A Stochastic Spiking Neural Network for Virtual Screening

A. Morro, V. Canals, A. Oliver, M. L. Alomar, F. Galán-Prado, P. J. Ballester, and J. L. Rosselló

Abstract— **Virtual screening (VS) has become a key computational tool in early drug design and screening performance is of high relevance due to the large volume of data that must be processed to identify molecules with the sought activity-related pattern. At the same time, the hardware implementations of spiking neural networks (SNNs) arise as an emerging computing technique that can be applied to parallelize processes that normally present a high cost in terms of computing time and power. Consequently, SNN represents an attractive alternative to perform time-consuming processing tasks, such as VS. In this brief, we present a smart stochastic spiking neural architecture that implements the ultrafast shape recognition (USR) algorithm achieving two order of magnitude of speed improvement with respect to USR software implementations. The neural system is implemented in hardware using field-programmable gate arrays allowing a highly parallelized USR implementation. The results show that, due to the high parallelization of the system, millions of compounds can be checked in reasonable times. From these results, we can state that the proposed architecture arises as a feasible methodology to efficiently enhance time-consuming data-mining processes such as 3-D molecular similarity search.**

Index Terms— **Data mining, neuromorphic hardware, spiking neural networks (SNNs), virtual screening (VS).**

I. Introduction

Data explosion is related to the huge capacity of data generation implemented by current technologies at different science fields, where data volume doubles every year [1]. According to recent studies, it is predicted that the volume of such data will become 26 fold in the next five years [2]. However, this data explosion has not led to a comparable information explosion, since data analysis methods are unable to manage billions of data records in reasonable times. Different data-mining methodologies have been developed based on the use of artificial neural networks (ANNs) [3], or simplified data set extractions from original data [4]. In this context, drug discovery is one of the fields in which the volume of information available for analysis has increased the most. To analyze this huge amount of information, a set of computational techniques have been developed; among them, virtual screening (VS) [5] stands out as a data-mining technique intended to predict which of the studied molecules neutralize the function of a disease-causing protein by binding to it. There are two main VS categories, structure-based and ligand-based, depending on whether a protein structure or a ligand molecule is used as the basis of the prediction. In ligand-based VS, a search for molecules with similar properties to that one used as a template is performed (this strategy is based on the principle that molecules with similar properties tend to have similar activities on the same proteins). One common application is the identification of nonobvious compounds that retain the desired biological activity of the query molecule acting as search template, but which are devoid of its disadvantages (e.g., being patented or toxic). Ligand-based VS by molecular shape similarity is naturally suited for this task, as different chemical scaffolds may support similar molecular shapes. Conceptually, molecular shape comparison is also attractive given the central role of shape-complementarity in molecular recognition events as an important indicator of a molecule's biological activity. Indeed, without such complementarity, the ligand and receptor atoms involved in binding would not be sufficiently close to allow favorable interactions, such as hydrogen bonds and ionic interactions. For instance, by using a molecule with affinity for a phosphatase target and searching for similarly shaped molecules in a large database of molecules, one can identify new phosphatase inhibitors [6].

Due to the exponential size increment experimented in molecular databases; there is a limitation to the use of shape recognition methods. The expansion of searchable data is mainly motivated by our desire to cover a wider region of the biologically relevant chemical space, which is extremely large [7], and thus improve the likelihood of finding innovative drug leads that otherwise would not even be included in the search. Consequently, it is of great importance to screen a molecular database as fast as possible.

In this brief, we address a neural network implementing a total of 99.200 equivalent spiking neurons that is able to increase the efficiency of a 3-D shape similarity technique known as ultrafast shape recognition (USR) [8]. USR is already a particularly fast VS technique [9] and its application has resulted in the discovery of molecules with previously unknown activity against a range of molecular and cell targets [6], [10]–[13].

A feasible technique for VS is the use of ANNs that is one of the fields of computational intelligence that has evolved the most over the last three decades. ANNs have been widely used for classification and regression [14] due to their outstanding characteristics of self-adaptability, self-organization, and real-time learning capability [15]. ANNs started with the first generation based on McCulloch–Pitts neurons as computational units [16] and are currently in the third generation [17] that introduces the use of spiking neurons as computational units. These are bioinspired descriptions that make use of sequences of delta functions to emulate the action potential of biological neurons. Recently, we introduced a variation of spiking neurons, the stochastic spiking neural networks (SSNNs) scheme. The main characteristic of SSNN is the facility to be implemented using digital gates [18]. The SSNN consists in a stochastic hardware implementation of spiking neurons using low resources in terms of logic gates that enable their use to create highly parallelized systems [19]. At the same time, SSNN is highly bioinspired since a key property of the spike trains measured in real neurons is their seemingly stochastic or random nature [20]. The stochastic nature of spike trains is in part due to the mechanism of synaptic transmission since each synaptic vesicle releases its *"quantum"* of transmitter from the neuron presynaptic terminal with a given probability [21].

