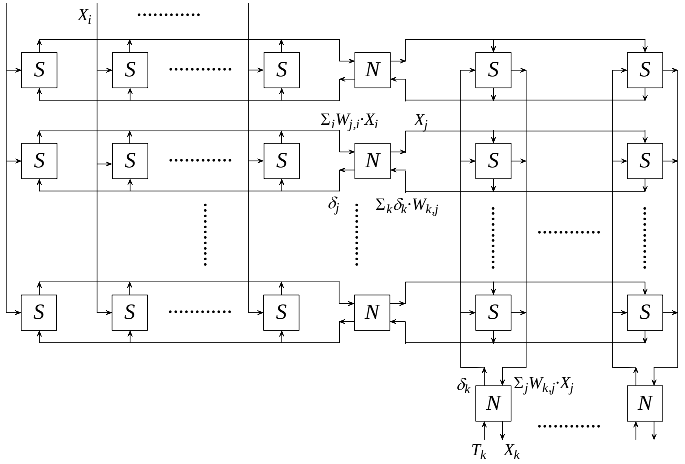
Low Power Design Techniques and Neural Applications
Barcelona, Feb. 23-27 2004

Analog Back Propagation on-chip learning

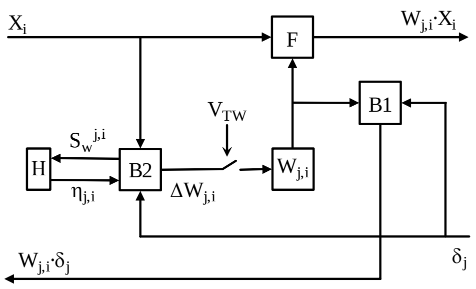
Maurizio Valle

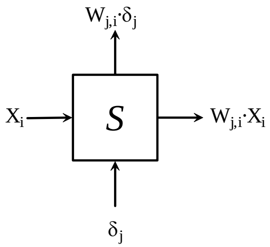
On-chip learning architecture (architectural mapping)



X_i
S
N
$\Sigma_i W_{j,i}\cdot X_i$
X_j
δ_j
$\Sigma_k \delta_k \cdot W_{k,j}$
δ_k
$\Sigma_j W_{k,j}\cdot X_j$
T_k
X_k

Synaptic module





The synaptic module

- **F is the feed-forward four-quadrant multiplier:** $W_{k,j}X_j$
- **B1 is the backward four-quadrant multiplier:** $\delta_k W_{k,j}$
- **B2 is the weight update four-quadrant multiplier:** $\Delta W_{k,j} = \eta_{k,j}\delta_k X_j$

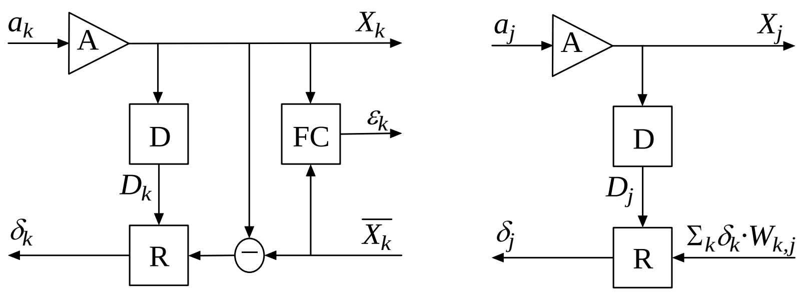
B2 generates also the sign *$S_{k,j}$*: $S_{k,j} = sign\left(\frac{\varepsilon_p}{\partial W_{k,j}}\right) = -sign\left(\Delta W_{k,j}\right)$

□ **H is the local learning rate adaptation circuit block**

- **WU is the weight block:** $W_{k,j}^{new} = W_{k,j}^{old} + \Delta W_{k,j}$

The WU performs also the short-term memorization of the weight value.

