

Neurone numérique de modèle HOGKIN-HUXLEY, réglage des différents paramètres :

SOMMAIRE :

Modélisation du neurone.....	p.2
Matériel.....	p.3
Communication avec le neurone.....	p.4
Paramètres :	
Initialisation de la sigmoïde.....	p.5
Réglage des autres paramètres.....	p.6

Modélisation du neurone :

Le neurone a été implémenté selon le modèle d'HOGKIN-HUXLEY, et répond donc aux équations suivantes :

$$\frac{dV_{mem}}{dt} \cdot C = \sum (g_{ion}(V_{equi} - V_{mem})) \quad (1)$$

$$g_{ion} = g_{max} \cdot m^p \cdot h^q \quad (2)$$

$$\tau \cdot \frac{dn}{dt} = n_{\infty} - n \quad (3)$$

$$n_{\infty} = \frac{1}{1 + e^{\pm \frac{V_{mem} - V_{offset}}{V_{slope}}}} \quad (4)$$

Avec : V_{mem} : tension membranaire du neurone
 g_{ion} : conductance des différents canaux ioniques
 m, h : cinétiques des différents canaux ioniques
 τ : constante de temps des cinétiques

On prendra en compte : le canal de fuite (leak), les canaux ioniques sodium (Na), potassium (K) et calcium (Ca).

Pour les différents tests on rajoute une tension et un courant réglables (V_{clamp} et I_{stim}) ainsi qu'un interrupteur pour chaque canal.

Le neurone numérique simule donc le circuit suivant :

