— 修士論文 —

ＦＰＧＡスパイキング神経ネットワーク間の通信プロトコル

平成28年2月2日　提出

指導教員　　河野 崇　准教授

東京大学大学院工学系研究科
電気系工学専攻

学籍番号：37-126513
氏名　　權 泰五

概 要

スパイキング神経ネットワークとはニューロンの電気活動のメカニズムを工学的に再現し、それらをネットワーク化したものである。大規模のスパイキング神経ネットワークが構築できれば脳のシミュレーションが期待できる。しかし、リソースの制限上、単一チップに構築できるニューロン数やそれらの接続数に制約ができてしまう。従って、大規模スパイキング神経ネットワークの構築には複数のチップからなるネットワークが不可欠である。

河野研ではスパイキング神経ネットワークの実装に FPGA を用い、さらに複数の FPGA からなる大規模のスパイキング神経ネットワークのコンセプトが作られていた。そのコンセプトの最下位レベルでは、1 つの FPGA が 1024 ニューロンの全結合ネットワークを持ち、128 個の FPGA がリング状に繋がっていることを想定している。また、通信方法としては Address Event Representation(AER) を用い、一方の FPGA で起きたニューロンイベントの情報送信の代わりにアドレス送信を行うことで通信部を簡潔化している。

本研究では、上記のコンセプトを具体化し FPGA スパイキング神経ネットワーク間を結ぶ第一段階の通信プロトコルを構築した。スパイキング神経ネットワークのニューロンモデルとしては、ニューロンの多様なダイナミクスが再現できる Digital Spiking Silicon Neuron(DSSN) モデルを用いた。ネットワークの規模は 256 ニューロンと 256^2 シナプスの全結合ネットワークである。

通信プロトコルの動作を確かめるため、通信プロトコルを通じて接続された 2 つのスパイキング神経ネットワーク (512 ニューロンの全結合ネットワーク) を用いて連想記憶を行った。連想記憶としては、相関学習による連想記憶を行い正しい結果が確認された。また、ヘブ則による連想記憶も試みた。

今後は 1024 ニューロンの全結合ネットワークに向けた通信プロトコルの構築と、2 つ以上の FPGA を繋げるための機能を拡張した通信プロトコルを構築したい。

目 次

第1章　序論

1.1　ニューロンとシナプス

神経系は外部からの刺激を統合し再び神経系内に刺激を与える高機能でかつ複雑なシステムである。神経系内で起きる信号処理は、従来のデジタルコンピュータに比べロバストでかつ柔軟性と自律性を持つ。例として神経系は周辺のノイズによる歪んだ入力から正しい反応を示すことが可能である。また、物理的なダメージによる機能の損失も自律的に補正可能である。

ニューロンとは電気や化学的な入力に対して自分自身が再び電気信号を生成し、それを他のニューロンに伝達する神経細胞である。各ニューロンは樹状突起と軸索、細胞体から構成される (図 1.1)。樹状突起で他のニューロンから刺激電流を受け取り、細胞体で刺激電流を統合して新しい刺激電流生成の判定を行う。新しく生成された刺激は軸索を通して他のニューロンに伝達される。軸索の先端と入力を受け取る他のニューロンの樹状突起の結合部分の間をシナプスと呼ぶ。ニューロンはお互い密に繋がっている。例えば皮質錐体のニューロンは 1000 樹状突起を持ち、軸索は平均的に 7200 シナプス結合をしている。

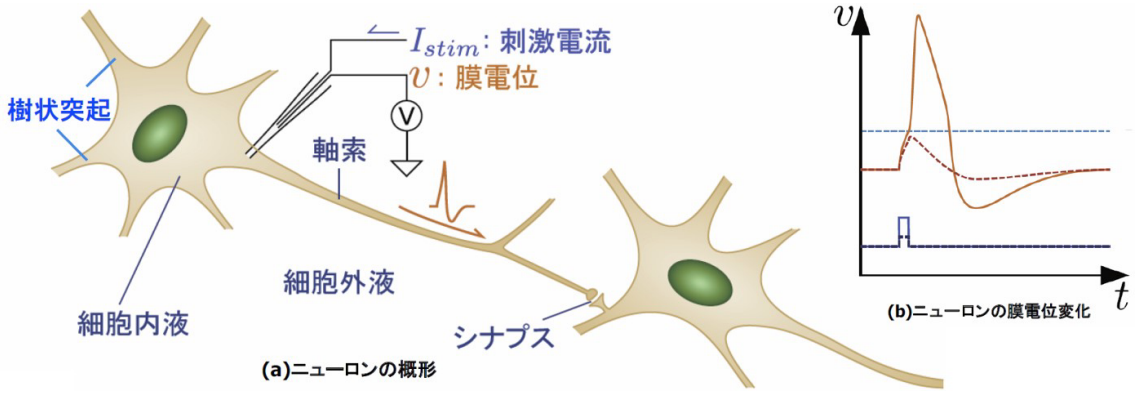

図 1.1　ニューロンと膜電位の模式図 [1] より引用

図 1.2 はニューロン細胞膜の模式図である。細胞内液と細胞外液は細胞膜によって仕切られて

おり、細胞膜は絶縁性の高い二重の脂質層である。細胞内液、細胞外液には多様なイオンが存在し、Na^+、K^+、Cl^- はイオンチャネルを通して細胞膜の内外を行き来することができる。細胞内と細胞外のイオンの濃度差によって電位差が生じ、細胞外を基準としたときの細胞内の電位を膜電位と呼ぶ。膜電位が負であり安定な状態であるとき、ニューロンは静止状態にあると呼び、この時の膜電位を静止膜電位と呼ぶ。膜電位はイオン電流の流入・流出によって増減される。膜電位が静止膜電位より低いときを過分極と呼ぶ。あるニューロンには電流の注入のような正の刺激に対して膜電位の急激な上昇がみられる（図 1.1(b)）。この膜電位の急激な上昇をスパイクと呼び、ニューロンがスパイクを生成したことを発火と呼ぶ。スパイクの生成は膜電位がある閾値を超えたときのみ行われると知られている。

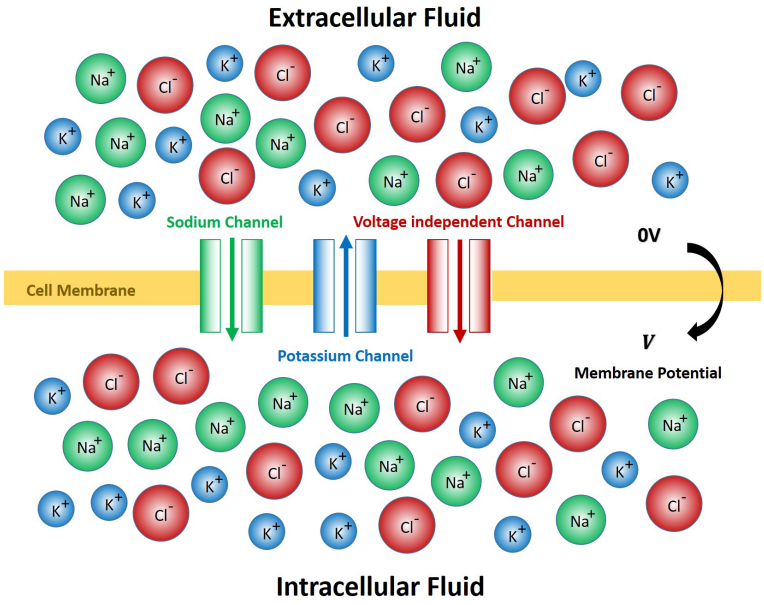

図 1.2　ニューロン細胞膜の模式図

ニューロンが発火をすると軸索を通して電気信号が送られシナプスを通して他のニューロンへ信号が伝わる。電流が前シナプスニューロンから後シナプスニューロンへ直接伝わるのではなく、図 1.3 のシナプスを介して伝達される。シナプスとは、軸索の先端と樹状突起との間にある 20nm 程の隙間である。電気信号が前シナプスニューロンから送られ軸索の先端に達すると、軸索の先端のシナプスボタンという膨らんだ部分から神経伝達物質が放出される。神経伝達物質は後シナプスニューロンの持つ受容体と結合しイオンチャネルが開く。イオンが細胞膜を通過することによってシナプス電流が細胞内に流れはじめ、膜電位が上昇する。膜電位の上昇が閾値を超えるとニューロンは活動電位を発生する。シナプスにはシナプス可塑性 (STDP) という性質があり、それは前シナプスニューロンと後シナプスニューロンのスパイク発生時間差によって結合強度が変

化する現象である。シナプス可塑性は神経ネットワークの学習において重要な役割を担っていると知られている。

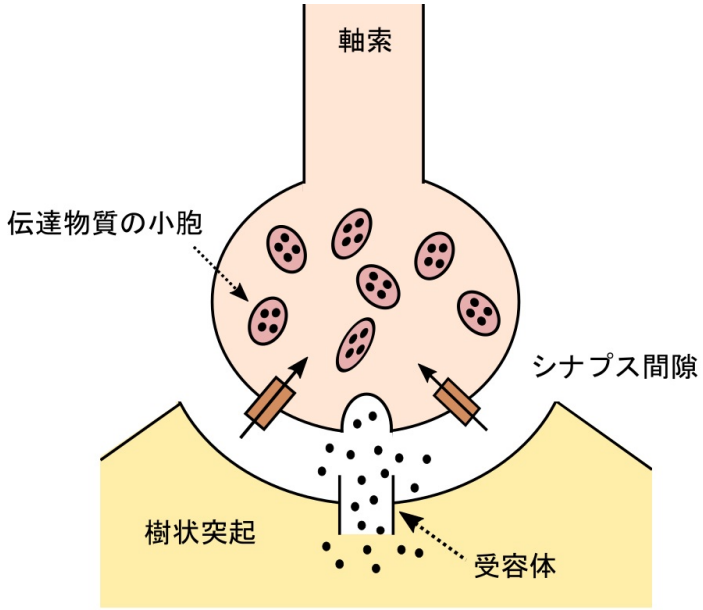

図 1.3　シナプスの模式図

1.2　ニューロンモデルとシナプスモデル

1.2 節ではニューロンモデルとしてイオンコンダクタンスモデル、現象論的モデル、定性的モデルを紹介する。シナプスモデルとしては動的モデルを紹介する。

1.2.1　イオンコンダクタンスモデル

イオンコンダクタンスモデルとはイオンチャネルを通じて移動するイオンのダイナミクスを記述したモデルである。Hodgkin-Huxley (H-H) モデルはニューロンのイオンのダイナミクスを記述する最も有名なモデルである。H-H モデルはヤリイカの軸索から得られた結果であり、興奮性刺激に対するスパイクの発生を次の 5 ステップによって説明する。まず 1 ステップとして、ほとんどの Na+チャネルが閉ざされた状態であり、膜電位は約 65mV を保つ。2 ステップとして、入力の刺激によって膜電位が急増し、Na+チャネルの開放率が増加する。3 ステップとして、Na^+ イオンが細胞内に流れはじめ、さらなる膜電位の増加を引き起こす。これによって Na^+ チャネルの開放率はさらに増え続けられる。4 ステップとして、膜電位が閾値を超えたら、この正のフィード

バックが繰り返される。膜電位が十分高い時、Na^+ チャネルは閉ざされ始まる一方で、K^+ チャネルの開放率は増加し続ける。膜電位はこの二つの要素によって減衰する。5 ステップとして、Na^+ チャネルの開放率が減衰し、続いて K^+ チャネルの開放率も減衰し、最終的に膜電位が静止状態に至る。この一連の 5 ステップを H-H モデルは次の 4 変数の常微分方程式を用いて表す。

$$C\frac{dv}{dt} = \overline{g}_{Na}m^3h(E_{Na} - v) + \overline{g}_K n^4(E_K - v) + g_L(E_L - v) + I_{stim} \tag{1.1}$$

$$\frac{dm}{dt} = \frac{0.1(v+40)}{1 - e^{\left(-\frac{v+40}{10}\right)}}(1-m) - 4me^{\left(-\frac{v+65}{18}\right)} \tag{1.2}$$

$$\frac{dh}{dt} = 0.07e^{\left(-\frac{v+65}{20}\right)}(1-h) - \frac{1}{1+e^{\left(-\frac{v+35}{10}\right)}}h \tag{1.3}$$

$$\frac{dn}{dt} = \frac{0.01(v+55)}{1 - e^{\left(-\frac{v+55}{10}\right)}}(1-n) - 0.125e^{\left(-\frac{v+65}{80}\right)}n \tag{1.4}$$

C は細胞膜のキャパシタンス (1uF/cm^2)、 v は膜電位、I_{stim} は刺激電流である。パラメータ E_{Na}、E_K, E_L は Na^+、K^+、漏れ電流の平衡電位 (それぞれ 50mV, −77mV, −54.4mV) である 。定数 $\overline{g}_{Na}$、$\overline{g}_K$ は Na^+、K^+ の最大のコンダクタンス (それぞれ 120 mS/cm^2, 36mS/cm^2) である。定数 g_L は漏れ電流のキャパシタンス (0.3mS/cm^2) である。変数 m、 h 、 n はチャネルゲートの開放率を表す。Na^+ チャネルは 3 つの活性ゲート (m) と 1 つの非活性ゲート (h) を持つ。K^+ チャネルは 4 つの活性ゲートを持つ (n)。。

1.2.2 現象論的モデル

イオンコンダクタンスモデルはニューロンの動作を正確に再現することができる一方で、それを表す方程式が複雑である。もう一つのニューロンモデルとしてスパイクの生成だけに注目した簡単なモデルがある。Integrateand-Fire(IF) モデルは膜電位にリセットをかけることによってスパイク生成を簡略化したモデルである。後日、静止状態を表すため漏れ電流を表す項が加われ、Leaky Integrateand-Fire(LIF) モデルが考案された。次の式が LIF モデルの式である。

$$C\frac{dv}{dt} = -\frac{v}{R} + I_{stim}, \qquad v \to v_0 \quad when \quad v > \theta \tag{1.5}$$

C は細胞膜のキャパシタンス、v は膜電位を表す。パラメータ R、v_0, τ_{ref}, θ は漏れキャパシタンスの逆数、静止膜電位、不応期、閾値を表す。膜電位 v が閾値 θ に達したとき、τ_{ref} 間に膜電位が v_0 にリセットされる。このとき後シナプスニューロンにスパイクが伝達される。LIF モデルはニューラルスパイク特性を示したいとき効果的である。

1.2.3 定性的モデル

もう一つのニューラルモデルとして定性的モデルがある。定性的モデルはニューロンの活動を幾何学構造や分岐構造の観点からを説明する。その代表的なモデルとして Izhikevich(IZH) モデルがあり、次の 2 変数微分方程式で表現される。

$$\frac{dV}{dt} = 0.04V^2 + 5V + 150 - U + I_{stim}, \quad V \leftarrow c \quad when \quad V \geq 30mV \tag{1.6}$$

$$\frac{dU}{dt} = -a(bV - U), \quad U \leftarrow U + d \quad when \quad V \geq 30mV \tag{1.7}$$

V は膜電位、U は状態変数である。IZH モデルは LIF モデルに比べニューロンの多様なダイナミクスを再現することができる。このモデルは LIF モデルと同様に状態変数にリセットをかけ、ニューロンの発火をイベント化している。

1.2.4 シナプスモデル

シナプスの動的モデルは、イオンチャネルの活性化と化学伝達物質の濃度の観点から伝達物質の放出のプロセスと後シナプス電流生成の両方を表現する [2]。伝達物質は前シナプス電位の上昇につれて放出される。このプロセスの最も簡単なモデルは次のようである。

$$[T](v_{pre}) = \frac{T_{max}}{1 + exp\left(-\frac{v_{pre} - v_p}{K_p}\right)} \tag{1.8}$$

$[T]$ はシナプス間隙の化学伝達物質の濃度、v_{pre} は全シナプスニューロンの膜電位、T_{max} は $[T]$ の最大値を表す。パラメーター v_p はこの関数値が半分になるときの値であり、K_p は傾きを表す。この式では全体のプロセスが十分速くかつ前シナプス電位から伝達物質への転換が持続的だと仮定することで伝達物質の放出を近似する。後シナプス電流の生成は後シナプスニューロンの受容

