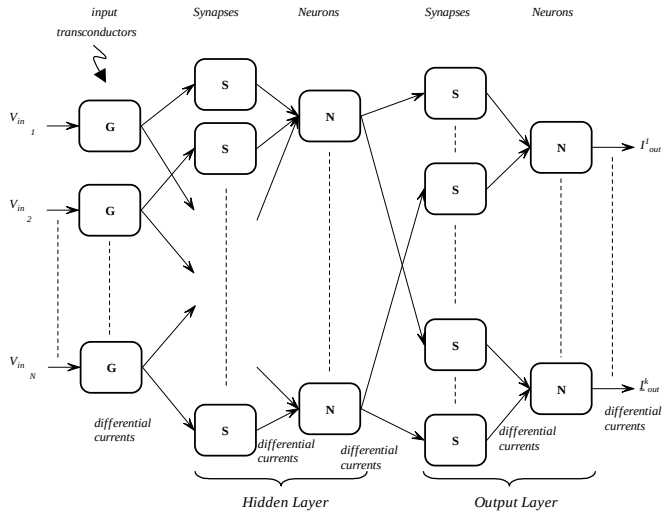
Analog Weight Perturbation on-chip learning

Maurizio Valle

Feed forward circuit architecture: main features

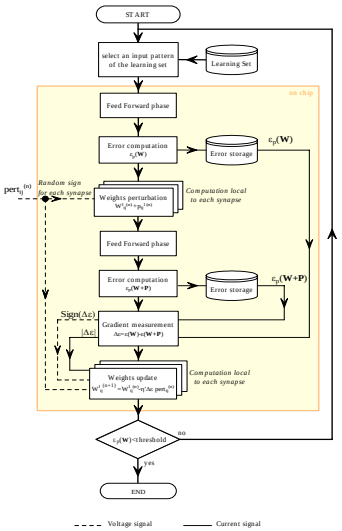
- Current mode circuits
- Translinear circuits
- Information coding through differential balanced signals
- Low Power / Low Voltage *i.e.: weak inversion region of operation of devices, supply voltage in the range [2.5÷3]V or lower*
- Implementation of normalised sums of currents: *both for the synaptic output currents and the output error (i.e. target - output neuron current) currents;*
- Programmable neuron circuit gain;
- Trade off between power and area i.e.: *the maximum transistor bias current has been set to 500nA.*

Feed forward circuit architecture

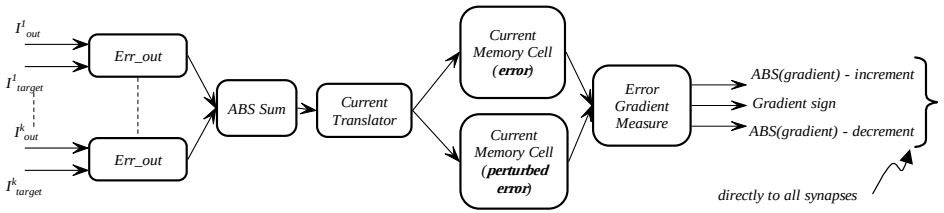


WP learning algorithm

- *select an input pattern of the learning set*
- *feed forward phase*
- *output error computation and storage*
  (i.e. $\varepsilon_p(w)=\Sigma|$(target output value – neuron output value)$|$)
- *weights perturbation*
- *feed forward phase*
- *perturbed output error computation and storage*
  (i.e. $\varepsilon_p(w+p)=\Sigma|$(target output value – neuron output value)$|$)
- *"measurement" of the error gradient $\Delta\varepsilon_p$*
  (in terms of the absolute value and sign)
- *back propagation to all synapses of the two signals: absolute value and sign of $\Delta\varepsilon_p$*
- *weights update: $w=w+\Delta w$*
- *if $\varepsilon_p(w)$ > given threshold goto point 1*
  elseif end