Neuron module



(a) (b)

Neuron module

- **A, activation function module:** $X_{k(j)} = \Psi(a_{k(j)})$
- **D, derivative module:** $D_{k(j)} = 1-(X_{k(j)})^2$
- **R, the error multiplier:** $\delta_k = (\overline{X}_k - X_k)D_k$ $\delta_j = (\sum_k \delta_k W_{k,j})D_j$
- **FC, the error circuit:** $\varepsilon_k = (\overline{X}_k - X_k)^2$

Correspondence tables between neural and electrical variables

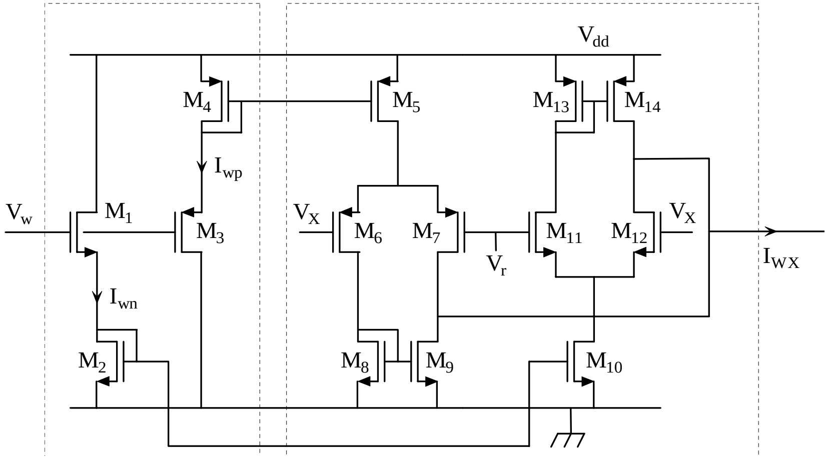
Synaptic Module	
Neural variables	Electrical variables
X_j	V_X
$W_{k,j}\cdot X_j$	I_{WX}
$W_{k,j}$	V_W
$\Delta W_{k,j}$	$I_{\Delta W}$
δ_k	V_δ
$\delta_k\cdot W_{k,j}$	$I_{\delta W}$
$\eta_{k,j}$	I_η
$S_{k,j}$	$V_{S\eta}$

Neuron Module	
Neural variables	Electrical variables
$a_{k(j)}$	I_a
$X_{j(k)}$	V_X
$\overline{X}_k$	V_T
$\Sigma_k\delta_k\cdot W_{k,j}$	$V_{\delta W}$
$\delta_{k(j)}$	V_δ
$D_{k(j)}$	I_d
ε_k	I_ε

On-chip learning algorithm

```
iterate on k
    select P in a random manner in the training set                     off-chip
    put P in input to the MLP                                           off-chip
    perform the feedforward phase                                       on-chip
    parallel for each synapse i^th, j^th
        compute $\Delta W_{j,i}(k) = -\eta_{j,i}(k)\cdot\frac{\partial E(k)}{\partial W_{j,i}}$    on-chip
        compute $S_{j,i}(k)$                                            on-chip
        $W_{j,i}(k+1) = W_{j,i}(k) + \Delta W_{j,i}(k)$                 on-chip
        if $S_{j,i}(k) = S_{j,i}(k-1)$                                  on-chip
            $\eta_{j,i}(k+1) = \eta_{j,i}(k)\cdot\left[\frac{\eta^{\max}}{\eta_{j,i}(k)}\right]^\gamma$    on-chip
        else
            $\eta_{j,i}(k+1) = \eta_{j,i}(k)\cdot\left[\frac{\eta^{\min}}{\eta_{j,i}(k)}\right]^\gamma$    on-chip
        endif
    end parallel
until convergence is reached                                            off-chip
```

F and B1 four-quadrant multiplier



The Ψ Block

The Ψ block is a non-linear transconductor that converts the weight voltage V_w into a differential current I_w=I_{wp}-I_{wn}. Being equal the aspect ratio (i.e., *W/L*) of M_1 and M_2 as well for M_3 and M_4, and supposing all of them biased in strong inversion, we can write:

$$I_{wn} = \begin{cases} \beta_n (V_w - V_{th1} - V_{th2})^2 & V_w \geq V_{th1} + V_{th2} \\ 0 & V_w < V_{th1} + V_{th2} \end{cases}$$

$$I_{wp} = \begin{cases} \beta_p (V_w - V_{th3} - V_{th4})^2 & V_w \leq V_{dd} + V_{th3} + V_{th4} \\ 0 & V_w > V_{dd} + V_{th3} + V_{th4} \end{cases}$$

where

$$\frac{1}{\sqrt{\beta_n}} = \frac{1}{\sqrt{\beta_1}} + \frac{1}{\sqrt{\beta_2}} \qquad \frac{1}{\sqrt{\beta_p}} = \frac{1}{\sqrt{\beta_3}} + \frac{1}{\sqrt{\beta_4}}$$

β_i and V_{thi} are the gain factors and the threshold voltages of M_i (i=1÷4) respectively.