This brief is the first real-life application of SSNNs [18], [19]. In particular, the proposed implementation is used to provide an

Manuscript received May 2, 2016; revised September 2, 2016; accepted December 31, 2016. Date of publication February 7, 2017; date of current version March 15, 2018. This work was supported in part by the Spanish Ministry of Science and Technology, the Regional European Development Funds (FEDER) under EU Projects TEC2011-23113 and TEC2014-56244-R, and in part by the French Government "Investissements d'Avenir" Program through A*MIDEX under Grant ANR-11-IDEX-0001-02.

A. Morro, V. Canals, A. Oliver, M. L. Alomar, F. Galán-Prado, and J. L. Rosselló are with the Electronics Engineering Group, Department of Physics, University of Balearic Islands, 07122 Palma de Mallorca, Spain (e-mail: j.rossello@uib.es).

P. J. Ballester is with the Cancer Research Center of Marseille, Institut Paoli-Calmettes, Aix-Marseille Université, F-13284 Marseille, France.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2657601

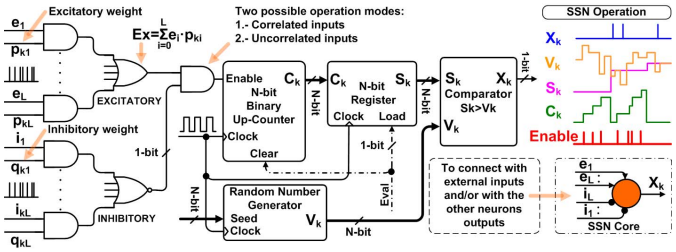

Fig. 1. Stochastic spiking neuron: digital block and schematic symbol. The color of the symbol is related to the v_k generator seed used.

important speed-up to VS by implementing the USR method to huge databases, where billions of molecular compounds must be screened in reasonable times in the research of new drug candidates. In Section II, we briefly explain the VS methodology to be implemented along with a detailed explanation of the SSNN architecture used. Then, we present the results when comparing the hardware and the commercial USR software. Finally, we conclude this brief.

II. Materials and Methods

Most of the data-mining processes implement a database similarity search in their core. Similarity search in huge databases in the context of ligand-based VS is a time-consuming processing task. This is due to the complexity of the molecular characteristics to discover (as their binding capacity by a given molecular target, which highly depends on the specific target). This is the reason for which the achievement of a molecular description data set optimum for hierarchical searches is complex to obtain. Therefore, due to the prohibitive computation time needed for this kind of searches it is of high importance to develop new high-speed computing techniques. USR is based on the observation that the shape of a molecule is uniquely determined by the relative positions of its atoms. This 3-D spatial arrangement of atoms is in turn accurately described by a set of distributions of interatomic distances. This convenient representation directly eliminates any need for the superposition of the compared molecules, as the resulting set of distances is independent of orientation or position. A highly concise encoding of molecular shape, 12 real-valued numbers per molecule, is ultimately achieved by characterizing each of the four distributions of atomic distances by their first three statistical moments. The shape similarity of two molecules is lastly calculated comparing those 12 parameters. Full details of the algorithm can be found in [9].

A. Stochastic Spiking Neural Networks

SSNNs represents a low-cost methodology to implement SNNs using digital gates. In Fig. 1, we provide the schematic of the stochastic spiking neurons used in this brief, equipped with a total of L excitatory (e) and L inhibitory (i) inputs ($e_{k1}, \ldots, e_{kL}; i_{k1}, \ldots, i_{kL}$). All the signals in Fig. 1 are of a single bit except c_k, s_k, and v_k signals. The input pulses (action potentials) present a switching activity that is transmitted to the soma of the kth neuron with probabilities p_{ki} and q_{kj} for both excitatory and inhibitory signals, respectively. Finally, all the incoming excitatory and inhibitory signals are added using an OR and a NOR gate, respectively (performing the union function to the input pulses) that are posteriorly joined with an AND gate (performing the intersection function between signals). The resulting signal is connected to a digital counter (providing a count c_k) that can be understood as being the *soma* of the neuron. The model does not include a continuous leak and feedback, and the counter is resettled with signal *Eval* in regular times (period $T_{\text{Eval}} = N \times T_{\text{CLK}} >> T_{\text{CLK}}$). The *Eval* signal also habilitates a register to store the total count along this period (output s_k). The parameter s_k is posteriorly compared with a threshold value v_k that is randomly generated by a specific digital block (typically a linear feedback shift register). The output of the kth neuron (bit signal x_k) is the result of this comparison (see Fig. 1) so that

$$x_k = \theta(s_k - v_k) \tag{1}$$

where $\theta(x)$ is the Heaviside function. In Fig. 1, we also show the schematic symbol that we use to represent the stochastic neuron where the color of the neuron is related to the specific threshold used. The neuron threshold $v_k(t)$ is a random signal with mean value $\langle v_k \rangle$ that is inserted to provide a more realistic behavior to the model since it has been reported that spike trains measured in real neural systems mainly present a stochastic or random nature so that different neurons with exactly the same input conditions provide at its outputs signals with completely different timings. From the behavioral point of view, the principal functionality of such a variable threshold is to synchronize or uncorrelate different neurons. Therefore, we say that two neurons are synchronized if they share the same threshold; otherwise, the neurons are uncorrelated.