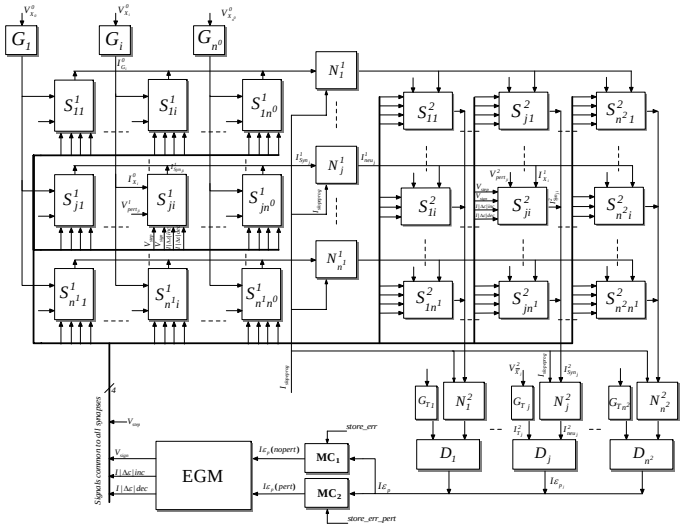


Circuits blocks common to the whole architecture



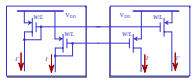
$$\Delta w_{ij} \cong -\eta \frac{\varepsilon(w + p_{ij}^{(n)}) - \varepsilon(w_{ij})}{p_{ij}^{(n)}}$$

Analog circuit MLP architecture with on-chip WP learning

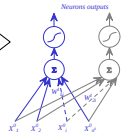


Circuit design criteria

- The current-mode circuit design approach → *Current mirrors, multipliers, adders*
- Information coding through differential balanced signals → *linearity increase and noise minimization*
- Current signals distribution



- Current signals normalization

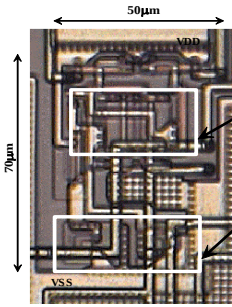


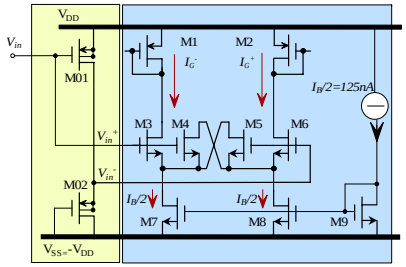
- The translinear principle (multiplier)
  - The gm/Id design methodology → $\frac{I_{DS}}{W} \le 6.25 \cdot 10^{-3}\left[\frac{A}{m}\right]$
- Bandwidth requirements → $\frac{I_{DS}}{W} \ge 2/7\, 10^{-3}\left[\frac{A}{m}\right]$
- Translinear loop devices sizes → $I_{DS} \approx 10^{-8} A$ *and* $W \approx 10^{-5} m$ → $I_{DS}=250nA$ $W=40\mu m.$

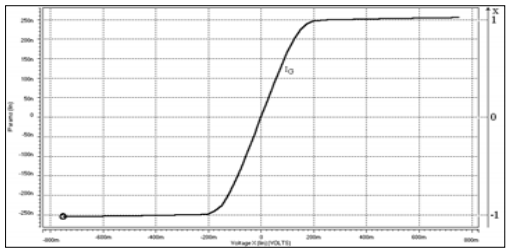
The input transconductor (block G)

- **I_G in the range [-250nA÷250nA]**
- **The cut-off frequency is about 1MHz.**
- **THD of I_G: V_{in} input sinusoidal voltage signal with 200mV peak to peak**

10KHz	100KHz	1MHz
0.8%	1.5%	3.2%



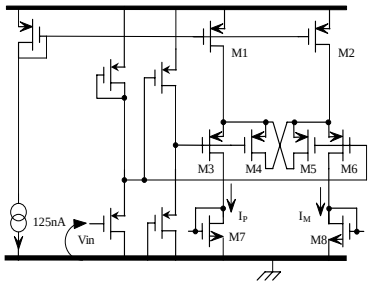


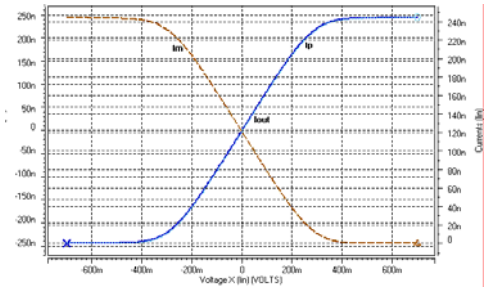


Weight Transconductor

• The linearity range of single ended to differential block is greater than “traditional” transconductor

• The symmetrical balanced voltages in input to transconductor permit to have right asymptotes [-250nA÷250nA].