The OTA Block

The resulting differential current I_w can be written as:

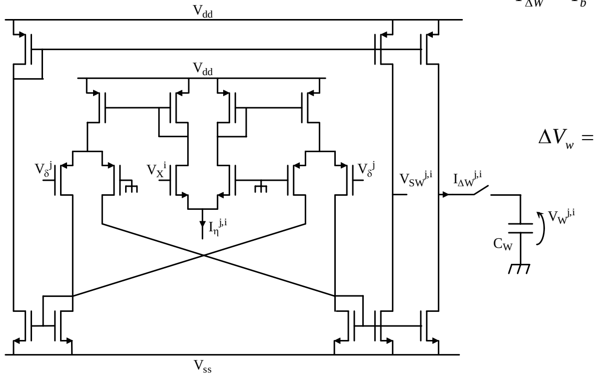
$$I_{WX} = (I_{wp} - I_{wn}) \tanh(\frac{V_X - X_r}{2nU_T}) = g_w(V_W) \tanh(\frac{V_X - X_r}{2nU_T})$$

where *n* is the weak inversion slope coefficient U_T is the thermal voltage, and V_r is the signal ground (i.e. the synaptic input is null for V_X=V_r).

If the value of the argument of the *tanh* function is small (i.e. $|V_X - V_r| \leq 100mV$), we can approximate it with its argument:

$$I_{WX} \cong \frac{1}{2nU_T} g_w(V_W)(V_X - X_r)$$

The B2 multiplier



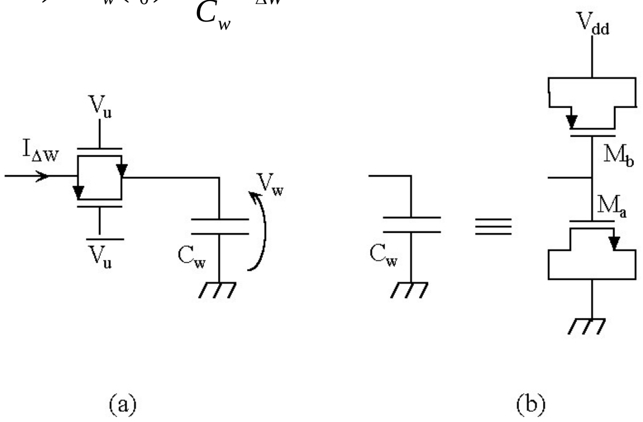
$$I_{\Delta W} = I_b \cdot \tanh\left(\frac{V_\delta}{2nU_T}\right) \cdot \tanh\left(\frac{V_S}{2nU_T}\right) \approx$$

$$\approx \frac{I_b}{4n^2U_T^2} \cdot V_\delta \cdot V_S$$

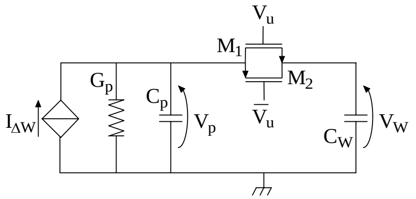
$$\Delta V_w = \frac{T}{C_w} \cdot I_{\Delta W} = \frac{T}{C_w} \frac{I_b}{4n^2U_T^2} \cdot V_\delta \cdot V_S$$

The Weight Unit

$$V_w(t_0+T)=V_w(t_0)+\frac{T}{C_w}I_{\Delta W}$$



Weight update circuit (1)



$$\Delta V_w=\frac{T}{C_w+C_p}\cdot I_{\Delta W}+\Delta V_{ci}+\Delta V_{cs}$$

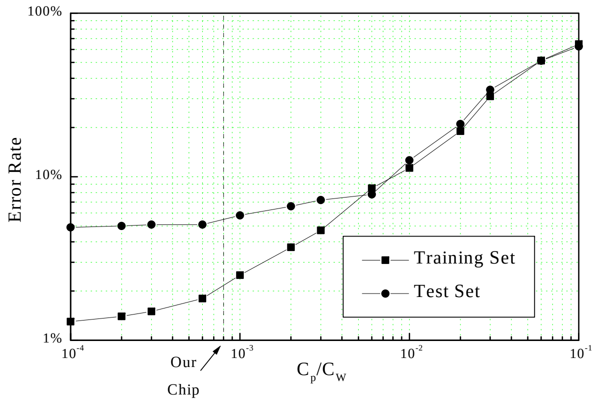
Ideal Weight Update

Charge Sharing Term
(hundreds millivolts)