器の活性化によって起きる。このプロセスの最もシンプルな過程は次のようである。

$$\frac{dr}{dt} = \alpha[T](1-r) + \beta r \tag{1.9}$$

$$I_{syn} = \overline{g}_{syn} r(v - E_{syn}) \tag{1.10}$$

r は開状態の受容体の割合を表す。定数 α と β は受容体の開から閉への遷移の割合とその逆の割合を表す。後シナプス電流 I_{syn} は $\overline{g}_{syn}$、E_{syn} 、 v から計算される。$\overline{g}_{syn}$、 E_{syn} は最大コンダクタンス、逆ポテンシャルを示す定数である。変数 v は後シナプス膜電位である。

1.3 シナプス学習モデル

シナプス可塑性はニューロン間の接続強さの変化を意味する。シナプスの伝達物質の放出量や受容体の量の変化はシナプス可塑性の根底にあると知られている。シナプス可塑性は短期可塑性と長期可塑性 2 クラスに分けられる。短期可塑性は、せいぜい何分程度持続され、シナプスの接続を強化することも弱化することも可能である。生物学の研究では短期可塑性は主に前シナプス活動電位によるシナプス伝達物質の放出確率の変化によって起き、外部刺激の周波数に依存すると証明された。長期シナプス可塑性は海馬や大脳皮質で観測される。長期増強（LTP）と長期抑圧（LTD）も刺激の周波数に依存すると知られている。1949 年に提案されたヘブ則は、ニューロン A からの入力がニューロン B の発火を起こしたと見なすときニューロン A から B へのシナプス強度は強化されると仮定した。この仮定は、海馬で起こるシナプス伝達の増強の観測によって証明された。次のヘブ則の方程式は [3] のスパイキングニューロンネットワークで紹介された。

$$\Delta W = A_+ exp\left(\frac{-|\Delta T|}{\tau_+}\right) \tag{1.11}$$

ΔW はシナプス重み W の変化量、ΔT は前シナプススパイクと後シナプススパイクの時間差を表す。パラメータ A_+ は振幅、τ_+ は時定数である。ただし、上式は LTP しか表していない。LTP に加え LTD も学習に重要な役割を果たすと考えられていたため、LTD が加わったのが次の式である。パラメータ A_- は振幅、τ_- は時定数である。

$$\Delta W = A_+ exp\left(\frac{-|\Delta T|}{\tau_+}\right) - A_- exp\left(\frac{-|\Delta T|}{\tau_-}\right) \tag{1.12}$$

1.4 スパイキング神経ネットワークの先行研究

シリコンニューラルネットワークはリアルタイムで神経系の活動を再現するために考案されたものである。[4] では、LIF モデルに基づいたシリコンニューラルネットワークが実装された。単一チップに 100 万ニューロンおよび各ニューロンあたり 256 シナプスを実装した規模であり 63mW の低電力を実現できた。LIF モデルの簡単な構造が大規模のデジタルシリコンニューラルネットワークを可能にするが、ニューロンのクラスⅡの特性を再現できない短所がある。IZH モデルを用いたデジタル・シリコンニューロンは [5] で報告された。[5] では単一 FPGA に 1024 ニューロンの全結合ネットワークを実装できた。IZH モデルをデジタル回路として実装するにおいて、ニューロンモデルの方程式は浮動小数点として計算された。ニューロンの多様なダイナミクスをコンパクトかつシンプルな回路でシミュレーションするためには、ニューロンモデルの選択が重要である。IZH モデルは、他モデルに比べ、非線形性がたった 2 次である上で少ない乗算器で実装ができるため、スパイキングニューラルネットワークに向いているモデルだと考えられている。しかし、IZH モデルはクラスⅡニューロンのパルス刺激に対する段階的反応の再現に不十分である。これは、IZH モデルがスパイク生成のプロセスで状態変数にリセットをかけてモデルを近似しているからである。この近似によって多様な刺激に対して似たようなスパイクを生成してしまう。例えば、膜電位の最大値がすべてのスパイクで一定であることが挙げられる。

定量的シリコンニューラルモデルとして Digital Spiking Silicon Neuron（DSSN）モデルが提案された [6]。このモデルはコンパクトなデジタル演算回路を用いてニューロンのいくつかのクラスをシミュレーションすることが可能である。DSSN モデルはハードウェアのリソースを削減する固定小数点の演算を用いてニューロンの複雑な行動を再現できることが示された。このモデルは、IZH モデルとは違い状態変数にリセットをかけないため、クラスⅡニューロンの段階的応答を効果的に再現することができた。

生体の神経系と同様な自動学習メカニズムを持つ人工システムを実現するため、ヘブ則のような時間依存性学習則を備えたシリコンニューラルネットワークが研究された。このような学習則を備えた上にニューロンモデルとして LIF を用いたシリコンニューラルネットワークが [7] で報告された。[7] ではデジタル回路としてニューラルネットワークが実装され連想記憶が試された。

より大規模なスパイキング神経ネットワークをシミュレーションするため、ニューラルネットワーク計算を行う ARM プロセッサを複数接続する通信プロトコルが [8] で研究された。[9] では、FPGA に実装したスパイキング神経ネットワークをつなぐための通信プロトコルが研究された。

1.5　論文の構成

第１章では序論を述べた。第２章では、本研究で用いるニューロンモデルとシナプスモデル、さらにそれらの接続について述べる。第３章では、大規模スパイキング神経ネットワーク構築のコンセプトについて述べ、第４章ではそのコンセプトに従うスパイキング神経ネットワークの構造と、ネットワーク間を結ぶ通信プロトコルの構造について述べる。第５章では、通信プロトコルの動作確認のため、通信プロトコル介して接続したスパイキング神経ネットワークの連想記憶の動作について述べる。第６章では、本論文のまとめを行う。

第2章　スパイキング神経ネットワークのモデル

第 2 章では本研究でシリコンニューロンモデルとして用いる DSSN モデルと、シナプスモデルとして用いる簡略化された動的モデルについて述べ、さらにシリコンニューロンとシリコンシナプス間の接続についても述べる。

2.1　シリコンニューロンモデル

本研究で用いたニューロンモデルは DSSN モデル [6] である。DSSN モデルはニューロンのダイナミクスを非線形力学の観点から再現した定量的モデルである。デジタル演算回路向けに考案されたこのモデルは以下の 2 変数微分方程式で表現される。

$$\frac{dv}{dt} = \frac{\varphi}{\tau}(f(v) - n + I_0 + I_{stim}) \tag{2.1}$$

$$\frac{dn}{dt} = \frac{1}{\tau}(g(v) - n) \tag{2.2}$$

$$f(v) = \begin{cases} a_n(v + b_n)^2 - c_n & when \quad (v < 0) \\ -a_p(v - b_p)^2 + c_p & when \quad (v \geq 0) \end{cases} \tag{2.3}$$

$$g(v) = \begin{cases} k_n(v - p_n)^2 + q_n & when \quad (v < r) \\ k_p(v - p_p)^2 + q_p & when \quad (v \geq r) \end{cases} \tag{2.4}$$

v は膜電位、n はイオンチャネルの活動性を表す変数である。. 式 (2.1) の I_0 はバイアス電流、I_{stim} は重みのかけられたシナプス電流の総和である。パラメータ φ と τ は時定数である。パラメータ r、a_x、b_x、c_x、k_x、p_x、q_x $(x = n, p)$ は変数のナルクラインをコントロールする定数である。本研究で設定されたパラメータ値は表 2.1 に示す。パラメータ値は [11] を参考にした。[10] によってニューロンの生成する膜電位のスパイクの幅は同じでないことが確認された。スパイクの幅も情報伝達に重要な意味を持つと考えられ、生成されるスパイクの形が毎回同じではなく、スパイ

クの幅の変化を表せるように考案されたのが DSSN モデルである。LIF モデルでは、発火が起きた後に変数にリセットをかけるようにしてニューロンの発火をイベントとして扱っている一方で、DSSN モデルではリセットをかけず連続的に変数を変化させることによってスパイクの幅の多様性を実現できた。

表 2.1　DSSN モデルのパラメータ

Par.	Value	Par	Value
a_n	8.0	a_p	8.0
b_n	0.25	b_p	0.25
c_n	0.5	c_p	0.5
k_n	4.0	k_p	16.0
p_n	$-2^{-1}-2^{-4}$	p_p	$2^{-5}-2^{-2}$
q_n	-1.317708517	q_p	-0.6875
φ	0.5	τ	0.003
r	-0.104166	I_0	-0.23

2.2　シリコンシナプスモデル

本研究で用いたシリコンシナプスモデルは、序論で述べた動的モデルに基づいているが、デジタル演算回路に向けて効率的な計算や実装を考慮し、さらに簡略化されたモデルを用いた [11]。その簡略化について以下で説明する。まず、式 (1.8) で表現された伝達過程を簡略化するため、伝達物質の濃度が最大値 1、最小値 0 の方形パルスの形で変化すると仮定する。図 2.1 は簡略化前後の伝達物質濃度の変化を示したグラフである。本研究で用いられる簡略化されたモデルでは、膜電位が閾値 (本研究ではゼロ) を超えた時点で [T] が 1 になり、閾値を下回るときに [T] が 0 になる。

次に後シナプス電流の生成プロセスも簡略化する。式 (1.9), (1.10) の受容体の割合 r と後シナプス電流 I_{syn} を組み合わせて新しい後シナプス電流 $\tilde{I}_s$ を定義する。$\tilde{I}_s$ の定義式は次のようである。

$$\frac{d\tilde{I}_s}{dt} = \tilde{\alpha}[T](1-\tilde{I}_s) - \tilde{\beta}\tilde{I}_s \tag{2.5}$$

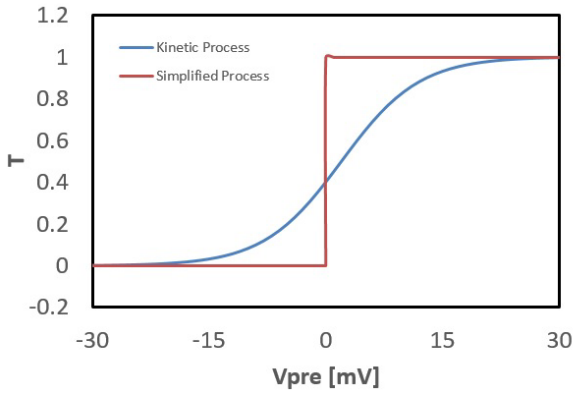

図 2.1　伝達物質の放出プロセス

$\tilde{\alpha}$ と $\tilde{\beta}$ は、受容体の開状態から閉状態への遷移割合とその逆の割合を表す。$[T]$ が方形パルスであり、$I_s = \frac{\tilde{\alpha}+\tilde{\beta}}{\tilde{\alpha}}\tilde{I}_s$ と新しく定義すると次の式が得られる。

$$\frac{dI_s}{dt} = \begin{cases} (\tilde{\alpha}+\tilde{\beta})(1-I_s) & when \quad [T]=1 \\ -\tilde{\beta}I_s & when \quad [T]=0 \end{cases} \tag{2.6}$$

ここで、さらに $\alpha = \tilde{\alpha}+\tilde{\beta}$、$\beta = \tilde{\beta}$ と定義すると、

$$\frac{dI_s}{dt} = \begin{cases} \alpha(1-I_s) & when \quad [T]=1 \\ -\beta I_s & when \quad [T]=0 \end{cases} \tag{2.7}$$

となる。I_s を $\frac{\tilde{\alpha}+\tilde{\beta}}{\tilde{\alpha}}$ とスケーリングした効果は、後から出てくる式 (2.8) の係数 c によって相殺される。図 2.2 は簡略化したシリコンシナプスモデルに従って生成されたシナプス電流の数値シミュレーションの結果である ($\alpha = 83.3$、$\beta = 333.3$) [11]。刺激電流 I_{stim} の値は、最初の 18.75ms 間は 0.04 であり、18.75 〜 37.5ms までは 0.06、37.5ms 以降は 0.08 となる。シミュレーションの結果から、シナプス電流の指数関数的な増加と減衰は伝達物質の放出時間によって決まることが分かる。つまり、[T]=1 の間はシナプス電流が増加し、[T]=0 の間は減衰しゼロに達する。

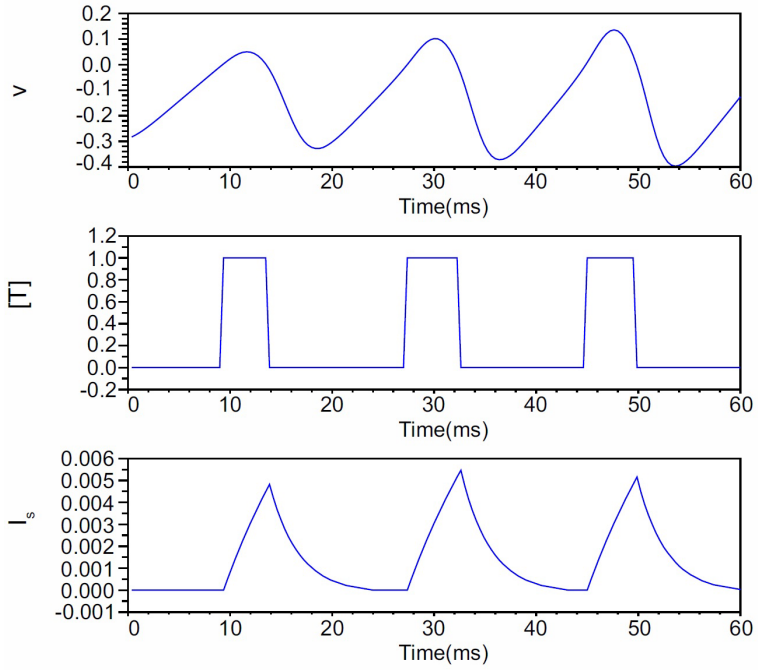

図 2.2　シナプス電流生成の数値シミュレーション [11] より引用

2.3　シリコンニューロンとシリコンシナプスの接続

式 (2.1) の I_{stim} は他のニューロンから与えられたシナプス電流の総和である。シナプス電流の総和を取るとき、各ニューロン間の接続強さを表すシナプス重みがかけられる。I_{stim} の計算式は以下に示す。

$$I^i_{stim} = c\sum_{j=1}^{N} W_{ij} I^j_s \tag{2.8}$$

i と j はニューロン ID、I^i_{stim} は i 番目ニューロンへの刺激電流、N はニューロンの数である。係数 c は I^i_{stim} が適切な範囲内に収まるようにするための定数であり、c の値によってニューロンが定期的に発火する。本研究で用いた c の値は 0.03125 である。W_{ij} は j 番目のニューロンから i 番目のニューロンへの接続の強さを表す。W_{ij} の値が正でありば後シナプスニューロンは興奮され、負でありば後シナプスニューロンは抑制される。

第3章　大規模スパイキング神経ネットワークのコンセプト

単一 VLSI チップにシリコン神経ネットワークを構築するには、ニューロン数やその接続にリソース上の制限が生じる。従って複数の VLSI チップを接続する手法を確立することが大事である。第 3 章では、大規模シリコン・スパイキング神経ネットワークのコンセプトと、スパイキング神経ネットワーク間の通信方法について述べる。

3.1　ローカル・ネットワーク

河野研究室では、複数の VLSI チップからなる大規模のシリコン・スパイキング神経ネットワークのコンセプトを持っていた。大規模シリコン・スパイキング神経ネットワークのコンセプトは階層構造である。最終的な大規模シリコン・スパイキング神経ネットワークは「ローカル・エリア・ネットワーク」がお互い接続しているコンセプトである。「ローカル・エリア・ネットワーク」は脳の大脳皮質に相当する規模である。「ローカル・エリア・ネットワーク」は「ローカル・ネットワーク」がお互い接続しているコンセプである。「ローカル・ネットワーク」は脳のコラムの規模に相当する。最後に「ローカル・ネットワーク」は「スモール・ネットワーク」がお互い接続しているコンセプトであって、「スモール・ネットワーク」は単一 VLSI チップに相当する。図 3.1 はローカル・ネットワークのコンセプトを図として示したものである。