Single ended to differential block

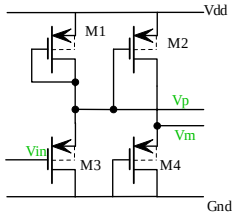
Suppose M_i ($i=1..4$) equal MOS in saturation region:

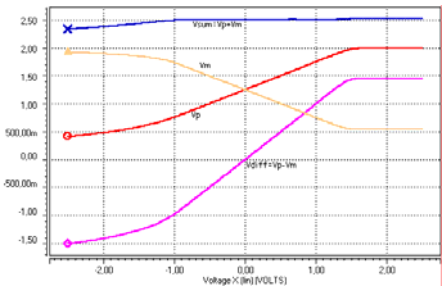
$I_{M1}= I_{M3}$, i.e.:
$K[(Vp-Vin)-V_{th}]=K[(Vdd-Vp)-V_{th}]$ and so
$Vp=½(Vdd+Vin)$

At the same time, $I_{M2}= I_{M4}$, i.e.:
$K[(Vdd-Vp)-V_{th}]=K[Vm-V_{th}]$ and so
$Vm=½(Vdd-Vin)$

Vsum=Vp+Vm=Vdd
Vdiff=Vp-Vm=Vin

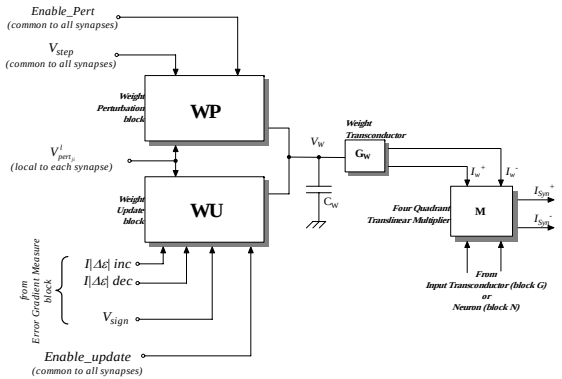
Vp and Vm are differential balanced voltage



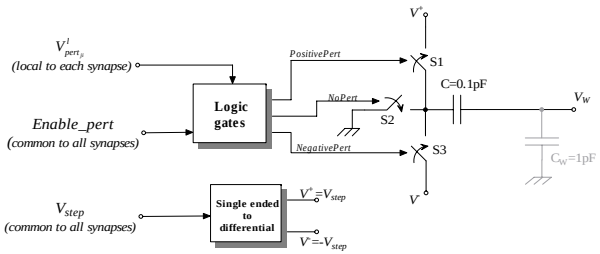


The synaptic circuit (block S)

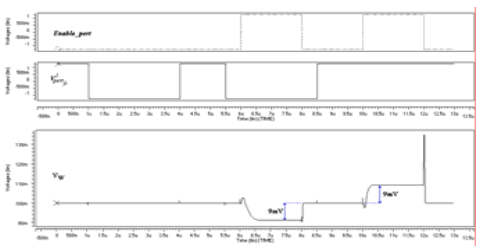
- **In the design of the synapse we have to take into account:**
  - the circuitry necessary to perform the feed-forward phase and,
  - the learning circuitry necessary for the perturbation and the update of the weight voltage in according to the Weight Perturbation algorithm.