Charge Injection Term
(few millivolts)

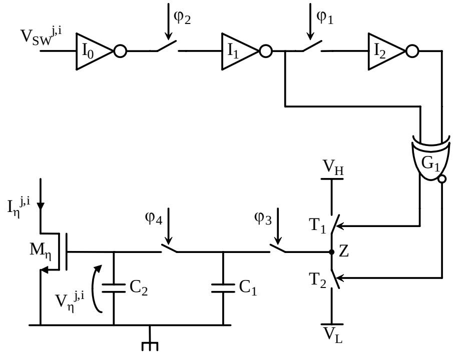
The main error term is due to the charge sharing between C_p and C_W when the switch is closed. The value of ΔV_{cs} depends on the values of C_p and C_w.

Weight update circuit (2)

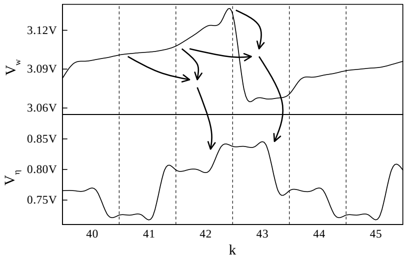


- Learning task: character recognition
- Network topology: 112×32×10 MLP
- Training set: 1000 char
- Test set: 1000 char

Local learning rate adaptation circuit (H) (1)



φ_i: four phases
non-overlapping clock



Local learning rate adaptation circuit (H) (2)

⇒ At the beginning all the φ_n are low: B2 has computed $I_{\Delta W}(t)$ and $S(t)$; $S(t)$ is already in input to the inverter I$_0$, and $S(t-1)$ [computed during the *(t-1)*th learning iteration] is in input to the inverter I$_1$.

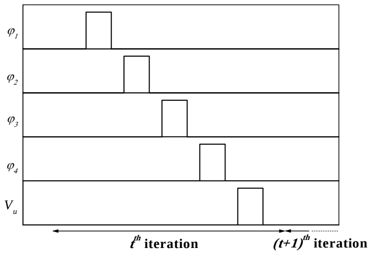
⇒ φ_1 high: $\overline{S}(t-1)$ is at input B of G$_0$.

⇒ φ_2 high: $S(t)$ is at input A of G$_0$. If $S(t)$ and $\overline{S}(t-1)$ have equal values, node C is connected to the voltage V_H (switch T$_1$ closed), otherwise node C is connected to the voltage V_L (switch T$_2$ closed).

⇒ φ_3 high: the value of $V_{C1}(t)$ is set as follows:

$$V_{C1}(t)=\begin{cases} V_H, & \mathrm{S(t)=S(t-1)} \\ V_L, & \text{otherwise} \end{cases}$$

φ_4 high: the capacitors C$_1$ and C$_2$ [C$_2$ stores the voltage $V_\eta(t)$] perform the charge sharing.



Local learning rate adaptation circuit (H) (3)

The updated value of the learning rate control voltage *$V_\eta(t+1)$* is given by the following expression:

$$V_\eta(t+1)=\frac{C_2}{C_1+C_2}V_\eta(t)+\frac{C_1}{C_1+C_2}V_{C1}(t)$$
$$=V_\eta(t)+\frac{C_1}{C_1+C_2}(V_{C1}(t)-V_\eta(t))$$
$$=V_\eta(t)+\gamma(V_{C1}(t)-V_\eta(t))$$

where $\gamma=\dfrac{C_1}{C_1+C_2}$.

Being the transistor M$_\eta$ biased in weak inversion, the new value of the learning rate current *$I_\eta(t+1)$* is given by:

$$I_\eta(t+1)=I_s\cdot e^{V_\eta(t+1)/nU_T}$$

where *I_s* is the specific current of the transistor M$_\eta$.

Local learning rate adaptation circuit (H) (4)

$$I_\eta(t+1)=I_\eta(t)\cdot\left(\frac{I_{C1}(t)}{I_\eta(t)}\right)^\gamma$$

$$I_{C1}(t)=I_s\cdot e^{V_{C1}(t)/nU_T}$$

The previous equation implements the learning rate adaptation rule, where $\gamma=\dfrac{C_1}{C_1+C_2}$, and *$I_{c1}(t)$* corresponds to *$\eta_{max}$* or *$\eta_{min}$*.