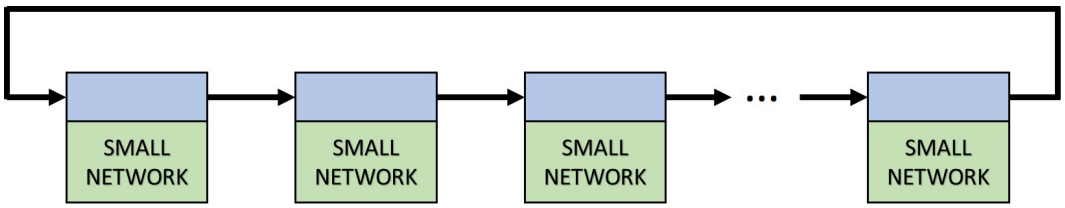

図 3.1　ローカル・ネットワークのコンセプト

「スモール・ネットワーク」は、ニューロン同士がお互い密につながっているシリコン・スパイキング神経ネットワークである。本研究では FPGA を用いてスパイキング神経ネットワークを構築しており、最新の FPGA は約 1000 ニューロンの全結合のネットワークを実装できると言われて

いる。河野研究室では 1024 ニューロンの FPGA を繋いて 64,000 ニューロンや 128,000 ニューロンからなるローカルネットワークをコンセプトとしている。それは人間の脳の 1 コラムが約 60000 ニューロンから構成されると言われているからである。従って、64 または 128FPGA を接続させる手法が必要である。その役割を果たすのが図 3.1 の青の通信プロトコルであり、拡大したのが次の図 3.2 である。

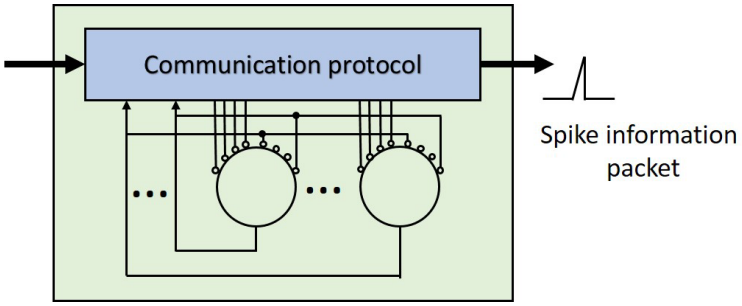

図 3.2　スモール・ネットワークの拡大図

通信プロトコルは「Spike Information Packet」(以降、パケット) を受け取り、他スモール・ネットワークからのシナプス電流を生成し自分のローカル・ネットワークに与える。また自分のスモール・ネットワークの発火情報からパケットを作り出し隣の FPGA に渡す。この方法で FPGA 同士はリング上につながり、各 FPGA から生成されたパケットが 1 周しながら他 FPGA に渡される。

3.2　FPGA 間の通信方法

本節ではローカルネットワークを実装した FPGA 間の通信方式について述べる。通信方式としては Address Event Representation(AER) を用いる。AER 方式とは、一方のチップで起きたイベントをもう一方のチップに伝達しようとするとき、イベントの起きたアドレスの順番のみ他チップに送信し、受け取った側ではアドレスの順番のみから新しく元のイベントを再現する方式である。その概念図を以下の図 3.3 に示す。

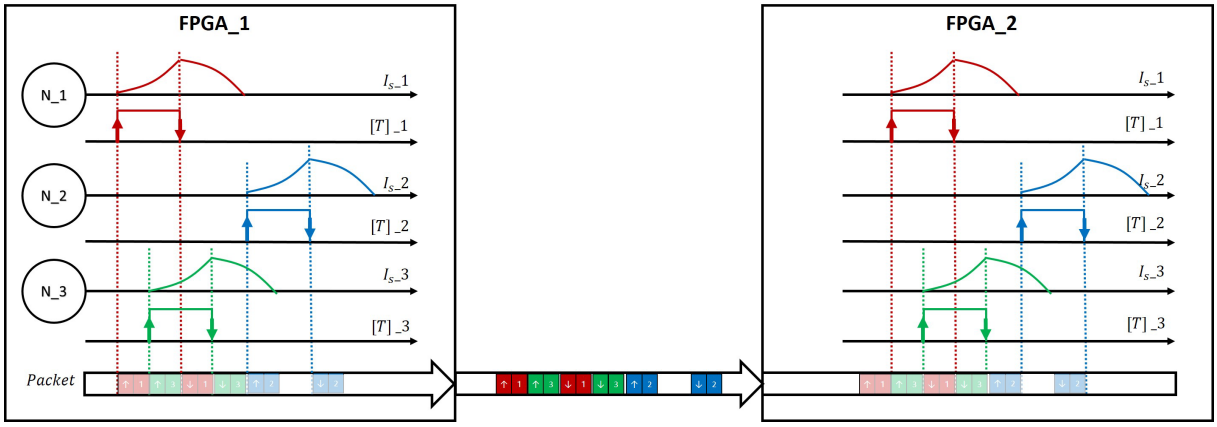

図 3.3　スパイキング神経ネットワークにおける AER の概略図

図 3.3 は、FPGA1 で生成されたシナプス電流 I_s を FPGA2 に伝達する方法を説明している。図 2.2 から分かるように、膜電位が閾値を超えている間だけシグナル [T] がオンになり、シナプス電流が上昇する。つまり、シナプス電流はシグナル [T] のオン・オフによって生成されるともいえる。したがって、FPGA1 で生成されたシナプス電流を送るときは、ニューロン ID と 1 ビットで表現されるシグナル [T] のみを送信し、受け取った側では送られてきたニューロン ID とシグナル [T] から元のシナプス電流を再生成する方法である。実際に送られるデータはパケット形式であって、本研究で定義したパケットの詳細を以下の図 3.4 に示す。

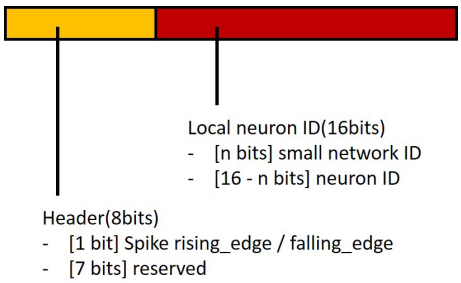

図 3.4　パケットの詳細

パケットは 8 ビットのヘッダーと 16 ビットのローカル・ニューロン ID から構成される。ヘッダーの 1 ビットはシグナル T の情報であり、[T] が立ち上がる時点で 1、[T] がたち下がる時点で 0 である。残りの 7 ビットは、通信プロトコルを拡張していくときに備えた未使用のビットである。

ローカル・ニューロン ID は、n ビットのスモール・ネットワーク ID と、16-n ビットのニューロン ID から構成される。スモール・ネットワーク ID は FPGA の ID であり、各 FPGA 内のニューロン ID が 16-n ビットで表現される。

第4章　スパイキング神経ネットワークと通信プロトコルのFPGA実装

4.1　スパイキング神経ネットワークの構造

本研究で構築したスパイキング神経ネットワーク（図 3.1 のスモール・ネットワーク）は先行研究の [12] を主に参考にして構築した。その規模は 256 ニューロンの全結合ネットワークである。全体の構造を図 4.1 に示す。スパイキング神経ネットワークは 16 つのシリコン・ニューラルネットワーク・モジュール (SNNM ｓ) から構成され、各 SNNM は 16 ニューロンの膜電位やシナプス電流をまとめて計算する。各 SNNM は、DSSN・ユニット、シリコンシナプス・ユニット、アキュムレータ・ユニット、ラーニング・ユニットの 4 つのユニットから構成される。DSSN・ユニットではニューロンの膜電位が計算され、膜電位と閾値との関係を考慮しシリコンシナプス・ユニットで後シナプス電流を生成する。アキュムレータ・ユニットでは、生成されたシナプス電流の総和をとって I_{stim} を生成する。ラーニング・ユニットでは、I_{stim} の計算に用いられるシナプス重みの更新を行う。

4.1.1　DSSN・ユニット

DSSN・ユニットでは、DSSN モデルの計算を行う。式 2.1〜2.4 の計算はオイラー法に従って以下の式 4.1、4.2 によって計算される。パラメータ値は表 2.1 の値である。

$$v(t+\Delta t) = \begin{cases} v(t) + \frac{\Delta t \varphi}{\tau}(8v^2(t) + 4v(t) - n(t) + I_0 + I_{stim}) & when \quad v < 0 \\ v(t) + \frac{\Delta t \varphi}{\tau}(-8v^2(t) + 4v(t) - n(t) + I_0 + I_{stim}) & when \quad v \geq 0 \end{cases} \tag{4.1}$$

$$n(t+\Delta t) = \begin{cases} n(t) + \frac{\Delta t}{\tau}(4v^2(t) + 2v(t) + \frac{1}{2}v(t) - \frac{1707}{2^{15}} - n(t)) & when \quad v < r \\ n(t) + \frac{\Delta t}{\tau}(16v^2(t) + 8v(t) - v(t) + \frac{2560}{2^{15}} - n(t)) & when \quad v \geq r \end{cases} \tag{4.2}$$

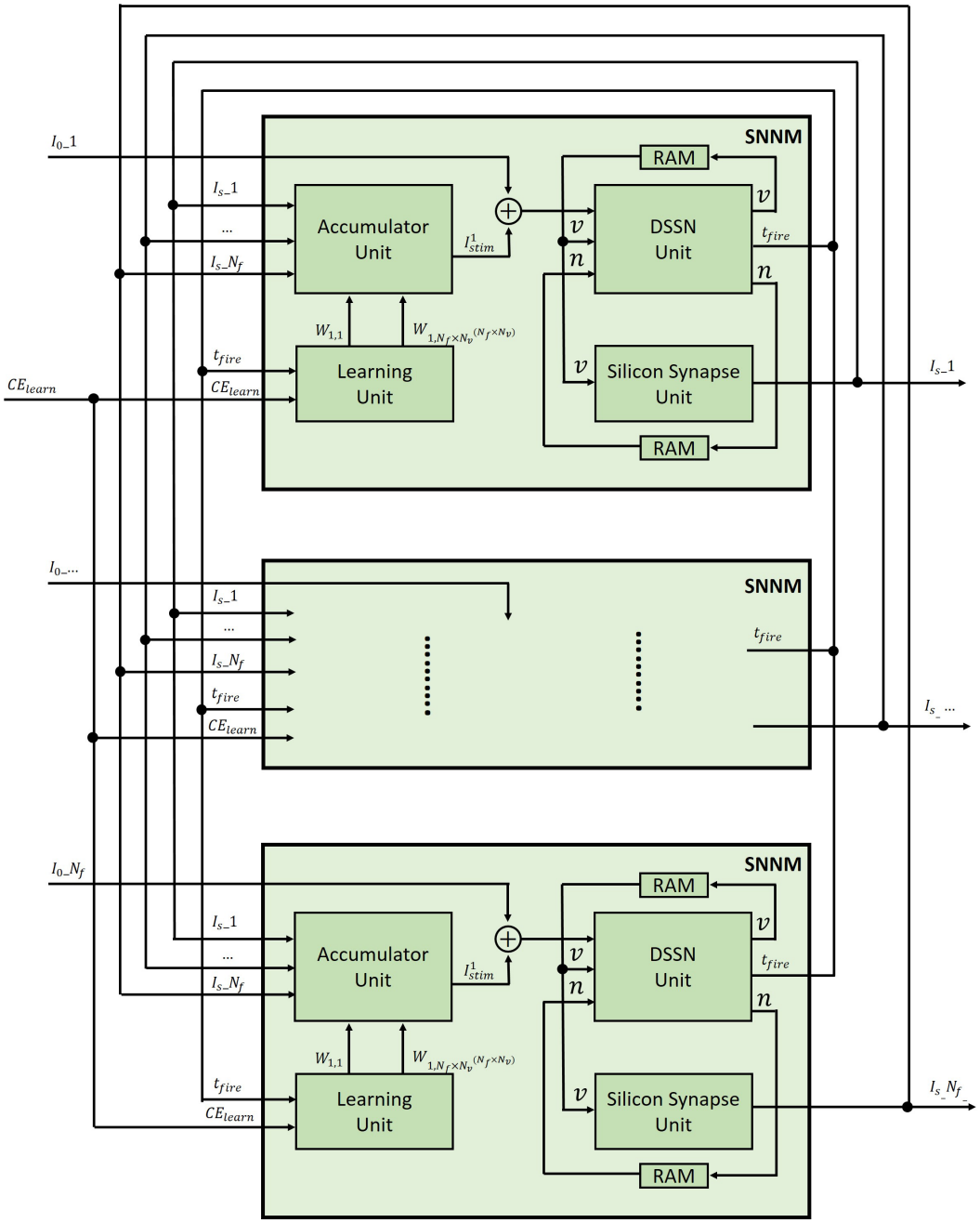

図 4.1　スパイキング神経ネットワークの構造

更新時間 Δt は 0.375ms として設定された。式 4.1、4.2 の 2^n として表せる係数は、掛算器を用いるのではなく、シフト演算を用いることでハードウェア・リソースの削減を図った。v と n は 18 ビットの符号付き固定小数点として計算され、そのうち 15 ビットが小数点の計算に用いられた。図 4.2、図 4.3 は式 4.1、4.2 の s と v の計算回路のブロックダイアグラムである。

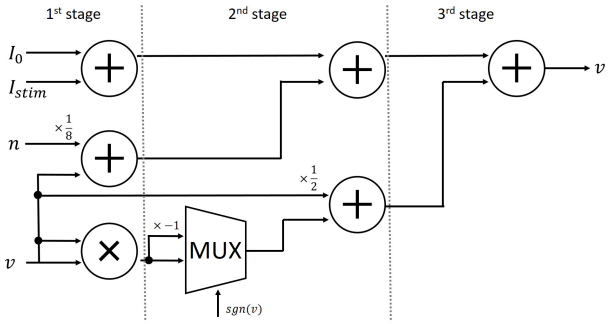

図 4.2　v 計算回路のブロックダイアグラム

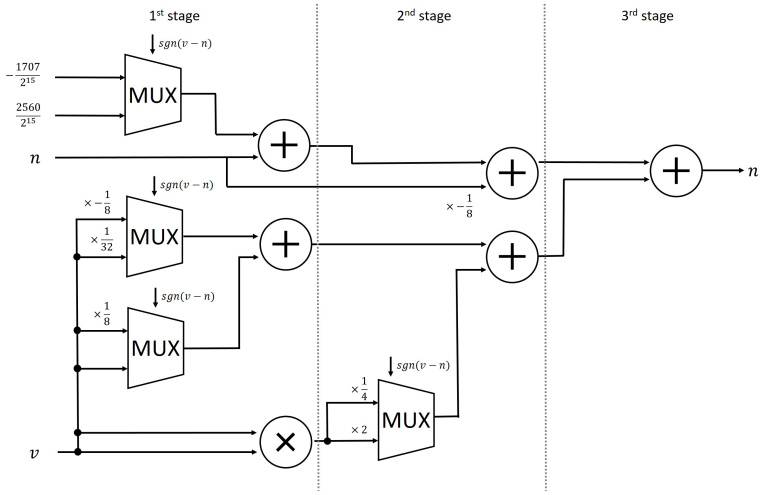

図 4.3　n 計算回路のブロックダイアグラム

＋、×、 MUX はそれぞれ足算器、掛算器、マルチプレクスを表す。マルチプレクスはコントロール・シグナルによって複数の入力のうち一つを選択する役割を果たす。式 4.1、4.2 で用いられた掛算器はそれぞれ 1 つであって、v^2 の計算に用いられた。従って、一つの DSSN・ユニットでは 1 つの掛算器、10 つの足算器、5 つのマルチプレクスが用いられる。マルチプレクスは論理回路として設計され、足算器と掛算器には XILINX 社の LogiCORE IP が用いられた。足算器と