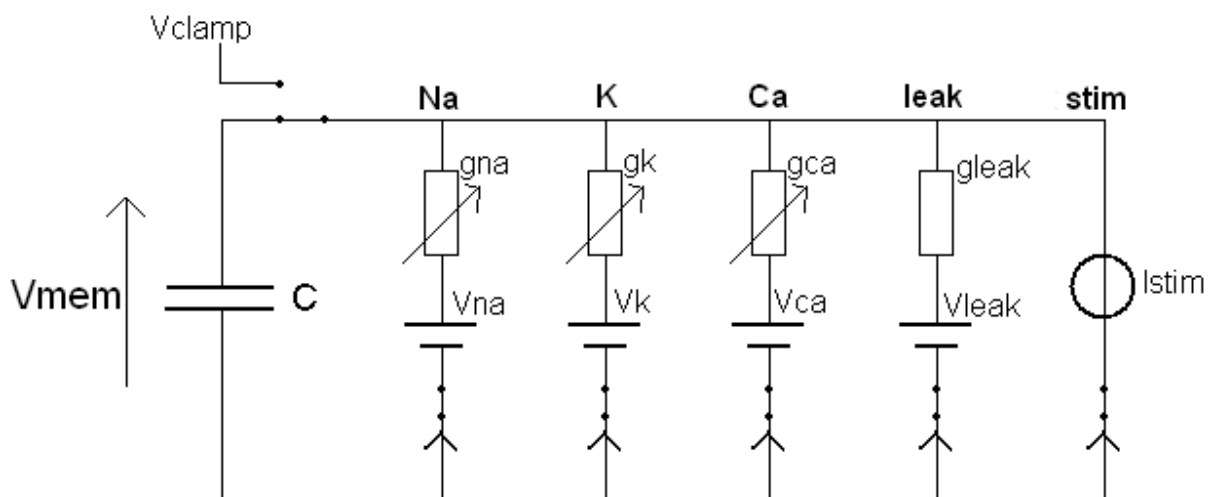
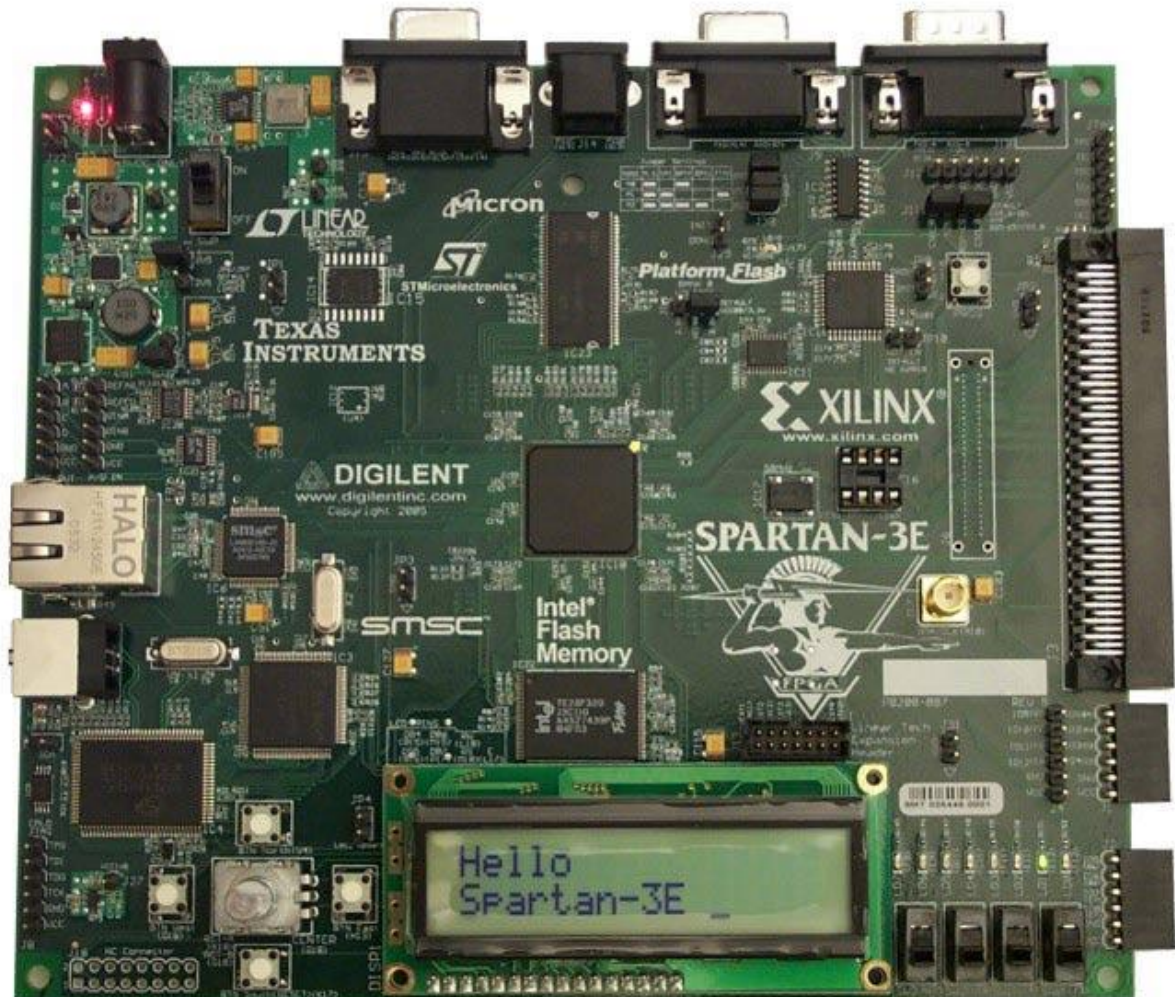


Schéma équivalent du neurone implémenté

Matériel :

On dispose de la *carte de développement Spartan 3E* de chez Xilinx (voir photo ci-dessous) et du logiciel *Xilinx ISE 9* pour programmer le *FPGA Spartan 3E XC3S500E package FG320* présent sur la carte.



Carte de développement Spartan 3E

Les périphériques présents sur la carte vont nous permettre de communiquer avec le FPGA et d'observer les signaux qui nous intéressent (cf. Communication avec le neurone).

Communication avec le neurone :

Pour changer les paramètres en direct, on implémente un picoblaze et on utilise les ports série de la carte spartan 3E.