The maximum and minimum vales of the learning rate current are:

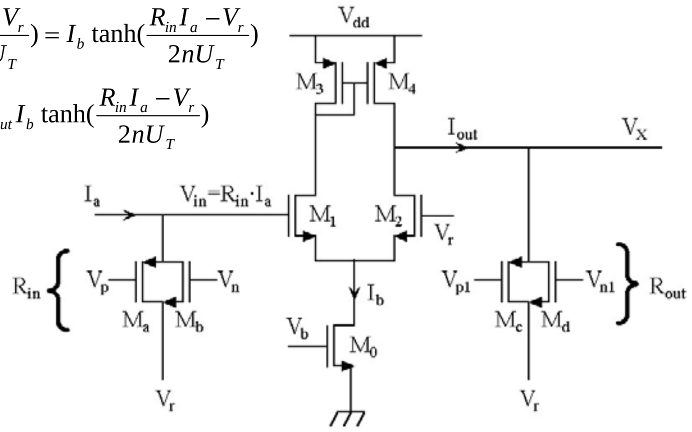
$$I_\eta^{\max}=I_s\cdot e^{\frac{V_H}{nU_T}}$$
$$I_\eta^{\min}=I_s\cdot e^{\frac{V_L}{nU_T}}$$

For instance, if (*V_H* - *V_L*)=0.3V, the ratio between the maximum and minimum values of the learning rate current can be one thousand.

The activation function A circuit

$$I_{out}=I_b\tanh\left(\frac{V_{in}-V_r}{2nU_T}\right)=I_b\tanh\left(\frac{R_{in}I_a-V_r}{2nU_T}\right)$$

$$V_X=R_{out}I_{out}=R_{out}I_b\tanh\left(\frac{R_{in}I_a-V_r}{2nU_T}\right)$$



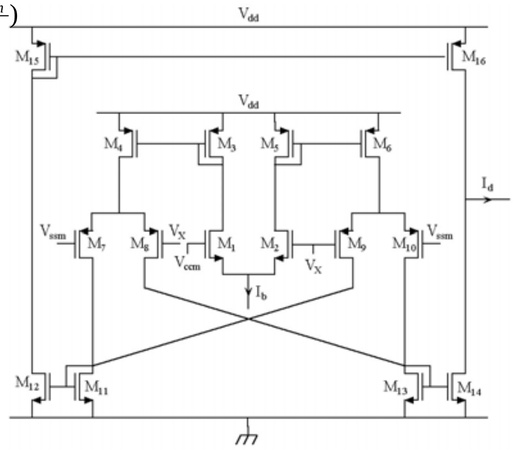
The derivative circuit D

$$I_d = I_b \tanh(\frac{V_{ccm}-V_X}{2nU_T})\tanh(\frac{V_X-V_{ssm}}{2nU_T})$$

$$I_d \cong \frac{I_b}{4(nU_T)^2}(V_{ccm}-V_X)(V_X-V_{ssm})$$

$$V_{ccm}-V_r = V_r-V_{ssm} = V_{norm}$$

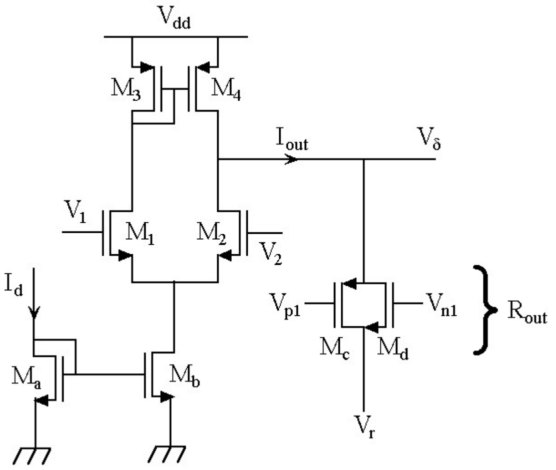
$$I_d \cong \frac{I_b V_{norm}}{4(nU_T)^2}\left(1-\left(\frac{V_X-V_r}{V_{norm}}\right)^2\right)$$