掛算器の計算には 1 クロックが消費され、v と n の計算には合計 3 クロックが消費される。

4.1.2 シリコンシナプス・ユニット

シリコンシナプス・ユニットではシナプス電流 I_s を計算する。式 2.7 をオイラー法に従い、式 4.3 としてアップデートの計算を行う。ハードウェア・リソースの削減のため、$\Delta t\alpha = 2^{-5}$ と $\Delta t\beta = 2^{-3}$ として定義した。二つの方程式間の切り替えは、膜電位が閾値を超えているかどうかを表すシグナル $[T]$ によって決まる。図 4.4 は I_s 計算回路のブロックダイアグラムである。2 つの足算器と 1 つの掛算器が用いらえれ、2 クロックにかけて計算される。

$$I_s(t+\Delta t) = \begin{cases} I_s(t) + \Delta t\alpha(1 - I_s(t)) & when \quad [T] = 1 \\ I_s(t) - \Delta t\beta I_s(t) & when \quad [T] = 0 \end{cases} \tag{4.3}$$

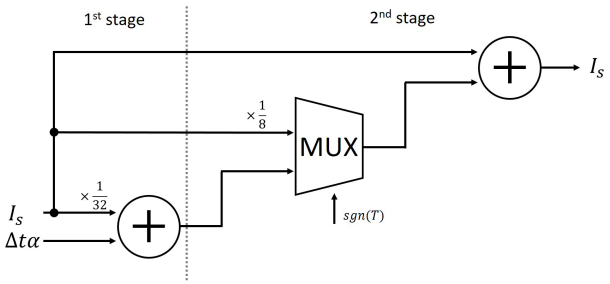

図 4.4　I_s 計算回路のブロックダイアグラム

4.1.3 アキュムレータ・ユニット

式 2.8 の計算には $(N_f \times N_v - 1) \times N_v$ 回の足算と $N_f \times N_v^2$ 回の掛算が必要とされる。掛算の結果を足していく計算なので、$N_f \times N_v^2 + 1$ クロックをかけて I_{stim} が更新される訳だが、更新時間を短縮するため、本研究のデザインでは 4 つの足算器と 4 つ掛算器を並列として計算を行った。並列構造によって更新時間は $\frac{N_f \times N_v^2}{4} + 1$ と短縮された。図 4.5 は I_{stim} 計算回路のブロックダイアグラムであり、2 クロックにかけて計算される。

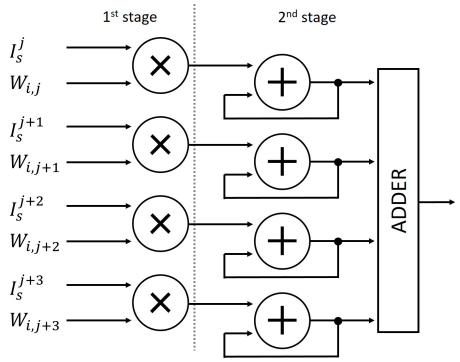

図 4.5　I_{stim} 計算回路のブロックダイアグラム

4.1.4　ラーニング・ユニット

ラーニング・ユニットも 4 つの並列構造を持ち、各並列ラインは以下の計算を行う。

$$W_{ij}(t+\Delta t) = \begin{cases} W_{ij}(t) & when \quad CE_{learn} = 0 \\ W_{ij}(t) + \Delta W_{ij} & when \quad CE_{learn} = 1 \end{cases} \tag{4.4}$$

CE_{learn} は学習モードを選択するシグナルである。$CE_{learn} = 0$ では、相関学習モードとして、W_{ij} が定数として保たれる 1 。$CE_{learn} = 1$ では、スパイク時間依存学習モードとして、W_{ij} が更新される。ここで足される ΔW_{ij} は式 1.18 で定義された式である。ラーニング・ユニットは図 4.6(A) に示すように、2 つの RAM と 1 つの足算器によって構成される。RAM1 と RAM2 はそれぞれ W_{ij} と ΔW_{ij} を保持する。図 4.6(B) は RAM の拡大図である。

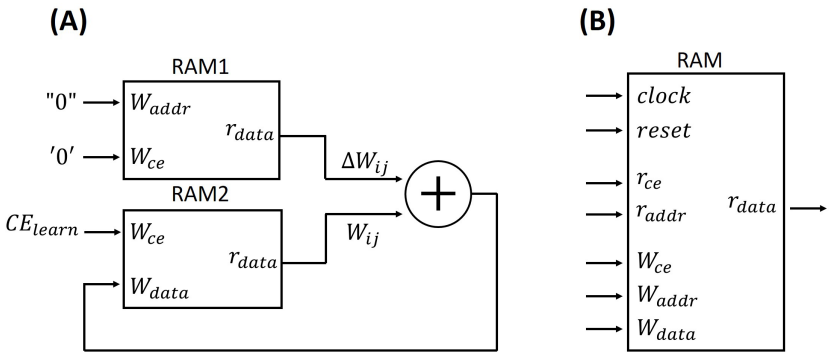

図 4.6　ラーニング・ユニットのブロックダイアグラム

4.1.5　システム・クロック

アキュムレータ・ユニットとラーニング・ユニットは両方とも $\frac{N_f \times N_v^2}{4} + 1$ クロックにかけて更新が行われる。DSSN・ユニットとシリコンシナプス・ユニットにおける更新時間は両方とも $N_v + 4$ である。図 4.7 はアップデート・ステップのクロックサイクルを表したものである。4 つのユニットはパイプラインの構造で計算される。

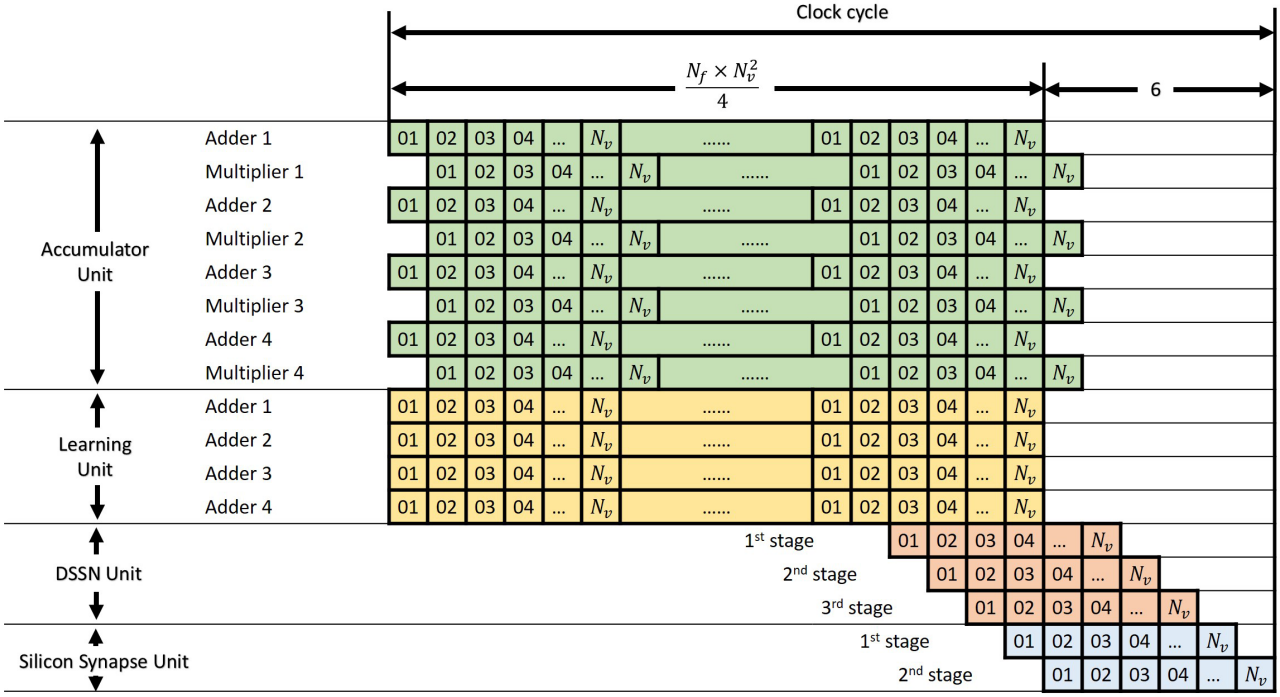

図 4.7　アップデート・ステップのクロックサイクル

アキュムレータ･ユニットは最初のクロックから $\frac{N_f \times N_v^2}{4} + 1$ クロックまでを消費し、ラーニング･ユニットは最初のクロックから $\frac{N_f \times N_v^2}{4}$ クロックまでを消費する。DSSN･ユニットは $\frac{N_f \times N_v^2}{4} - N_v + 3$ 番目のクロックから計算をスタートし、合計 $N_v + 2$ のクロックを消費する。シリコンシナプス・ユニットは合計 $N_v + 1$ のクロックを消費し、1 度の更新に $\frac{N_f \times N_v^2}{4} + 6$ のクロックがかかる。従って、アップデート・ステップ Δt と回路を回すシステムクロック f_s との関係は次となる。

$$\Delta t = \frac{1}{f_s}\left(\frac{N_f \times N_v^2}{4} + 6\right) \tag{4.5}$$

4.2　通信プロトコルの構造

図 3.2 の通信プロトコルは、FPGA に実装されたすべてのニューロンに対して図 3.3 の動作を行う。それを実現するための、通信プロトコルの構造を以下の図に示す。通信プロトコルは、I_{stim} ジェネレータ、パケット・ジェネレータ、高速シリアル通信プロトコルから構成される。I_{stim} ジェネレータでは、受け取ったパケットからシナプス電流 I_s を生成し、それらの総和をとった I_{stim} をスモール・ネットワークへ刺激電流として与える。パケット・ジェネレータでは、スモール・ネットワーク内の各ニューロンに対するシグナル [T] に基づき、図 3.4 のパケットを生成する。高速シリアル通信プロトコルでは、24 ビットのパケットをシリアル通信として FPGA 外部にデータを送信する。以降の節で各動作について述べる。

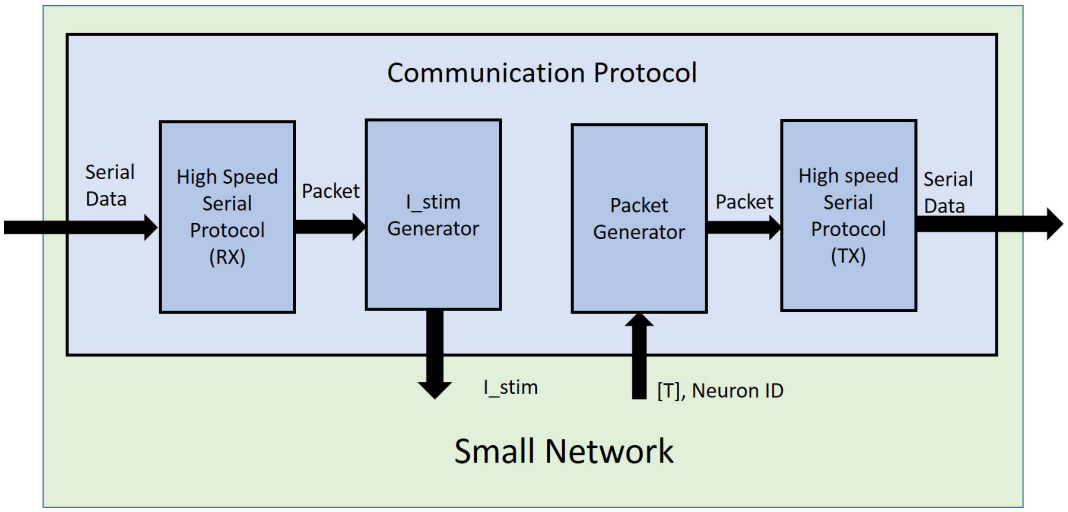

図 4.8　通信プロトコルの構造

4.2.1　パケット・ジェネレータ

「パケット・ジェネレータ」の動作を図 4.9 に示す。「パケット・ジェネレータ」はスモール・ネットワークの「DSSN・ユニット」から、膜電位 v と閾値との関係により生成されたシグナル [T] を受け取り、[T] の立ち上がりの時点で T=1 とニューロン ID の情報を含んだパケットを生成し、[T] の立ち下がりの時点で T=0 とニューロン ID の情報を含んだパケットを生成する。つまり、1 回の発火で 2 つのパケットが生成されるわけである。また、スモール・ネットワークでは N_f 個の「DSSN・ユニット」でニューロンの膜電位の計算を並列計算しているため、複数のニューロンで発火が同時に起こる可能性もあり得る。その最大の同時発火数は「DSSN・ユニット」の数である

N_f であるので、N_f 個のパケットをメモリに入れ、スモールネットワークのシステム・クロックの N_f 倍でメモリのパケットを出力する。そうやってメモリから出力されたパケットは高速シリアル通信プロトコルを通して隣の FPGA に渡される。

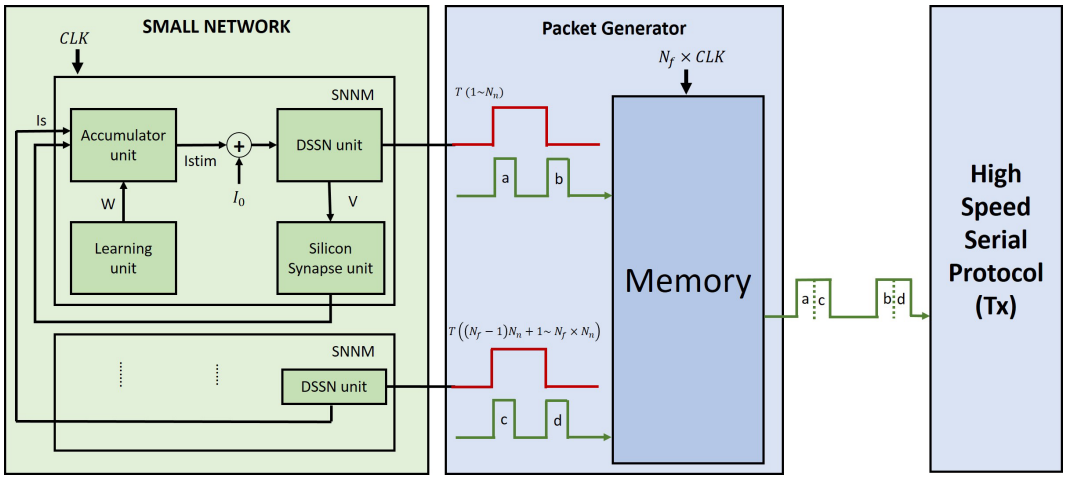

図 4.9　パケット・ジェネレータの動作

4.2.2　I_{stim} ジェネレータ

I_{stim} ジェネレータでは、受け取ったパケットからシナプス電流 I_s を生成し、それらの総和をとり I_{stim} を生成し、スモール・ネットワークへ刺激電流として与える。I_{stim} ジェネレータの構成を図 4.10 に示す。I_{stim} ジェネレータは、「パケット・セパレータ」、「シリコンシナプス・ユニット」、「アキュムレータ・ユニット」、「ラーニング・ユニット」から構成される。「シリコンシナプス・ユニット」、「アキュムレータ・ユニット」、「ラーニング・ユニット」は、4.1 節で述べたものと同じ構成である。

まず受け取ったパケットを、「パケット・セパレータ」でニューロン ID をコントロールシグナルとして N_f ラインに分ける。つまり、「パケット・セパレータ」はディ・マルチプレクサの役割をする。N_f ラインに分けられたシグナル [T] は、「シリコンシナプス・ユニット」入力され、式 (4.3) に従って I_s が生成される。生成された I_s は「アキュムレータ・ユニット」で式 (2.8) に従って、I_{stim} が計算される。I_{stim} 計算の際に、式 (4.4) の ΔW_{ij} も計算されるが、それはは前シナプスニューロンの発火時刻と後シナプスニューロンの発火時刻によって決まる。図 4.10 の構造において前シナプスニューロンの発火時刻は、「I_{stim} ジェネレータ」内で計算されるシナプス電流の発生時刻であり、後シナプスニューロンの発火時刻は「スモール・ネットワーク」内で計算される