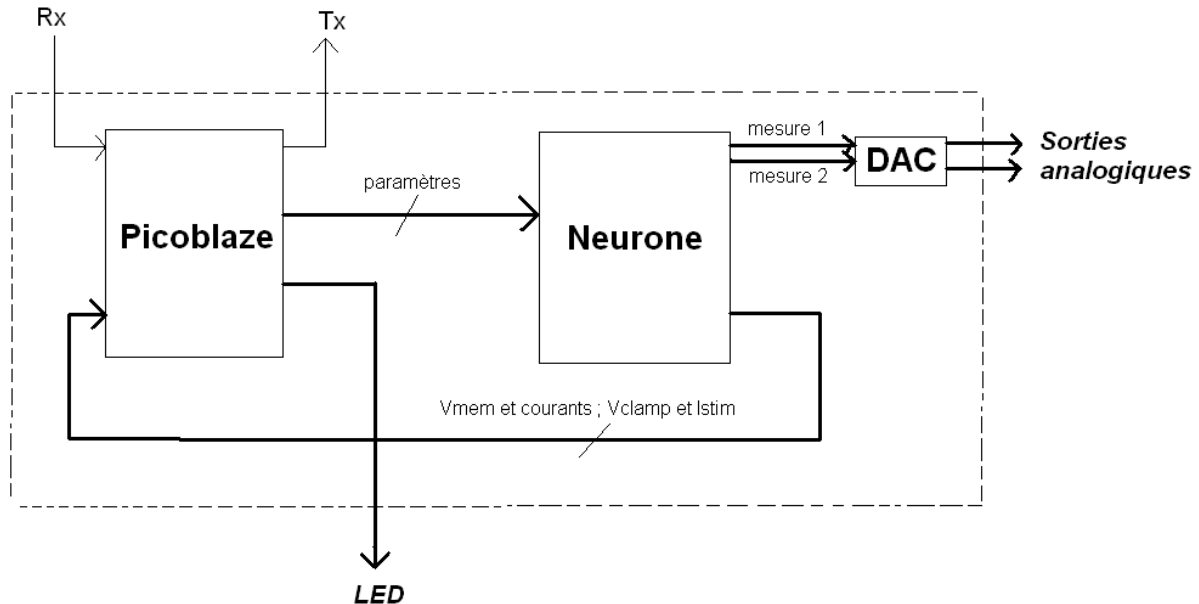


Schéma synoptique du neurone numérique

- Entrée :
 - Elle se fait via le port série (Rx) à une fréquence de 115k. Grâce à cette entrée, on peut exciter les LED, choisir les signaux observés en sortie et surtout modifier les paramètres de notre neurone.
- Sorties :
 - Les LED ne servent qu'à tester le bon fonctionnement du port série et du picoblaze.
 - La transmission (Tx) par le port série n'est pas encore gérée par le picoblaze, mais permettrait à terme de transmettre des valeurs à l'ordinateur.
 - L'observation se fait à l'aide de deux sorties analogiques de la carte Spartan 3E. Chaque sortie peut rendre compte en temps réel d'une des huit grandeurs suivantes : V_{mem} , V_{clamp} , i_{Na} , i_K , i_{Ca} , i_{leak} , $\sum i$

Paramètres :

Les équations (1), (2) et (3) sont fonctionnel au reset du FPGA, en revanche la forme de la sigmoïde de l'équation (4) doit être transmise et inscrite dans la RAM. Ensuite, il ne reste plus qu'à transmettre et initialiser les autres paramètres.

Initialisation de la sigmoïde :

Pour initialiser la sigmoïde on envoie 2048 valeurs sur 3bits allant de « 000 » à « 1FF » codant pour une demi sigmoïde tronquée, le reste sera construit par symétrie.