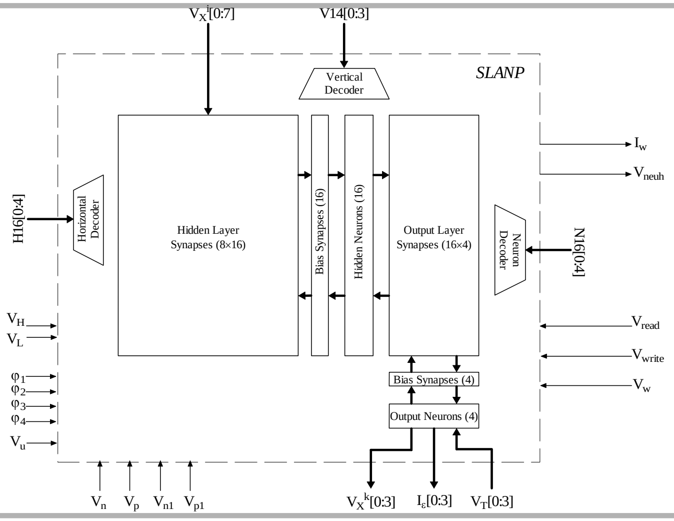
The error circuit R

$$V_\delta = R_{out} I_d \tanh(\frac{V_1-V_2}{2nU_T})$$

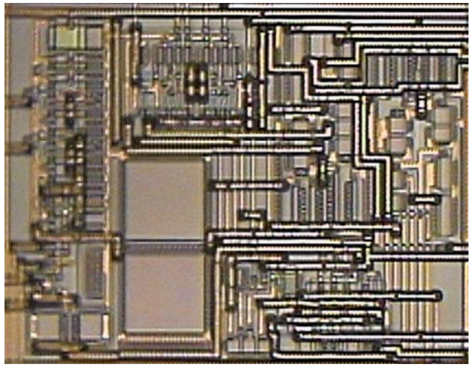
$$V_\delta \cong \frac{R_{out} I_d}{2nU_T}(V_1-V_2)$$

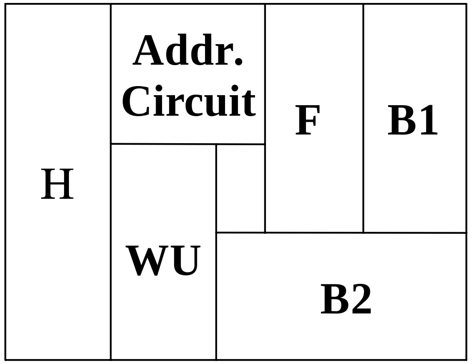


The SLANP chip (1)

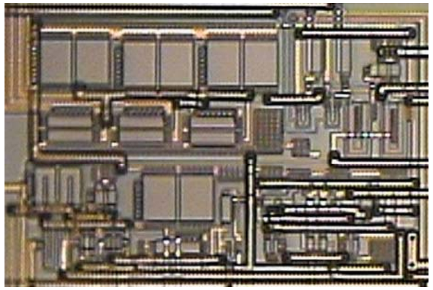


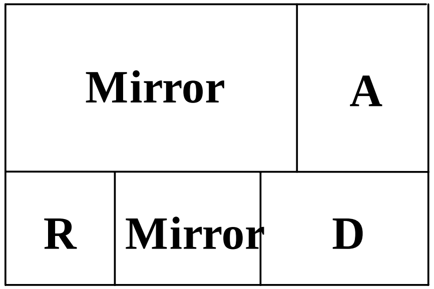
The SLANP chip (2) - the synaptic module