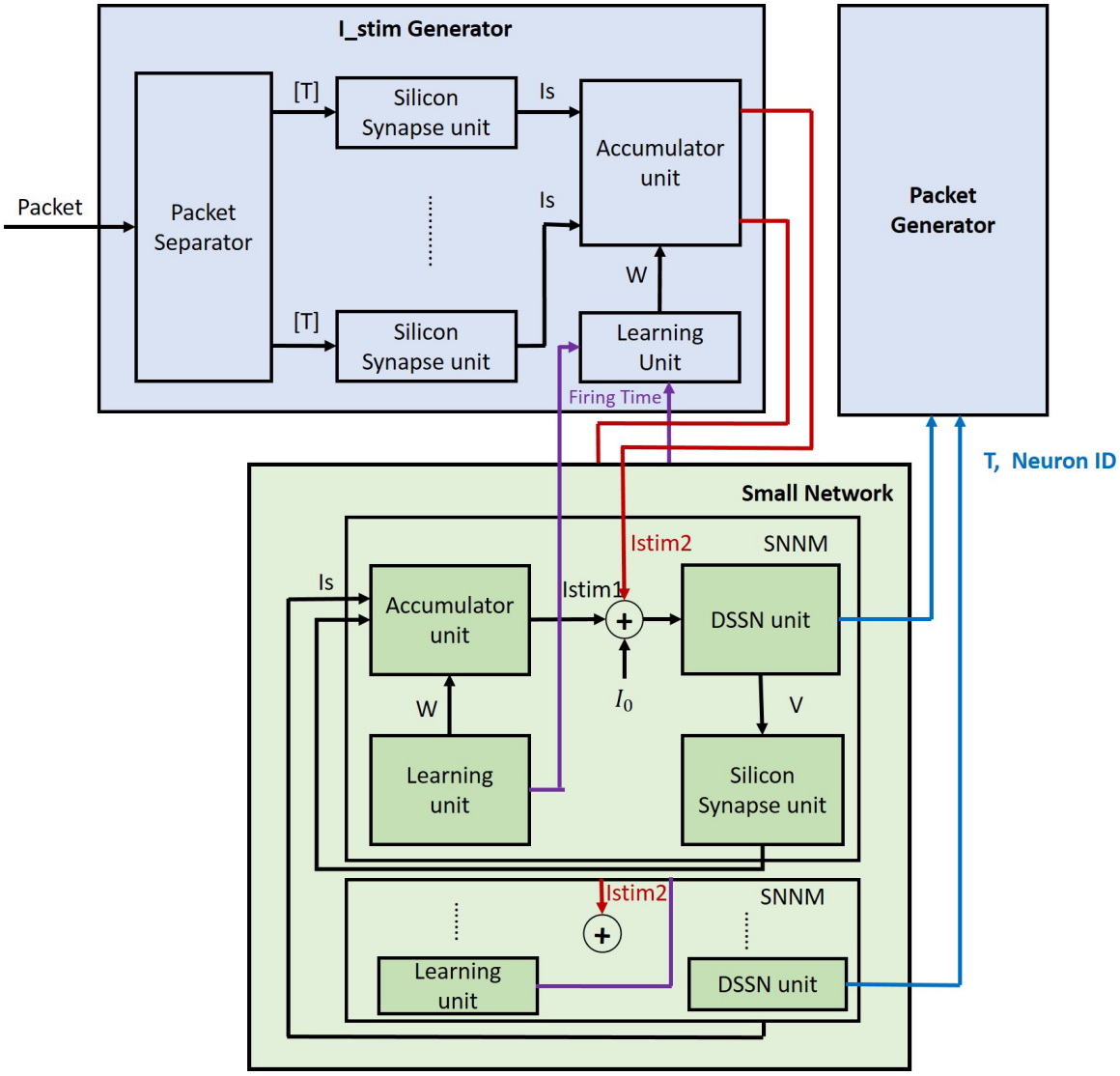

図 4.10　I_{stim} ジェネレータの構成

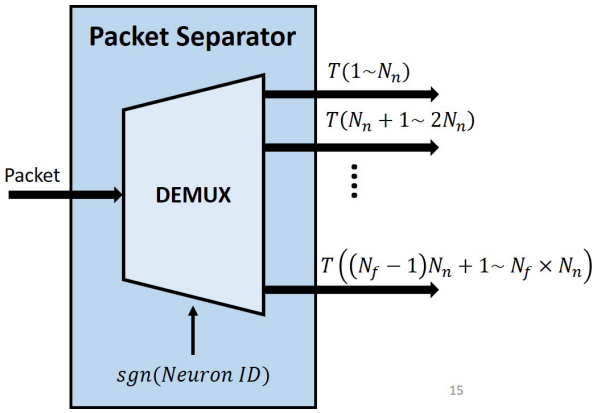

図 4.11　パケット・セパレータ

シナプス電流の発生時刻に相当する。時刻のデータは「ラーニング・ユニット」で計算されるので、図 4.10 の紫色の後シナプスニューロンの発火時刻が「I_{stim} ジェネレータ」の「ラーニング・ユニット」に入力される。そうやって生成された I_{stim} は「スモール・ネットワーク」の「DSSN・ユニット」の新しい刺激電流として入力される。従って、膜電位 v の計算式 4.1 の I_{stim} は以下となる。

$$I_{stim} = \frac{1}{2} I_{stim1} + \frac{1}{2} I_{stim2} \tag{4.6}$$

4.2.3　高速シリアル通信プロトコル

「高速シリアル通信プロトコル」は作られたパケットを隣の FPGA に渡す役割を果たす。「高速シリアル通信プロトコル」はトランシーバー部 (Tx) とトランシーバー部 (Rx) 部から構成され、その接続を図 4.12 に示す。FPGA の外部では高速シリアル通信であるため、Low voltage differential signaling (LVDS) でデータが渡される。

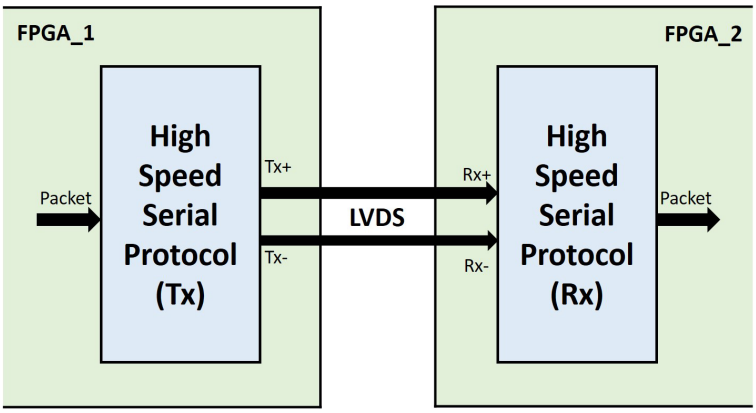

図 4.12　高速シリアル通信プロトコルの接続

本研究では高速シリアル通信プロトコルとして XILINX 社の LogiCORE IP の「Aurora64B/66B」を用いた。「Aurora64B/66B」は XILINX 社の FPGA にあるトランシーバーを簡単に実装し、軽量のユーザー・インターフェスを提供する。「Aurora64B/66B」の仕様を以下の表 4.1 にまとめた。

表 4.1　Aurora64B/66B の仕様

Supported Maximum Lane Number	12
Supported Maximum Line Rate	11.3 Gbps
Supported Lane Width/lane	64 bits
Encoding Method	64b/66b

「高速シリアル通信プロトコル」を用いてローカル・ネットワークを構築するためには、図 4.7 の更新時間 Δt 以内にパケットが 1 周しなければならない。言い換えれば、Δt 以内に自分以外の FPGA から生成されたすべてのパケットを受け取らなければならないことである。先行研究 [11] によると、Δt が $375\mu s$ 以内であるとき、シナプス電流の波形が生体のそれを再現できることが調べられた。従って、ローカル・ネットワークを構成するスモール・ネットワークの数が 128 であり、各スモールネットワークを構成するニューロン数が 1024 であるとするとき、通信プロトコルのスループット A は以下の要件が要求される。

$$A[bps] \times 375[\mu s] > 128 \times 1024 \times 24[bits] \times 2 \tag{4.7}$$

$$A > 16.7[Gbps] \tag{4.8}$$

表 4.1 の Aurora64b66b の仕様からわかるように、AURORA は 12 レーンまで実装可能であり、2 レーン以上を用いれば要求スループットを満たせることが分かる。

4.3 通信プロトコルのディレイ

スモール・ネットワークのシナプス電流からパケットが生成され、隣のスモール・ネットワークに刺激電流として渡されるまで、通信プロトコルでの計算や論理回路の構造によってディレイが生じる。次の図 4.13 は構築したプロトコルのディレイを示す。

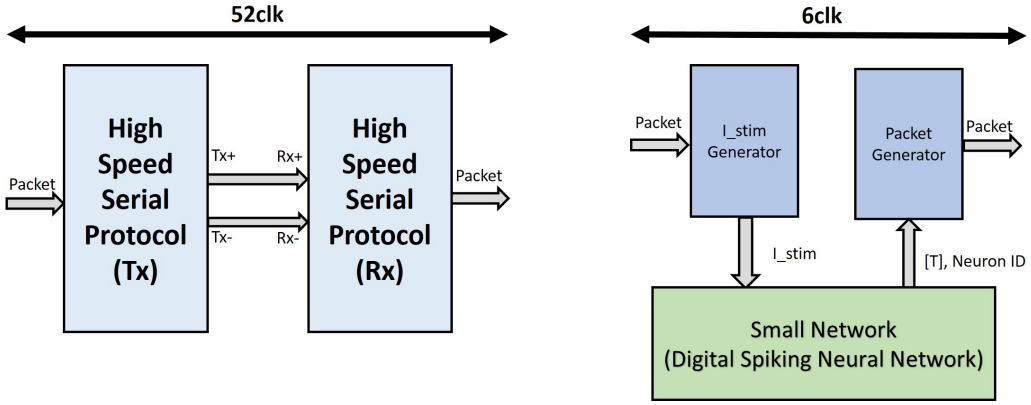

図 4.13　通信プロトコルのディレイ

「高速シリアル通信プロトコル」を通るとき 52clk のディレイが生じ、「パケット・ジェネレータ」と「I_{stim} ジェネレータ」を通りながら 6clk のディレイが生じる。「高速シリアル通信プロトコル」の動作クロックは「スモール・ネットワーク」の動作クロックの N_f 倍である。動作クロックの違いを考慮し、1024 ニューロンのスモールネットワーク ($N_f = 64$、$N_n = 16$) 向けの通信プロトコルのディレイを計算してみた。ただし、クロックの違いが 64 倍は大きいので、4 レーン「高速シリアル通信プロトコル」を用いると想定し、16 倍のクロック差で動作すると想定した。まず $N_f = 64$ であるため、更新時間 Δt が 375μs であるとすると、式 4.5 によりシステムクロックは 10.9MHz となる。また、1 つの通信プロトコルを通る時に $\frac{52}{16} + 6$clk のディレイが生じる。したがって、128 個の 10241024 ニューロンのスモールネットワークをリング状につなげた時生じるディレイを計算してみると

$$T_{delay} = \frac{\left(\frac{52}{16} + 6\right) \times 128}{[10.9MHz]} = 108.6[\mu s] \tag{4.9}$$

となる。4.2.3 節でも述べたように、Δt=375μs の間にパケットが 1 周しなければならない要件があるが、T_{delay}=108.6μs であるため、その要件が満たされるといえる。

4.4 FPGA 実装

構築したローカル･ネットワークおよび通信プロトコルを FPGA デバイスに実装した。ボードとして XILINX Virtex-7 XC7VX690T-2FFG1761C FPGA 搭載の XILINX 社の VC709 評価ボードを用いた。しかし、実装の段階において「高速シリアル通信プロトコル部」に当たる AURORA64b66b およびその周辺モジュールを加えたところでタイミングエラーが生じ、完全なる通信プロトコルの実装までには至らなかった。したがって、リソースの占有率を確かめるため、AURORA とその周辺モジュールを除外した「通信プロトコル」と「スモール・ネットワーク」を実装した。スモール・ネットワークの規模は 256 ニューロンの全結合ネットワーク ($N_f = 16, N_v = 16$) である。

デジタル回路の開発ツールとしては XILINX 社の Vivado Design Suite 2015.4 を用い、合成や実装のツールとしては Vivado Design Suite で提供されている「Vivado Synthesis Tool」、「Vivado Implementation Tool」を用いた。次の表 4.2 と 4.3 にローカルネットワーク部および通信プロトコル部のリソース利用の詳細を示す。

表 4.2　スモール・ネットワーク部のリソース利用

Logic Utilization	Utilization	Availiable
LUT	48347(11.16%)	433200
FF	42050(4.85%)	866400
BRAM	64(4.35%)	1470
DSP	80(2.22%)	3600

BRAM はシナプス重み W_{ij} の保持に用いられ、DSP は掛算器として用いられる。表 4.2 と 4.3 から分かるように、スモール・ネットワーク部で利用される DSP 数は通信プロトコル部で利用される数より 16 個多い。この 16 個の DSP は、図 4.2 と 4.3 で用いられる 1 つの掛算器であって、スモール・ネットワークを構築する DSSN ユニット数と一致する。また、スモール・ネットワーク部および通信プロトコル部は各 256^2 シナプスを持つため、1BRAM 当たり $256^2/64 = 1024$ シナプスのデータを保持すると考えられる。

表 4.3　通信プロトコル部のリソース利用

Logic Utilization	Utilization	Availiable
LUT	51589(11.90%)	433200
FF	44688(5.16%)	866400
BRAM	64(4.35%)	1470
DSP	64(1.78%)	3600

高速シリアル通信部のリソース利用率を調べるため、AURORA64 b 66 b およびその周辺モジュールを実装した。周辺モジュールは XILINX 社から提供される Example Code を用いた。以下に高速シリアル通信部のリソース利用の詳細を示す。

表 4.4　高速シリアル通信部 (AURORA) のリソース利用

Logic Utilization	Utilization	Availiable
LUT	3069(0.71%)	433200
FF	7362(0.85%)	866400
BRAM	4(0.27%)	1470

表 4.2〜4.4 より、1024 ニューロンの全結合のローカルネットワークのリソース利用を計算してみた。その規模は 256 ニューロンの全結合ネットワークの 4 倍となるため、LUT、FF、DSP に関しては 4 倍になる考えらえる。BRAM 数に関してはシナプス数が 16 倍となるため、シナプス重みの保持に用いられる BRAM 数は $64 \times 16 = 1024$ となる。本研究で用いた Virtex-7 XC7VX690T-2FFG1761C に対して 1024 ニューロンの全結合のローカルネットワークおよび通信プロトコルを実装すると考えるとき、ローカルネットワークと高速シリアル通信部 (AURORA) に用いられる BRAM 数を 1028 であり、通信プロトコルに用いられる BRAM 数は 442 に留まってしまう。したがって、1BRAM 当たり 1024 シナプスデータを保持できることを考慮すると 442×1024 シナプスを接続できる通信プロトコルが実装できると考えられる。

第5章　連想記憶による通信プロトコルの動作確認

連想記憶とは入力パターンの欠片から学習されたパターンに一番近い 1 つのパターンを検出することを意味する。このような記憶のメカニズムは動物と人間の海馬や皮質の一部からも発見された。数多い研究者がニューラルネットワークに基づいた記憶能力を研究してきており、その一つとして Hopfield は全結合のニューラルネットワークは連想記憶が可能であると示した [13]。図 5.1 は Hopfield ニューラルネットワークの構造を示す。ニューロン同士は全結合であり、シナプス重み W_{ij} によってニューロン間の接続の強さを表している。各ニューロンの状態は式 5.1 の 2 状態を表す関数 $sgn(.)$ によってコーディングされる。