The synaptic circuit: weight perturbation (block WP)



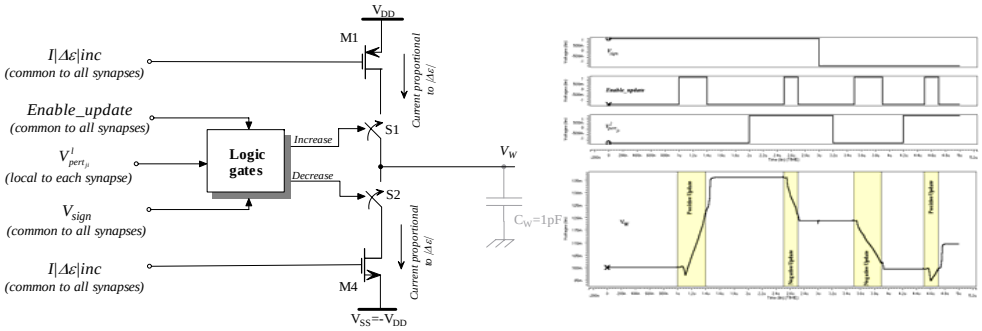
- **if pert=“low”**
  - Negative perturbation, i.e.
    $V_W{}^k{}_{perturbed}=V_W{}^k-\Delta V_{Wpert}$
- **if pert=“high”**
  - Positive perturbation, i.e.
    $V_W{}^k{}_{perturbed}=V_W{}^k+\Delta V_{Wpert}$



$$\Delta V_{W\,pert} = \frac{V_{step}}{11}$$

Inputs		Outputs			Note
Enable_pert	$V^l_{pert\,ji}$	NoPert	PositivePert	NegativePert	
0	X	1	1	0	No perturbation
1	0	0	1	1	Negative Perturbation
1	1	0	0	0	Positive Perturbation

The synaptic circuit: weight update (block WU)



- **Taking into account:**

$$\Delta W^{\,l}_{ji} = -\eta' \cdot \Delta\varepsilon_p \cdot pert^{\,l}_{ji} \qquad \eta' = -\eta/step$$

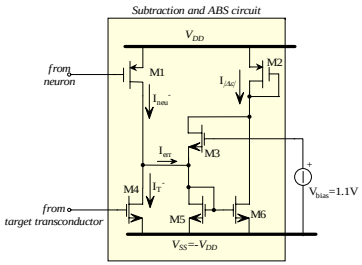
$$pert^{\,l}_{ji} = \begin{cases} +1 \\ -1 \end{cases} \; With\ equal\ probability$$

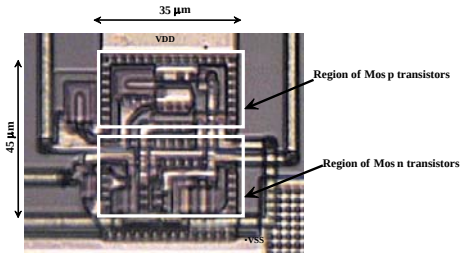
$$l \in [1 \div N],\ j \in [1 \div n^{l}]\ i \in [1 \div n^{l-1}]$$

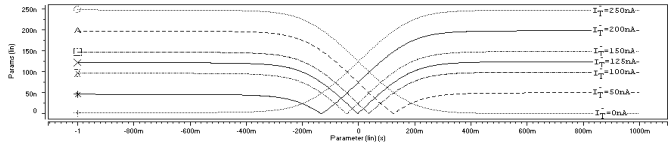
Inputs			Outputs		Note
Enable_update	V^{l}_{pertji}	V_{sign}	Increase	Decrease	
0	X	X	1	0	No update
1	0	0	1	1	Weight Decrement
1	0	1	0	0	Weight Increment
1	1	0	0	0	Weight Increment
1	1	1	1	1	Weight Decrement

The substract and ABS circuit (block D)

The block D estimates the error function. It computes the module of the difference between the neuron output current and the output current of the transconductor that codifies the target information.