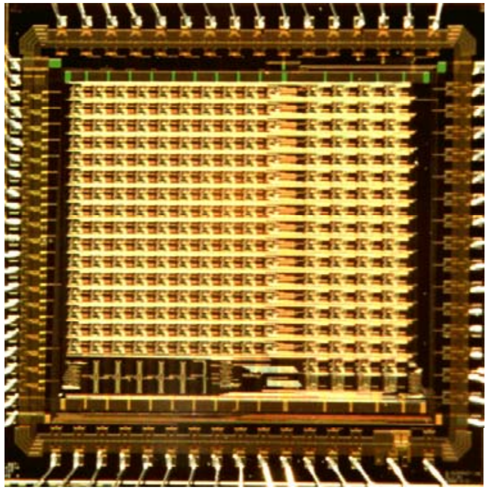


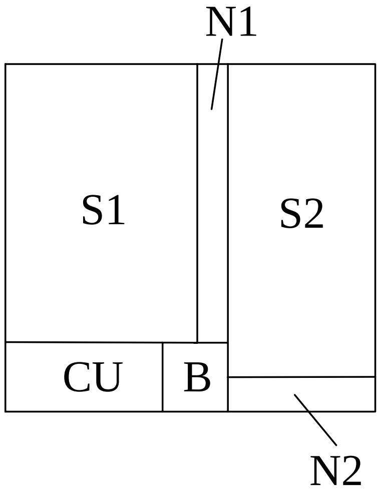
The SLANP chip (3) - the neuron module



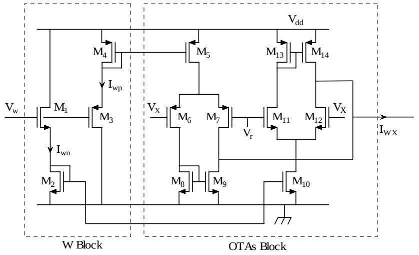


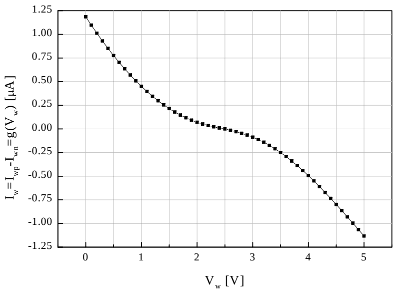
The SLANP chip (4)



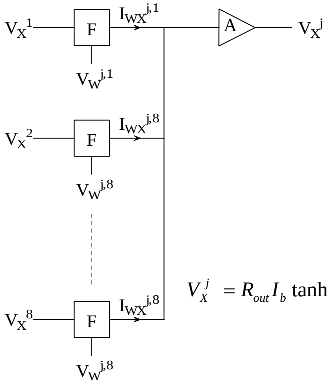


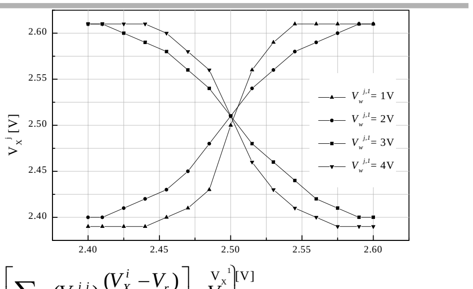
Experimental results (1)





Experimental results (2)

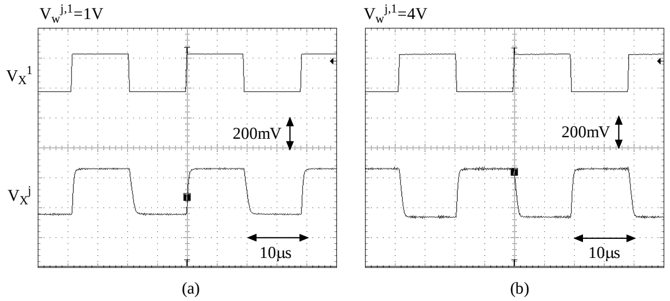




$$V_X^j = R_{out} I_b \tanh\left\{ \frac{R_{in}\left[\sum_i g(V_W^{j,i}) \frac{(V_X^i - V_r)}{2nU_T}\right] - V_r}{2nU_T} \right\}$$

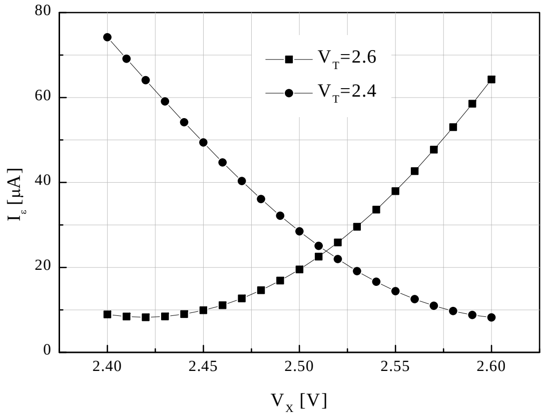
$$V_X^j = R_{out} I_b \tanh\left\{ \frac{R_{in} g(V_W^{j,1}) \frac{(V_X^1 - V_r)}{2nU_T} - V_r}{2nU_T} \right\}$$

Experimental results (3)

