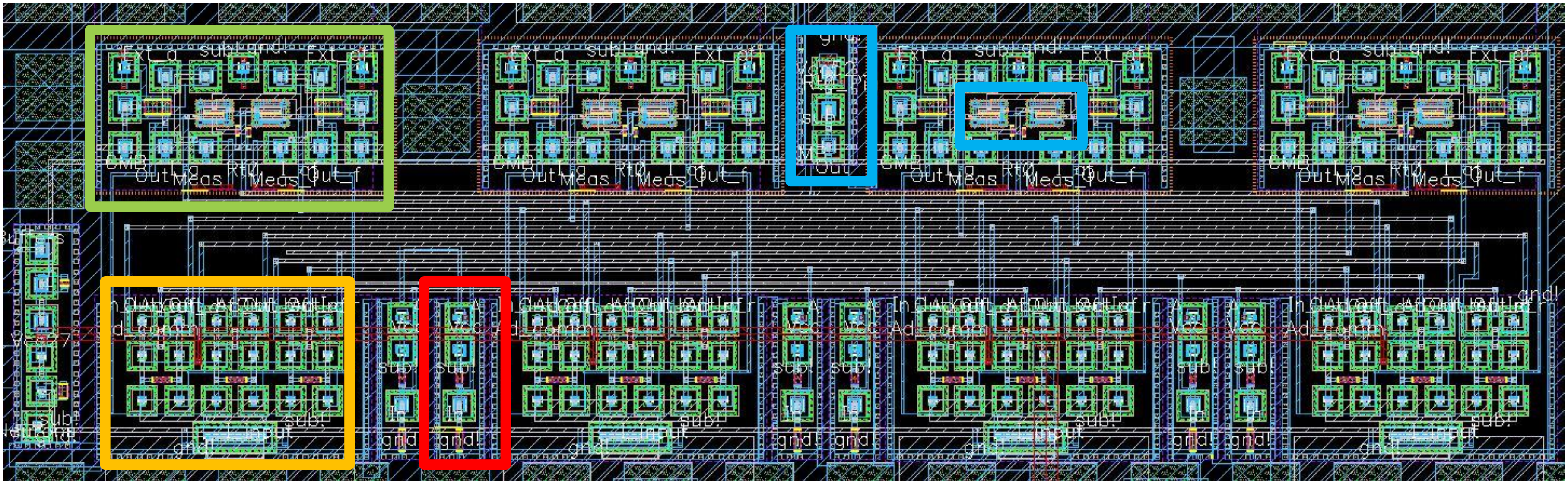
External inputs (differential)

Buffered outputs (differential)

Weights control

Reset

Equalizer: Active Area Layout



- ~ 170 transistors / 4 neurons
- Transistors ~ N^2
- Feedbacks ~ N^2

Differential amplifiers and Gilbert cells

Weight configuration

Load and buffers

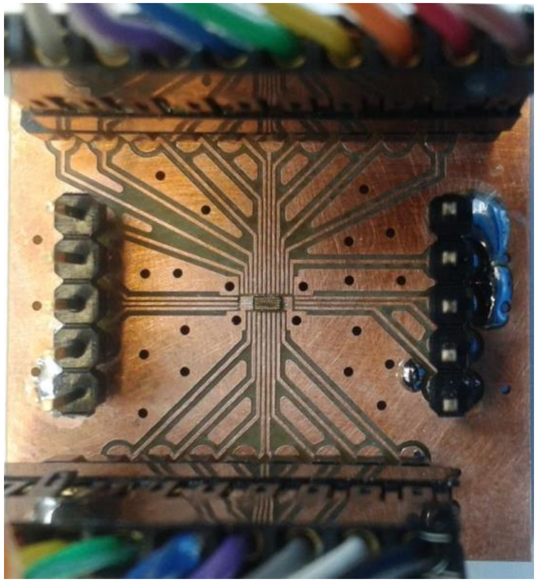
Integrated reset function

Measurements

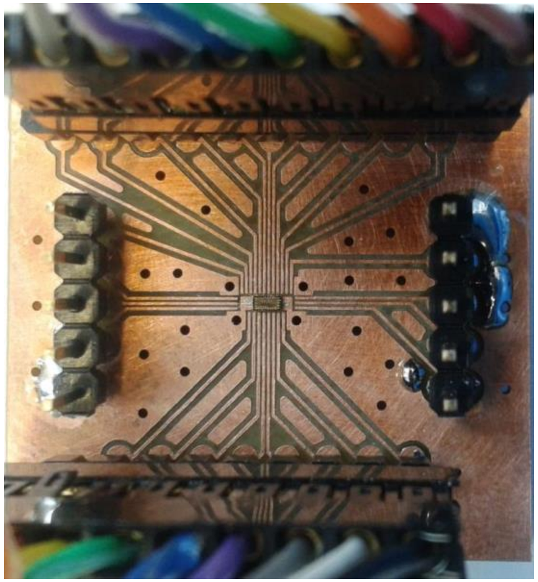
$$\boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{u}'(t)}{\mathrm{d}t} = -\boldsymbol{u}'(t) + \boldsymbol{W} \cdot \boldsymbol{v}'(t) + \boldsymbol{e}',$$