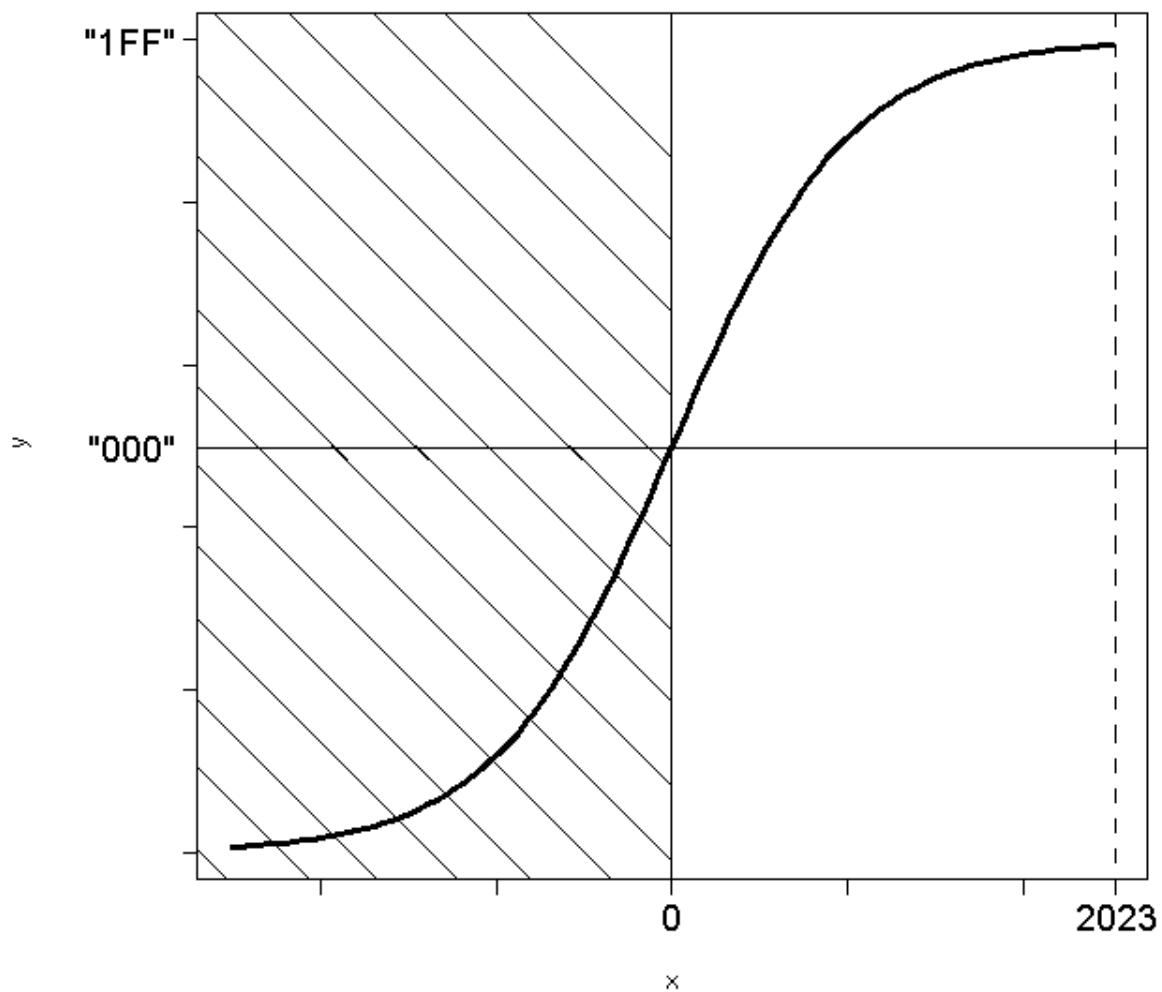


Schéma du codage de la sigmoïde

Pour ce faire, on envoie par la liaison série nos 2024 valeurs, à la suite, avec la syntaxe suivante :

R 000 R 001 R 002 R 003 R 003 etc.

Réglage des autres paramètres :

Rappelons que chacun de ces paramètres peut être changé un par un dans n'importe quel ordre.

- Tous les paramètres suivants s'initialisent avec la syntaxe suivant : **P « XX » « YY »**
Où « XX » code pour une adresse contenant les 8 bits de donnée « YY ».
- Le LSB sera toujours contenu par le 2^{ème} « Y » de l'adresse la plus faible de chaque paramètre.
Par exemple :

P 80 FD	le LSB est « D »
P 81 06	

Pour ce faire on transmettra les valeurs des paramètres suivants comme indiqué :

P 80 FD			
P 81 06 --delta t / c--	16 bits signés	avec	$C = \frac{T_{clock}}{0,9316 * \text{delta t} / c} F$
P 82 06			
P 83 00 --gmax Na--	16 bits signés	avec	$LSB = 0,5 nS$
P 84 06			
P 85 00 --gmax K--	16 bits signés	avec	$LSB = 0,5 nS$
P 86 06			
P 87 00 --gmax Ca--	16 bits signés	avec	$LSB = 0,5 nS$
P 88 2C			
P 89 00 --g leak--	16 bits signés	avec	$LSB = 0,5 nS$
P 8A 88			
P 8B 13 --V Na--	16 bits signés	avec	$LSB = 10 \mu V$
P 8C F0			
P 8D D8 --V K--	16 bits signés	avec	$LSB = 10 \mu V$
P 8E F0			
P 8F D8 --V Ca--	16 bits signés	avec	$LSB = 10 \mu V$
P 90 A8			
P 91 E4 --V leak--	16 bits signés	avec	$LSB = 10 \mu V$
P 92 10 --p,q Na--	2 X 4bits signés : « p & q »		
P 93 10 --p,q K--	2 X 4bits signés : « p & q »		
P 94 10 --p,q Ca--	2 X 4bits signés : « p & q »		
P 95 00			
P 96 00 --Voffset m_Na--	16 bits signés	avec	$LSB = 10 \mu V$
P 97 00			
P 98 00 --Voffset h_Na--	16 bits signés	avec	$LSB = 10 \mu V$