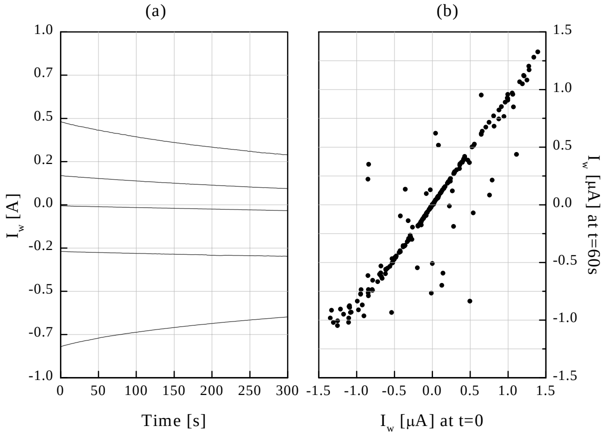


Transient response of the circuit for a positive (a) and negative (b) weight values (upper traces: synaptic input signals; bottom traces: neuron output signals).

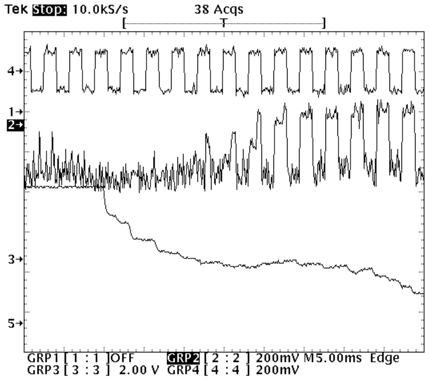
Experimental results (4) (circuit FC)



Experimental results (5) (Weight decay due to leakage currents on the weight capacitor C_w.)



Experimental results (6) - learning

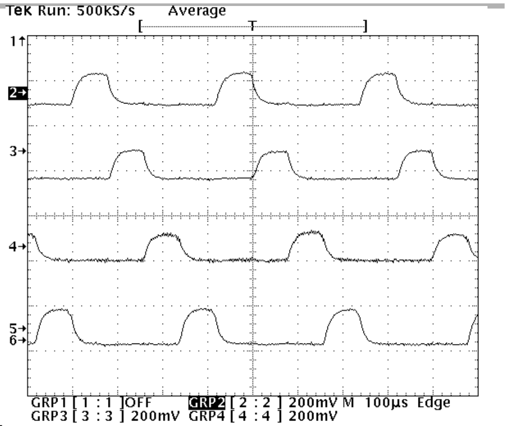


Training of the NOT function. Top trace – target signal; middle trace – output signal; bottom trace – a weight signal. The network were configured as 1×8×1 MLP and the learning rates were fixed to 0.5V. The learning iteration was 800μs.

Experimental results (7) - learning

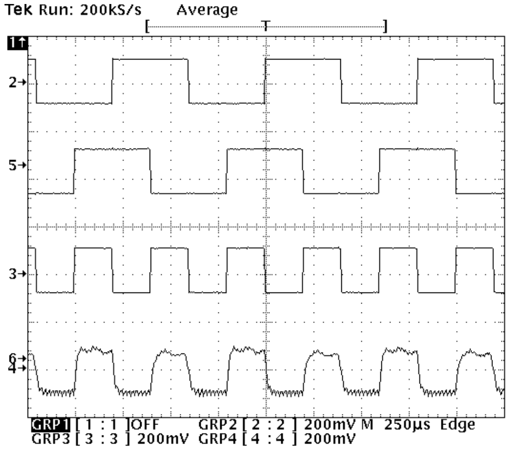
Training set for the CLASSIFICATION problem.

Input Pattern	Target
11001100	0001
10011001	0010
00110011	0100
01100110	1000



Four output neuron signals at the end of the training process for problem described by ?. The network were configured as 8×16×4 MLP and all the learning rates were locally adapted: the minimum and maximum learning rate values were 0.4 and 0.7V. The learning iteration was 80µs.

Experimental results (8) - learning



Training of the 2 input XOR function. The two input signals (first and second traces), the target signal (third wave) , and output signal (fourth wave) at the end of the training process. The network was configured as 2×2× 1 MLP and all the learning rates were locally adapted: the minimum and maximum learning rate values were 0.4 and 0.7V. The learning iteration was 200µs.

Performance

Network size	8×16×4 MLP
On-chip learning algorithm	by-pattern BP with local learning rate adaptation
Technology	ATMEL ES2 ECPD07
Transistor count	22000
Chip size	3.5mm×3.5mm
Power consumption	25mW
Recall computational power	106MCPS
Computational power	2.65MCUPS
Computational density	216000CUPS/mm^2
Energy efficiency	106000CUPS/mW