B. Massive Data-Mining With SSNN

In a previous work [19], we demonstrated that correlated and uncorrelated spike signals can be combined in an SSNN for the implementation of complex processing tasks such as fast pattern recognition. In Fig. 2(a), we show a generalization of the example studied in [19] where a total of L neural ensembles are configured to recognize which vector provided by a database $db_j = (db_{j1}, \ldots, db_{jm})$, $j \in (1, \ldots, L)$ is the nearest to a fixed reference vector $r = (r_1, \ldots, r_m)$. Each neural ensemble input inhibits the activity of an output neuron that is stimulated by an external signal (b). The output activity of each ensemble vanishes abruptly when the compared vectors are equal $db = r$, thus maximizing activity s_{jr}.

These results are only possible when combining both correlated and uncorrelated neurons [in Fig. 2(a), correlated neurons are described with identical colors, while different colors imply decorrelation]. To clarify this, consider the circuit in which a signal (c) is evaluated from two inputs (a, b) using an OR (or an AND) gate. In these two cases, the output activity can be obtained from the union (or the intersection for the AND case) of both input signals during all times and $c = a \cup b$ (or $c = a \cap b$). For completely correlated signals (that we denote as $a||b$), we can simplify $a \cup b = \max(a, b)$ and $a \cap b = \min(a, b)$. For uncorrelated signals ($a \perp b$), we have that $a \cup b = a + b - ab$ and $a \cap b = ab$. For a more proper definition of correlated and uncorrelated signals, we define the independence factor between signals (a, b) as

$$I(a, b) \equiv \frac{a \bigcup b - \max(a, b)}{min(a, b) - ab} \tag{2}$$

where a and b are referred to the switching activity of two desired neural signals, and $a \cup b$ the switching activity of the resulting join signal through an OR gate. From (2), we state that two signals are correlated ($a||b$) or uncorrelated ($a \perp b$) using the next rules

$$\begin{aligned} a \parallel b &\text{ if and only if } I(a, b) = 0 \\ a \perp b &\text{ if and only if } I(a, b) = 1. \end{aligned} \tag{3}$$

It is important to note that the output signals of synchronized and uncorrelated neurons (as was defined previously depending if they share the v_{th} signal or not) are correlated ($I = 0$) and uncorrelated ($I = 1$) signals, respectively. In a more general case, the neural