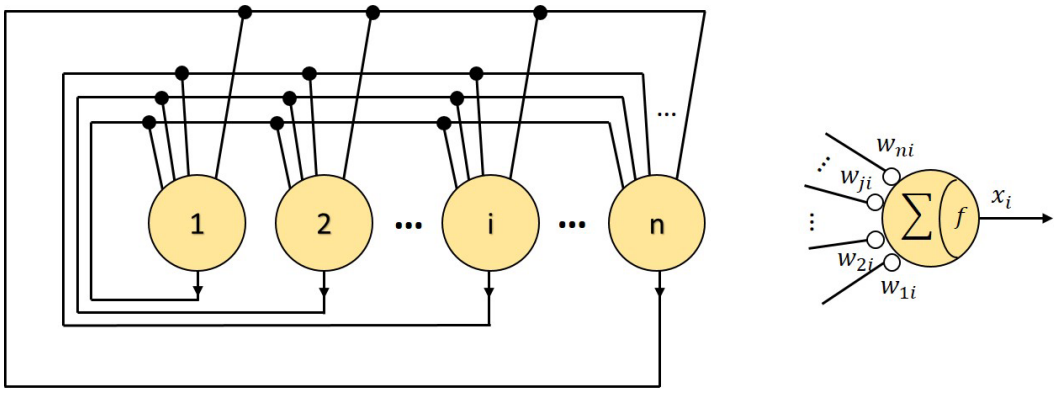

図 5.1　Hopfield ネットワークの構造

$$x_i = sgn\left(\sum_{j=1}^{n} W_{ij}x_j\right) \equiv \begin{cases} 1 & when \quad \sum_{j=1}^{n} W_{ij}x_j \geq 0 \\ -1 & when \quad \sum_{j=1}^{n} W_{ij}x_j < 0 \end{cases} \tag{5.1}$$

x_i はニューロン i の状態を表す。W_{ij} はニューロン i と j 間の接続の強さを表すシナプス重みである。自分以外のニューロンからの刺激の総和 $\sum_{j=1}^{n} W_{ij}x_j$ が 0 より小さければニューロンの出力は-1 となり、0 より大きければ+1 となる。

[14] で紹介されるオーバーラップは学習パターンと入力パターン間の類似性を図るための指標である。次の式 M_u が u 番目の学習パターンとニューロンの状態とのオーバーラップの計算式である。

$$M_u(t) = \frac{1}{N}\left|\sum_{j=1}^{N} x_j^u exp\left(i\phi(t)\right)\right| \tag{5.2}$$

N はニューロンの数、$\phi_j(t)$ は時刻 t におけるニューロン j の位相値である。$\phi_j(t)$ は次の式により決まる。

$$\phi_j(t) = 2\pi k + 2\pi\frac{t - t_j^k}{t_j^{k+1} - t_j^k} \quad \left(t_j^k \leq t < t_j^{k+1}\right) \tag{5.3}$$

t_j^k はニューロン j の k 番目の発火時刻である。本研究で構築したニューラルネットワークは定期的に発火し、$\phi_j(t)$ がニューロンの位相をコーディングする。従って、式 5.2 により M_u が 1 になるとニューラルネットワークがその学習パターンを検出できたといえる。

位相がコーディングされたニューラルネットワークでは、ニューロン間の同期が重要になってくるため、位相同期化指標 (Phase Synchronization Index, PSI) も同ニューラルネットワークの連想記憶動作の評価指標として用いることが出来る [15]。PSI は次の式によって求まり、0 と 1 の間の値を持つ。PSI が 1 のとき、ネットワークニューロンが同期して発火しているといえる。

$$PSI(t) = \frac{1}{N}\left|\sum_{j=1}^{N} exp\left(i2\phi(t)\right)\right| \tag{5.4}$$

$\phi_j(t)$ は式 5.3 の位相値である。ニューロンの発火が学習パターンと一致するときニューロンの位相は 0 か π となるため、$\phi_j(t)$ に係数 2 をかけ PSI をスケーリングする。

5.1 相関学習による連想記憶

5.1.1 ネットワークのコンフィギュレーション

本研究ではスモールネットワークとして全結合の 256 シリコンニューロンを構築し、通信プロトコルを通じて 2 つのスモールネットワークを繋げ、512 シリコンニューロンの全結合ネットワークを構築した。また式 4.4 を CE=0 と設定し、相関学習によって W_{ij} を決めた。相関学習とは学習パターンを用いて全結合ネットワークの W_{ij} を事前に決めておく学習方法である [13]。その式を次に示す。

$$W_{ij} = \begin{cases} \sum_{u=1}^{p} x_i^u x_j^u - 1 & when \quad i \neq j \\ 0 & when \quad i = j \end{cases} \tag{5.5}$$

p は学習パターン数、x_i^u は学習パターン u のニューロン i の状態を表す。学習パターンは黒白のバイナリパターンであり、x_i^u は ± 1 を持つ。最終的に用いた W_{ij} の計算式は $\frac{1}{p}$ でスケーリングした次の式を用いた。

$$W_{ij} = \begin{cases} \frac{1}{p} \sum_{u=1}^{p} x_i^u x_j^u & when \quad i \neq j \\ 0 & when \quad i = j \end{cases} \tag{5.6}$$

本研究では学習パターンとして白黒の画像を用いたため、白のピクセルは $x_i^u = -1$、黒のピクセルは $x_i^u = 1$ である。通信プロトコルを介した 512 全結合ネットワークにおける連想記憶を行った。512 全結合ネットワークに対する学習パターンは白黒 32 × 16 ピクセルの画像である (図 5.2(A))。入力パターンとしては学習パターンＡと 5 %～40 %のエラーを持つ 8 種類のパターンを用いた図 5.2(B)。

ピクセルの黒白によって異なる I_{stim} を入力することでスパイキング神経ットワークに入力パターンを与えた。ピクセルが黒の時 $(x_i = 1)$ に $I_{stim} = 0.0425$ のインパルス刺激を、ピクセルが白の時 $(x_i = -1)$ に $I_{stim} = 0$ を与えた。インパルス刺激は 45 回のアップデート間に入力した。その以降は $I_{stim} = 0.0295$ とした。

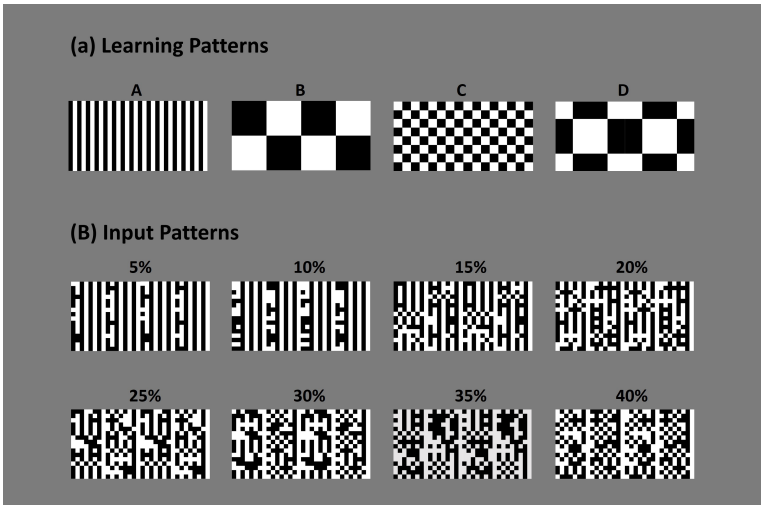

図 5.2　相関学習による連想記憶の学習パターン (A) と入力パターン (B)

5.1.2　結果

通信プロトコルを介して接続している 512 ニューロンの全結合ネットワークにおいて、5%〜40%の 8 種類の入力パターンに対する連想記憶を行った。8 種類の入力パターンは図 5.2 に示したものである。連想記憶の結果はデジタル回路のシミュレーターである ModelSim SE-64 10.1c から得られた。図 5.3〜5.5 は 20%、30%、35%のエラーを持った入力パターンに対するニューロン発火の様子である。黒い点がニューロンの発火を意味する。図 5.3〜5.5 において 15.75ms に入力パターンが現れた。その後、保存されたシナプス重みにしたがってニューロン活動が変化していくが、図 5.3 では 92.25ms からパターン A と $\overline{A}$ が順序に現れ始め、図 5.4 では 146.25ms からパターン A と $\overline{A}$ が順序に現れ始めた。図 5.5 ではパターン A と $\overline{A}$ が現れなかった。Hopfield ネットワークにおける連想記憶では、ある学習パターンとその逆パターンが決まった時間おきで順序に現れたとき正しく学習パターンが検出できたとことなので、図 5.3〜5.5 においては 20%、30%に対して正しい連想記憶の動作が確認できたといえる。

図 5.6 と図 5.7 は 5%〜40%の 8 種類の入力パターンに対するオーバーラップとニューロンの同期特性である。図 5.6 のオーバーラップに計算においては、ある学習パターンに対して M が 1 となるとニューロンの発火がその学習パターンと一致したことを意味する。図 5.6 から分かるように入力パターンが 30%までのエラーを持つときは学習パターン A と同じパターンが検出されたが、入力パターンの持つエラーが 35%、40%であるときはどのパターンも検出できなかった。特にエラーが 35%、40%の時は、どの M も振動しており、振幅は学習パターンによって異なった。図 5.7 のニューロンの同期特性の計算においては、ニューロンの発火が同期してかつ周期的であるとき 1 となる。したがって、連想記憶の動作において正しく学習パターンが検出できたといえるには、M も PSI も 1 にならなければならない。図 5.7 では、入力パターンが 30%までのエラーを持つと

きに時間が経つにつれて同期し、エラーが 35%、40%の時は同期しなくなることが分かる。したがって、図 5.6 と図 5.7 から入力パターンの持つエラーが 30%までの時は、学習パターンが正しく検出でき、エラーが 35%以上であるときは学習パターンが検出できないことが確認できた。また、学習パターンを思い出し始める時刻はエラーが 20%の時に最大の 390.375ms であって、入力パターンのエラー率と思い出すまでの時間に関係性はないと考えれる。

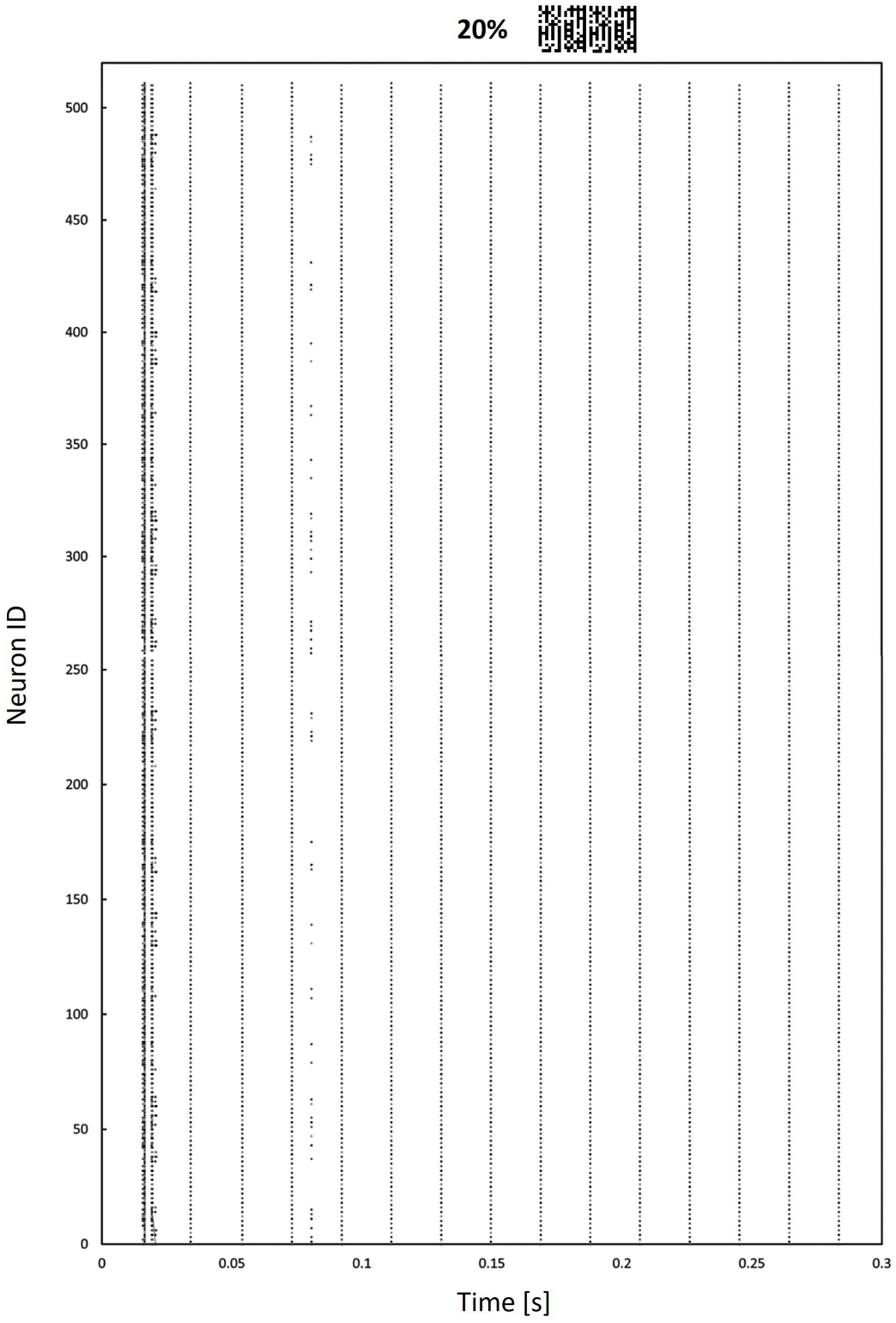

図 5.3　相関学習による連想記憶のニューロン発火 (入力パターンエラー 20%)

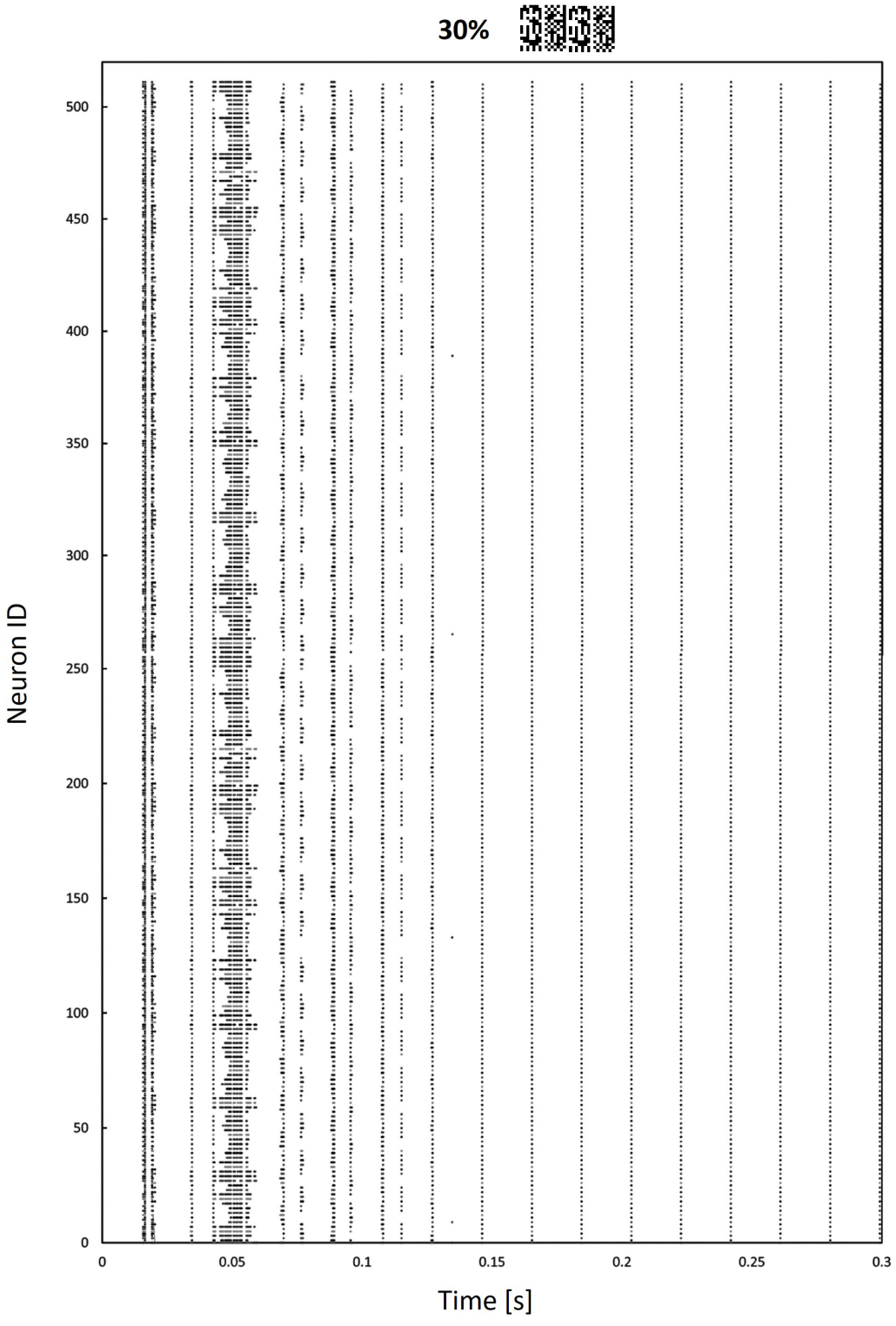

図 5.4　相関学習による連想記憶のニューロン発火 (入力パターンエラー 30%)