P 99 00		
P 9A 00 --Voffset n_K--	16 bits signés avec	$LSB = 10 \mu V$
P 9B 00		
P 9C 00 --Voffset m_Ca--	16 bits signés avec	$LSB = 10 \mu V$
P 9D 00		
P 9E 00 --Voffset h_Ca--	16 bits signés avec	$LSB = 10 \mu V$
P 9F FF		
P A0 FF --Vslope m_Na--	16 bits signés avec	$LSB = 10 \mu V$
P A1 FF		
P A2 FF --Vslope h_Na--	16 bits signés avec	$LSB = 10 \mu V$
P A3 FF		
P A4 FF --Vslope n_K--	16 bits signés avec	$LSB = 10 \mu V$
P A5 FF		
P A6 FF --Vslope m_Ca--	16 bits signés avec	$LSB = 10 \mu V$
P A7 FF		
P A8 FF --Vslope h_Ca--	16 bits signés avec	$LSB = 10 \mu V$
P A9 09		
P AA 00		
P AB 00 --accumax m_Na--	"0000" & 20 bits signés avec	$\tau = \frac{T_{clock}}{4 \cdot 2^{-9}} (3 + accumax) \quad ms$
P AC 5F		
P AD 00		
P AE 00 --accumax h_Na--	"0000" & 20 bits signés avec	$\tau = \frac{T_{clock}}{4 \cdot 2^{-9}} (3 + accumax) \quad ms$
P AF 91		
P B0 04		
P B1 00 --accumax n_K--	"0000" & 20 bits signés avec	$\tau = \frac{T_{clock}}{4 \cdot 2^{-9}} (3 + accumax) \quad ms$
P B2 32		
P B3 0C		
P B4 00 --accumax m_Ca--	"0000" & 20 bits signés avec	$\tau = \frac{T_{clock}}{4 \cdot 2^{-9}} (3 + accumax) \quad ms$
P B5 32		
P B6 0C		
P B7 00 --accumax h_Ca--	"0000" & 20 bits signés avec	$\tau = \frac{T_{clock}}{4 \cdot 2^{-9}} (3 + accumax) \quad ms$
P B8 3B --interrupteurs--	"00" & clamp & Na & K & Ca & leak & stim	
	Avec : $clamp = '1' \Leftrightarrow V \leq V_{clamp}$ Na,K,Ca,leak ou stim = '1' \Leftrightarrow interrupteur ouvert	

P B9 00

P BA 00 --Vclamp--

16 bits signés avec

$$LSB = 10 \mu V$$

P BB 00

P BC 00

P BD 00

P BE 00 --Istim--

“00” & 16bits & 14 bits de precision

Avec $LSB = 2^{14} * LSB_{precision} = 81,9 \text{ pA}$

P BF 14 --mux mesure 1 et 2-- ‘0’ & 3bits & ‘0’ & 3bits

Avec : « 0 & mux_mesure1 & 0 & mux_mesure 2 »

mux_mesure 1 ou 2 =

- « 000 » ⇔ sortie analogique <= Vmem
- « 001 » ⇔ sortie analogique <= Vclamp
- « 010 » ⇔ sortie analogique <= i_{Na}
- « 011 » ⇔ sortie analogique <= i_K
- « 100 » ⇔ sortie analogique <= i_{Ca}
- « 101 » ⇔ sortie analogique <= i_{leak}
- « 110 » ⇔ sortie analogique <= i_{stim}
- « 111 » ⇔ sortie analogique <= $\sum i$

Remarque 1: Le nombre de bits géré par le DAC est de **12bits**, il faudra donc s’attendre à des effets de bord.

Le choix des bits transmis par le FPGA au DAC sont facilement modifiables dans le code VHDL, les LSB’s seront donc différents selon que l’on choisisse de regarder un signal de faible ou de grande amplitude.

Remarque 2: Le bloc VHDL correspondant à l’équation (4) ne fonctionne actuellement pas. Ceci serait dû à un problème dans la génération d’adresse d’accès à la RAM contenant la sigmoïde (à vérifier).

Pour d’autres questions : « madec@enseirb.fr »