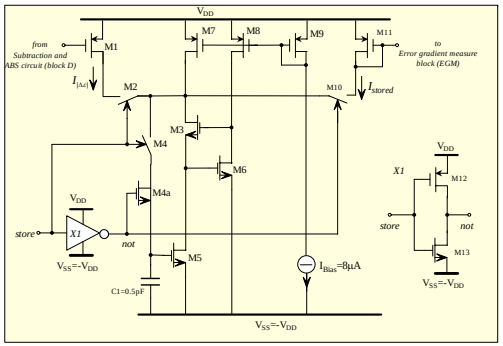


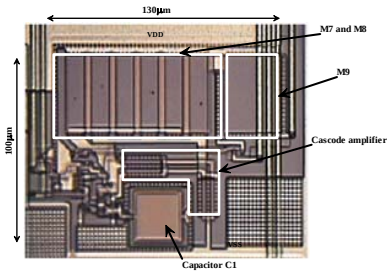




The current memory cell (block MC)

The current memory cell is used to store the current that codifies the value of the error function before and after the weight perturbation, i.e. the current $I_{|\Delta\varepsilon|}$



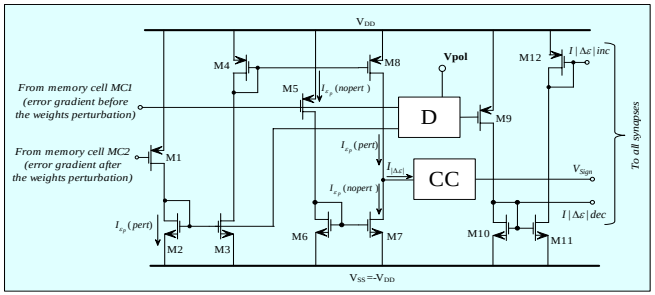


The error gradient measure (block EGM)

The EGM block provides to each synapse the sign of the gradient V_{sign} and the gates voltages of the diode connected transistors carrying the current coding the absolute value of the error function gradient ($I_{|\Delta\varepsilon|}$)

The inputs of the EGM block are:

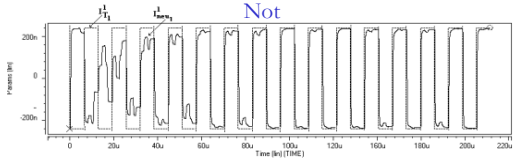
- the current value coding the error gradient <u>before</u> the weights perturbation (i.e. $I_{ep}(nopert)$)
  the current value coding the error gradient <u>after</u> the weights perturbation (i.e. $I_{ep}(pert)$).



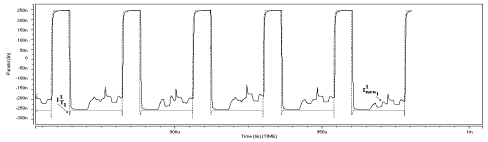
Simulation results

Function	Number of layer N	Network topology	Input bias	Number of synapses	Pattern presentation time	Vstep [mV]	Time duration of Enable_update [ns]	Convergence	
								#Epoch	Time
Not	1	1×1	0	1	6ms	200	200	10	60ms
And	1	2×1	1	3	6ms	200	100	30	720ms
Or	1	2×1	1	3	6ms	200	100	30	720ms
Ex-Or	2	2×2×1	1+1	9	6ms	20	100	800	19.2ms

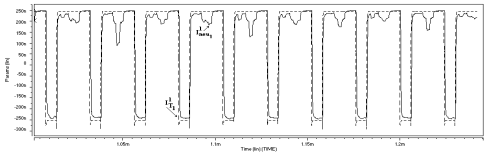
Not



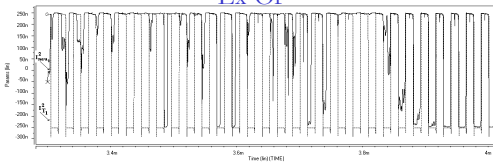
And



Or



Ex-Or



Conclusions (1)

	Computational density [CUPS/mm2]	Energy efficiency [CUPS/mW]
[Lehmann94]	**22.5×10^3**	**13.5×10^3**
[Berg96]	**1.15**	**92**
[Cauwenberghs96]	**8.7×10^3**	**35×10^3**
[Morie99]	**20.4×10^3**	**32×10^3**
[Bo00]	**216×10^3**	**106×10^3**
This architecture	**11×10^6**	**62×10^6**