$$\boldsymbol{v}'(t) = \frac{R \cdot I_t}{N-1} \cdot \tanh\left(\frac{\boldsymbol{u}'(t)}{2 \cdot V_t}\right).$$

11 Variables $(\boldsymbol{W}, \boldsymbol{e}', t)$



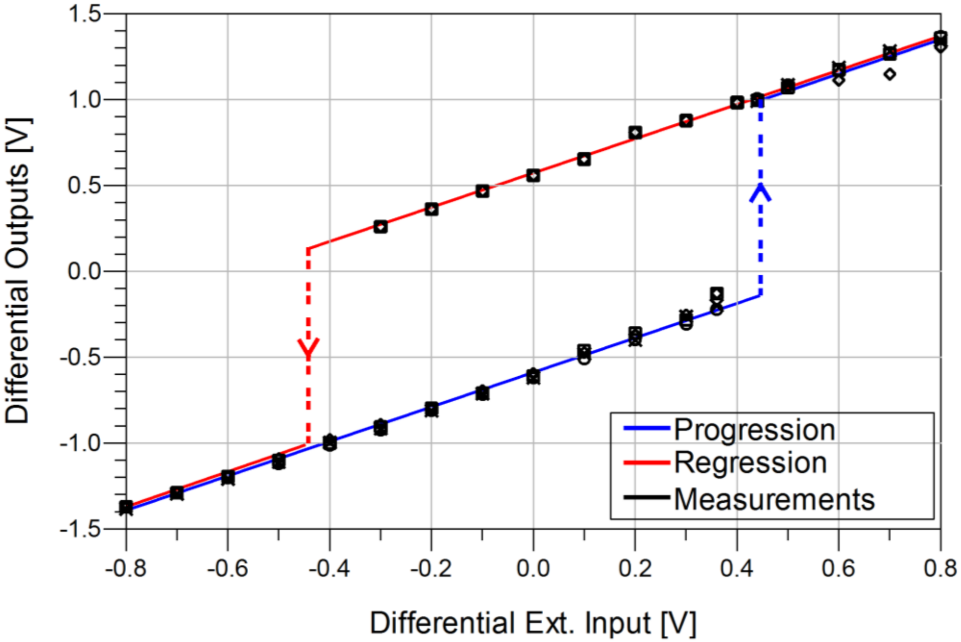
Measurements



$$\boldsymbol{\Upsilon} \cdot \frac{\mathrm{d}\boldsymbol{u}'(t)}{\mathrm{d}t} = -\boldsymbol{u}'(t) + \boldsymbol{W} \cdot \boldsymbol{v}'(t) + \boldsymbol{e}',$$

$$\boldsymbol{v}'(t) = \frac{R \cdot I_t}{N-1} \cdot \tanh\left(\frac{\boldsymbol{u}'(t)}{2 \cdot V_t}\right).$$

11 Variables $(\boldsymbol{W}, \boldsymbol{e}', t)$



Slow-varying transfer function

- Equal external inputs;
- Equal weights $(w_{ji} = 1,\ \forall\, j \neq i)$;
- No time dependence;

$$u_j' - \frac{R \cdot I_t}{(N-1)} \cdot \sum_{\substack{i=1 \\ i \neq j}}^{N} \tanh\left(\frac{u_i'}{2 \cdot V_t}\right) = e_j'$$

Conclusions

Conclusions

Summary

- Revitalize nonlinear signal processing with analog circuits.
- Vector equalization using recurrent neural networks:
  - Validation of the circuit model through simulation;
  - Key parameters for best performance.
- Novel 4-neurons fully adaptive vector equalizer:
  - Throughput: ~16 Gbps;
  - Equalization time: ~250 ps;
  - Power: 35 mW;
  - Integrated reset function.
- Measurements:
  - Single neuron characterization;
  - Equalizer static transfer function.
- Outlook:
  - Observe the equalizer outputs at real-time;
  - Extend the concept to complex equalization and decoding;
  - Increase the number of neurons (scalability issues).

Thank you for your attention!

Acknowledgments

- Deutsche Forschungsgemeinschaft (DFG)
- Ulm University - Institute of Communications Engineering
- IHP, Frankfurt/Oder, Germany