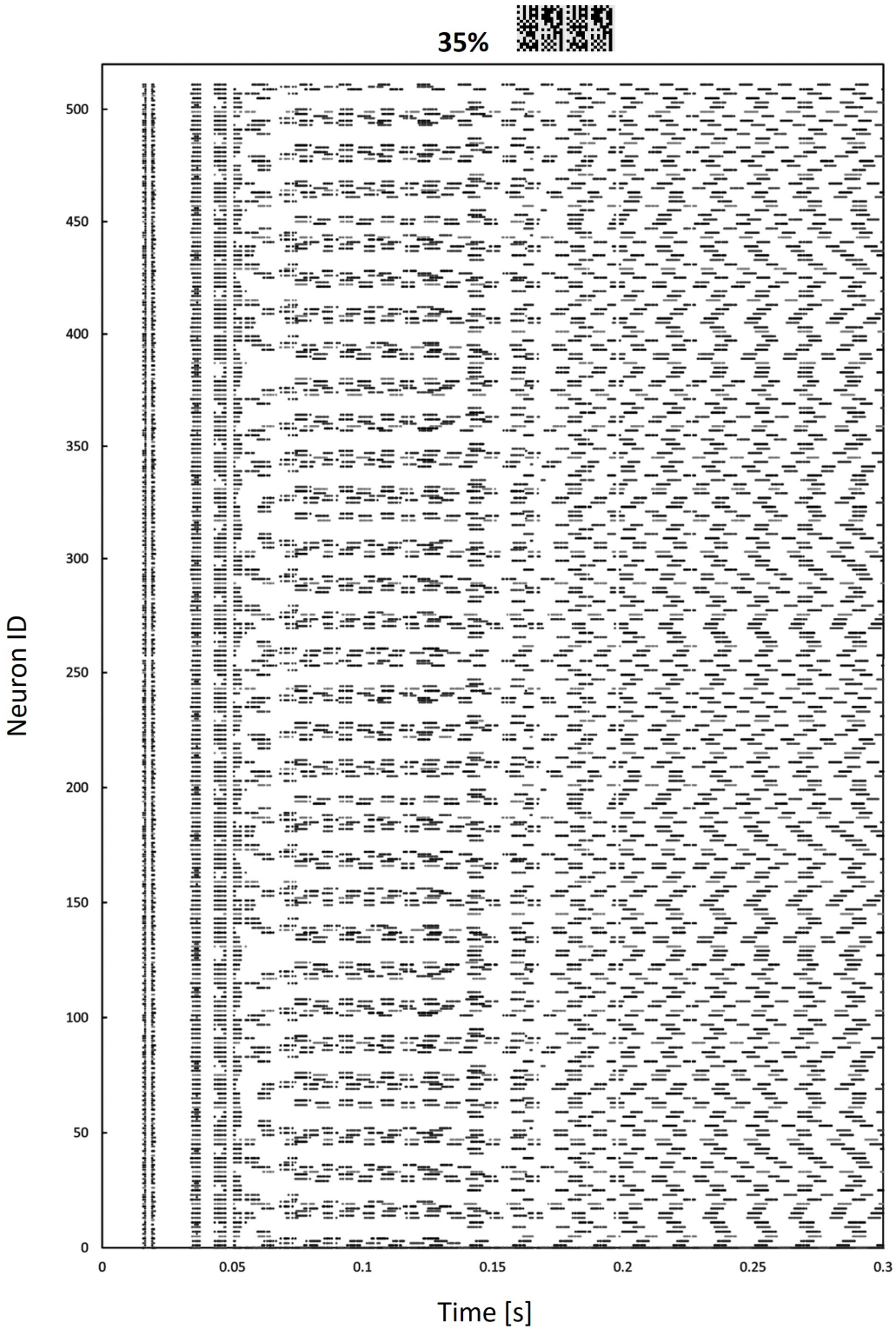

図 5.5　相関学習による連想記憶のニューロン発火 (入力パターンエラー 35%)

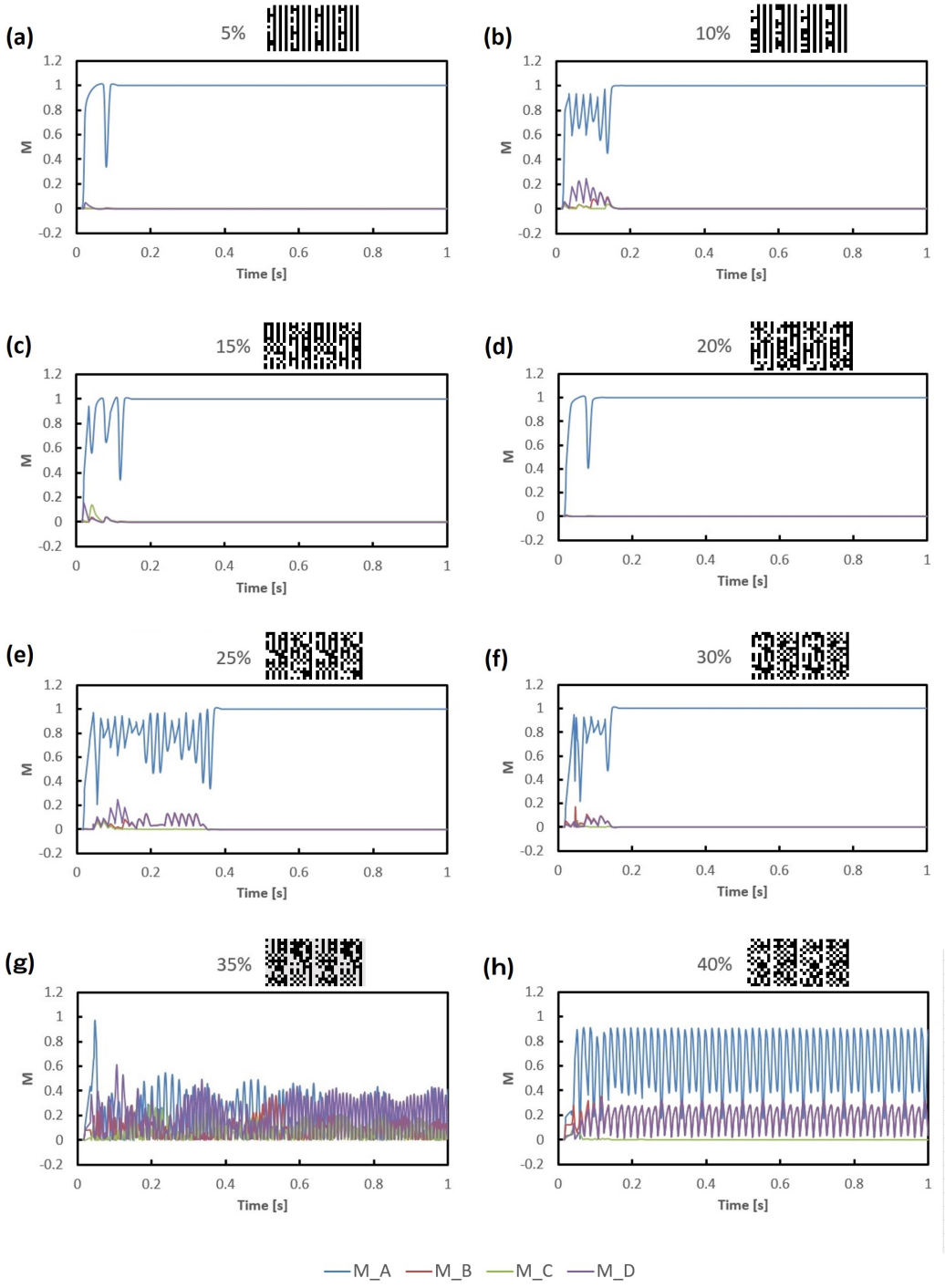

図 5.6　相関学習による連想記憶のオーバーラップ

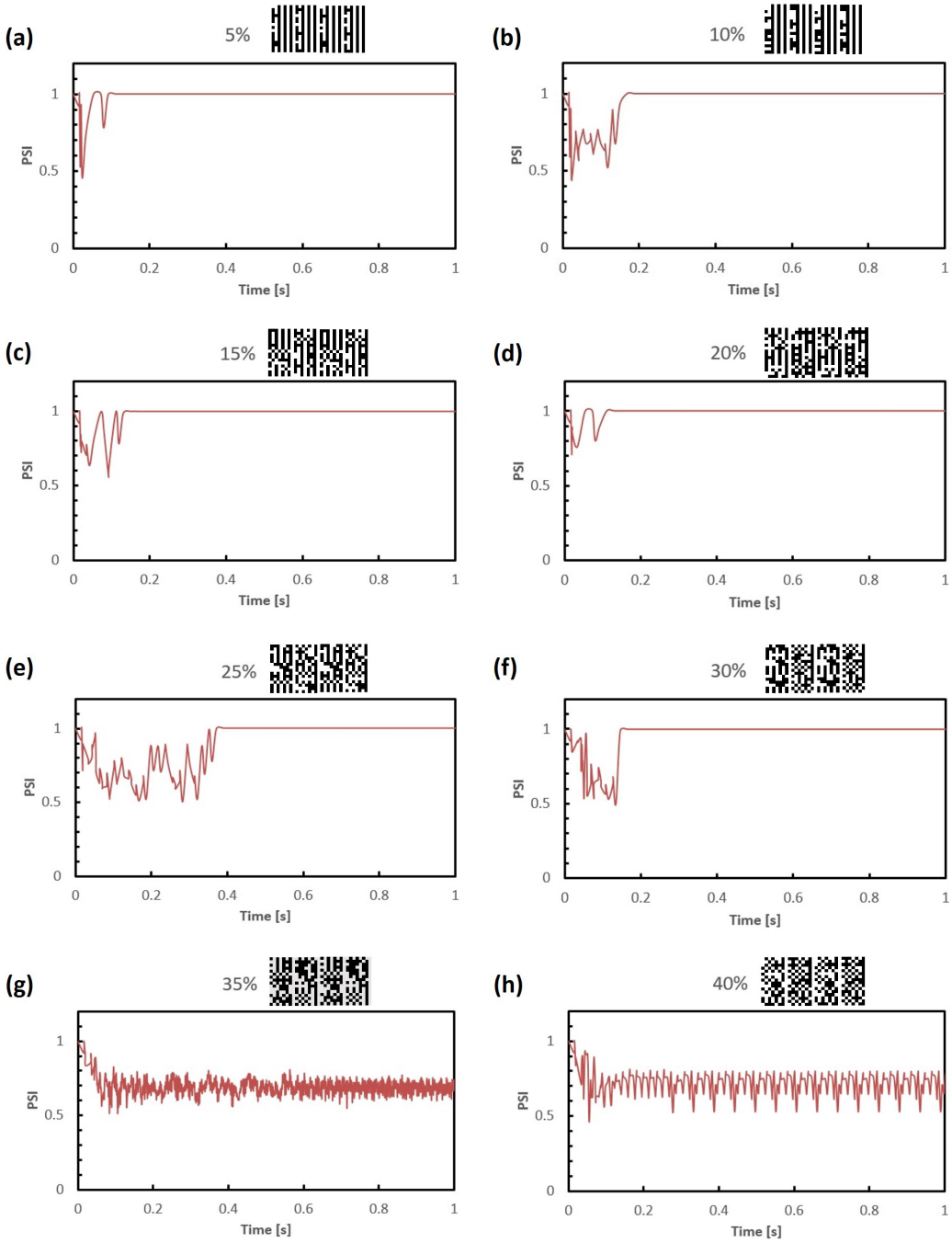

図 5.7　相関学習による連想記憶のニューロン同期

5.2　ヘブ則による連想記憶

5.2.1　ネットワークのコンフィギュレーション

ヘブ則による連想記憶では式 4.4 の CE=1 モードによって式 5.1 の W_{ij} が決まる。最初 CE=1 の学習モードによって W_{ij} が決まり、その後 CE=0 モードとして学習パターンを検出する。学習モードでは、学習パターンとその逆パターンが順番にニューラルネットワークに入力される。逆パターンはヘブ則による学習の効率を高めるために用いられた。異なるパターン対間のインターバルは 24ms である。24ms という時間は、ニューロンの不応期を避けるための時間である。学習パターンのピクセルは各ニューロンに相当し、ピクセルが黒の時にそのニューロンにパルス電流が与えられた。パルス電流は相関学習と同様に振幅は 0.0425 であり、6ms 間 0.0425 を、18ms 間 0 が与えられた。W_{ij} はゼロから始まり更新されていく。学習が終わった後、CE=0 モードとして学習パターンが検出された。その時の I_{stim} の入力は 5.1.1 節で説明した方法と同様である。

5.2.2　結果

まず通信プロトコルを介して接続している 512 ニューロンの全結合ネットワークにおいて図 5.8 の学習パターンを A と $\overline{A}$ を学習させ、その後パターン A と 30%異なるパターンを入力した。学習時のパターンの入れ方は $A, \overline{A}, A, \overline{A}, \cdots$ である。連想記憶の結果は相関学習による連想記憶と同様に ModelSim SE-64 10.1c から得られた。図 5.9 は 30%のエラーを持った入力パターンに対するニューロン発火の様子である。時刻 15.75ms に入力パターンが現れ、122ms から学習パターン A と $\overline{A}$ が決まった時間おきで順序に現れ始めた。図 5.10 の (a) と (b) は 30%の入力パターンに対するオーバーラップとニューロンの同期特性である。(a) も (b) も 151.635ms から 1 となり、パターンＡが検出されたことが確認できた。

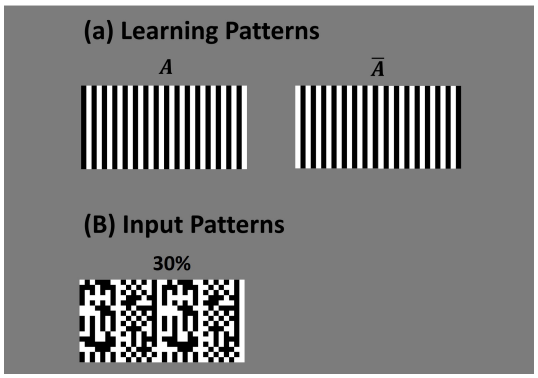

図 5.8　ヘブ則による連想記憶の学習パターン 1 つ (A) と入力パターン (B)

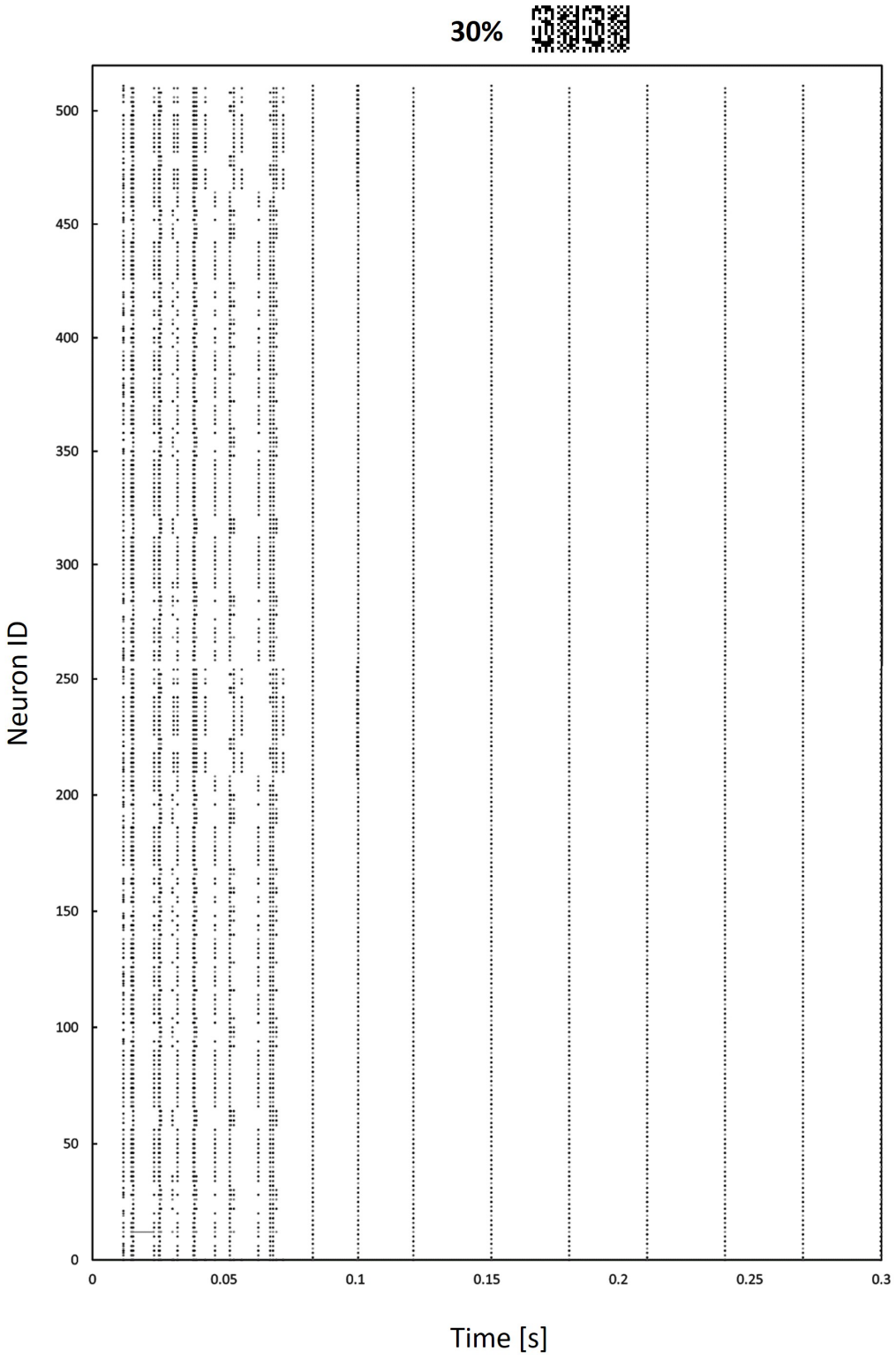

図 5.9　ヘブ則による連想記憶のニューロン発火 (入力パターンエラー 30%)

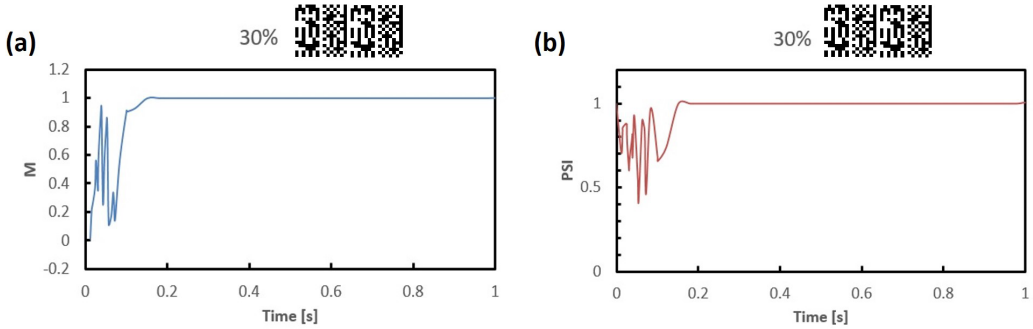

図 5.10　ヘブ則による連想記憶のオーバーラップ (a) とニューロン同期 (b)

次に通信プロトコルを接続していない 256 ニューロンの全結合ネットワークにおいて図 5.11 の学習パターン A、$\overline{A}$、B、$\overline{B}$ を学習させ、その後パターン A を入力パターンとして与えた。学習時のパターンの入れ方は $A, \overline{A}, A, \overline{A}, \cdots, B, \overline{B}, B, \overline{B}, \cdots$ である。同様に結果は ModelSim SE-64 10.1c から得た。図 5.12 の (a) は 0%のエラーを持った入力パターンに対するニューロン発火の様子である。図 5.12 の (a) からわかるようにパターン A もパターン B も現れなかった。それは図 5.12(b) のオーバーラップと (c) のニューロン同期特性が両方とも 1 にならないことからも分かる。従ってヘブ則によって 2 種類のパターンを学習させた時は、学習の段階でシナプス重みが正しく決まらなかったといえる。

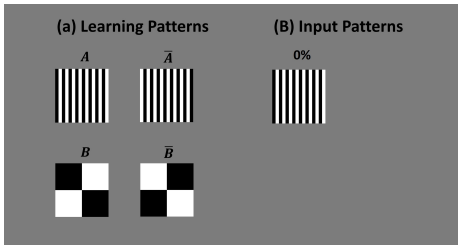

図 5.11　ヘブ則による連想記憶の学習パターン 2 つ (A) と入力パターン (B)

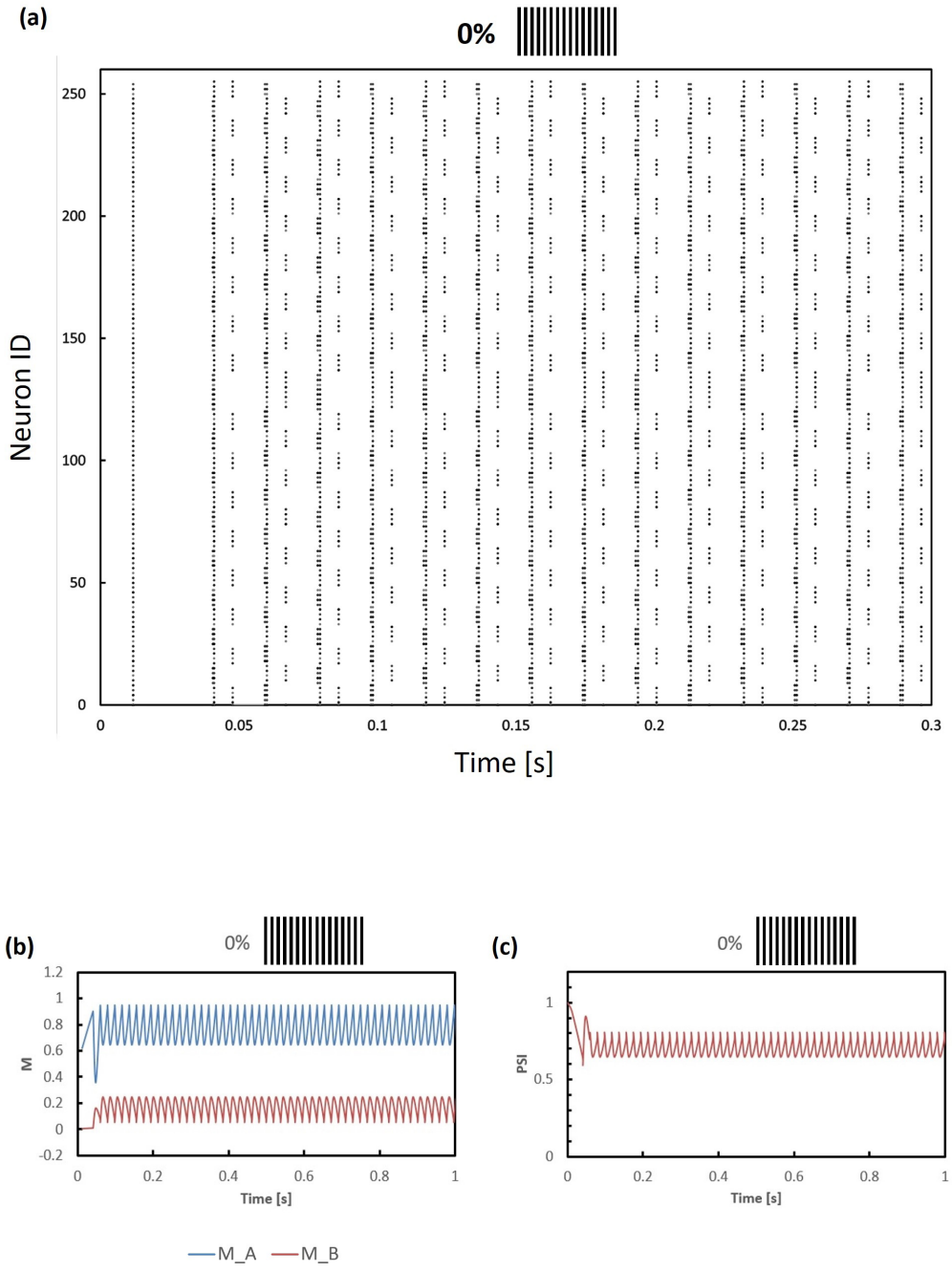

図 5.12　ヘブ則による連想記憶のニューロン発火 (入力パターンエラー 0%)(a)、オーバーラップ (b)、ニューロン同期 (c)

第6章　結論

本研究では大規模スパイキング神経ネットワーク構築のため、複数の FPGA スパイキング神経ネットワークを繋ぐ通信プロトコルを構築した。ニューロンモデルとしてはデジタル演算回路向けに考案された DSSN モデルを用い、シナプスモデルとしては動的モデルを用いた。スパイキング神経ネットワークの規模は 256 ニューロンの全結合ネットワークにした。膜電位やシナプス電流の計算リソース低減のため、並列及びパイプライン構造を用いた。FPGA 間の通信には AER 方式を用いた。異なる FPGA 間のニューロンを接続することは、ニューロンの生成するシナプス電流を伝達することでもあるため、AER 方式としてシナプス電流生成をコントロールするシグナル [T] とニューロン ID を他 FPGA に伝達した。シグナル [T] とニューロン ID はパケットとしてまとめて送信した。パケットを受け取ったネットワークは再びシナプス電流を生成し、刺激電流としてそれを受けとった。パケットの送信やシナプス電流と刺激電流の生成の一連の動作を通信プロトコルで行った。

本研究では 256^2 シナプスを持つ通信プロトコルを構築した。2 つの 256 ニューロンの全結合ネットワークを通信プロトコルを介して繋げ、512 ニューロンの全結合ネットワークを構築した。パケットが通信プロトコルを通る際にはディレイが生じるので、1024 ニューロンのネットワークが 128 個リング状に繋がっていると想定し、パケットが 128FPGA を通過するとき 108.6μs のディレイが生じることを確かめた。これは本研究のニューロンモデルの更新時間である 375μs より小さい値であって、1 回の更新ステップ以内に 128FPGA から生じる全てのパケットを受け取られることが確認できた。また、128×1024 ニューロンが同時に発火した時に要求される通信プロトコルのスループットが 16.7Gbps 以上であることも、高速シリアル通信プロトコルとして XILINX 社の AURORA を用いればその要件を満たせることも確認できた。

通信プロトコルを介して構築された 512 ニューロンの全結合ネットワークを用いて、連想記憶を試すことにより通信プロトコルの動作を確かめた。まず相関学習による連想記憶を行い、学習パターンと 30% まで異なる入力パターンに対してはパターンを思い出せることを確かめた。またヘブ則による連想記憶も試みた。

本研究では、スパイキング神経ネットワークの規模は 256 ニューロンの全結合ネットワークであり、通信プロトコルはそれに合わせた 256^2 シナプスを持つ規模であった。最終的に目指す大規模スパイキング神経ネットワークの規模は 1024 ニューロンの全結合ネットワークを 128 個繋げた規模であるため、今後は 1024 ニューロンと接続できる通信プロトコルを構築したい。また FPGA リソースの制限上、1024×128 ニューロンを全結合にすることが出来ないため、どの FPGA のど

のニューロンを接続させるかを示すデータシートを持たなければならない。このデータシートに従ってシナプス電流を生成し指定されたニューロンに刺激電流を与える機能も今後構築しなければならない機能である。

謝辞

本研究の一からはじめ、論文の作成まで丁寧なご指導いただきました河野先生に深く感謝いたします。また、研究室生活を問題なく進められるようサポートしてくださった合原研究室・河野研究室のメンバー、職員の皆様にも感謝いたします。

参考文献

[1] Takashi Kohno, “SIGNAL TRANSMISSION IN NEURONS,”. Mthematical Physiology, Unesco Encyclopedia of Life Support Systems, Submitted. in 6.188

[2] A. Destexhe, Z. F. Mainen, and T. J. Sejnowski, Kinetic models of synaptic transmission, In: Methods in Neuronal Modeling, 2nd ed. Mass.: MIT Press, Ed, C. Koch and I. Segev, pp. 1-25.(1999).

[3] W. Gerstner and W. M. Kistler, “Spiking neuron models. single neurons, populations,plasticity,” pp. 361-394, (2002).

[4] Paul A. Merolla,et al. ”A million spiking-neuron integrated circuit with a scalable communication network and interface” Science, 345 668 (2014).

[5] D. B. Thomas and W. Luk, “Fpga accelerated simulation of biologically plausible spiking neural networks,” IEEE FCCM, pp. 45-52, (2009).

[6] T. Kohno and K. Aihara, “Digital spiking silicon neuron: concept and behaviors in gj-coupled network,” Proceedings of International Symposium on Artificial Life and Robotics, vol. OS3-6, (2007).

[7] H. Tanaka, T. Morie, and K. Aihara, “A cmos spiking neural network circuit with symmetric/asymmetric stdp function,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E92-A, pp. 1690-1698, (2009).

[8] X. Jin, A. Rast, F. Galluppi, S. Davies, and S. Furber, “Implementing spike-timingdependent plasticity on spinnaker neuromorphic hardware,” The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, (2010).

[9] T. Dorta, M. Zapata, J. Madrenas, G. Snchez ”Aer-srt: scalable spike distribution by means of synchronous serial ring topology address event representation”. Neurocomputing (2015)

[10] H. Alle and R. P. J. Geiger, “Combined analog and action potential coding in hippocampal mossy fibers,” Science, vol. 311, pp. 1290?1293, (2006).

[11] Jing Li, Yuichi Katori and Takashi Kohno “An FPGA-based silicon neuronal network with selectable excitability silicon neurons”, Frontiers in neuroscience, 6, 183, (2012).

[12] Frontiers Jing Li, Yuichi Katori and Takashi Kohno “Hebbian Learning in FPGA Silicon Neuronal Network”, Proceedings of the 1st IEEE/IIAE International Conference on Intelligent Systems and Image Processing (2013).

[13] J. J. Hopfield, “Neural networks and physical system with emergent collective computational abilities,” Proc. Nat. Acad. Sci, vol. 79, pp. 2554-2558, (1982).

[14] E. Domany and H. Orland, “A maximum overlap neural network for pattern recognition,” Physics letters, vol. 125, pp. 32-34, (1987).

[15] M. Rosenblum, A. Pikovsky, J. Kurths, C. Schafer, and P. A. Tass, “Phase synchronization: from theory to data analysis. in handbook of biological physics,” pp. 279-321, 2001.