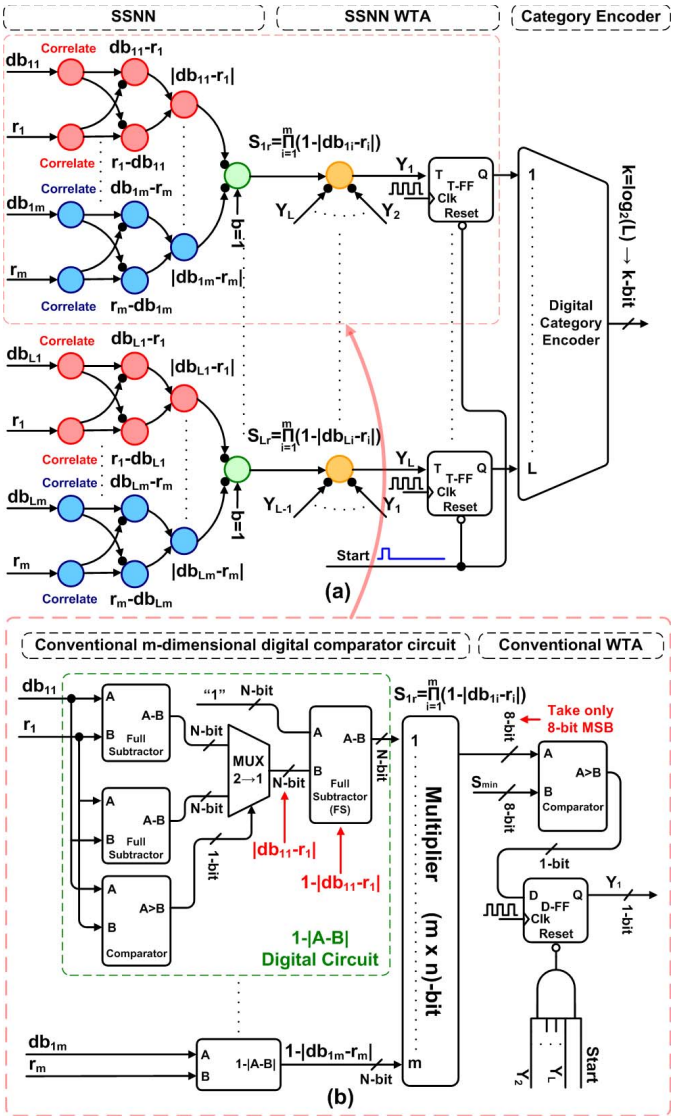

Fig. 2. (a) Neural architecture to compare "L" m-dimensional vectors in parallel. Arrow (circle) represents an excitatory (inhibitory) connection. (b) Conventional digital architecture to compare "L" m-dimensional vectors in parallel.

signals are joined or colliding inside the SSNN following the next general expressions:

$$a \cup b = \max(a, b) + (\min(a, b) - ab)I(a, b) \quad (4)$$

$$a \cap b = \min(a, b) - (\min(a, b) - ab)I(a, b) \quad (5)$$

where expression (4) come from definitions (2) and (5) considering that $a \cap b = a + b - a \cup b$.

For the case of each correlated cluster of Fig. 2(a), $I = 0$ for all cases and the function implemented at the output of the ensemble is

$$(db_{ji} \cap \overline{r_i}) \cup (r_i \cap \overline{db_{ji}}) = \max(db_{ji} - r_i, r_i - db_{ji}). \quad (6)$$

Expression (6) is the absolute value $|db_{ji} - r_i|$.

In particular, this ensemble of correlated neurons [Fig. 2(a)] consists of a first layer of stochastic spiking neurons (model previously described) used to correlate the signals to be compared (component i of vectors db and r). A second layer evaluates in parallel $(db_{ji} - r_i)$ and $(r_i - db_{ji})$, and finally, a third layer performs the function $\max\{(db_{ji} - r_i), (r_i - db_{ji})\} = |r_i - db_{ji}|$. In Fig. 2(a), the fourth layer of neurons providing the signals s_{jr} performs the product of all the uncorrelated incoming signals (that is, the collision probability of input spikes). For the evaluation of s_{jr}, we must consider that $I(x_i, x_k) = 1$ for any pair of signals coming from different clusters. Therefore, the circuit is able to compare m-dimensional vectors by superposing different uncorrelated clusters, thus creating a vector comparator kernel. The spiking signal present at the output of each kernel is an estimation of the similarity (s_{jr}) of two objects (reference vector r, and a database vector defining the category j of db_j). Parameter s_{jr} is a switching bit with a firing activity proportional to the similarity between both objects (vector r and the category j of db_j). Vector r can be considered to be a known object with well-defined properties. In the case of the implementation of the USR method, r represents a compound with known biological activities (as an example, a commercial drug). In the SSNN circuit, each kernel excites a neuron with output signal (y_j), constituting the fifth layer of the SSNN, that inhibits all the other outputs $y_k (k \neq j)$ that are connected to other kernels, thus implementing a winner-take all (WTA) structure. All these output neurons are correlated and therefore share the same threshold v_{th}. With this scheme, hundreds of kernels can be combined in parallel to implement a similarity comparator in one single chip. Taking advantage of the fact that the action potential timing follows the probabilistic laws [19], and assuming that $b = 1$ for simplicity, the switching activity of signal s_{jr} is found to be

$$s_{\text{jr}} = \prod_{i=1}^{m} (1 - |db_{ji} - r_i|). \quad (7)$$

The switching activity of s_{jr} is maximal when db_j is similar or equal to r. When diverge both vectors, then the switching activity s_{jr} tends to zero quickly. Hundreds of kernels can be connected in parallel in a medium-sized field-programmable gate array (FPGA), thus increasing considerably the mining speed with respect to the processor-based techniques. Therefore, hundreds of vectors can be compared at the same time by using the metric shown in (7). For a reference vector r, the circuitry provides at its output the closest database vector db_j.

We define v_{th} as the fixed threshold selected for the output neurons that is kept constant. Therefore, v_{th} is the minimum number of pulses (action potentials) needed to activate each output neuron (y_j) (excited by s_{jr}). Since only a fixed number of time steps (N) for each vector comparison is implemented, a minimum similarity value to be distinguished is therefore fixed (switching activity $s_{\min}$ so that $N \times s_{\min} = v_{\text{th}}$). If all the similarities at the input of the WTA [actions potentials with switching activities s_{jr} in Fig. 2(a)] are lower than $s_{\min}$, then the more probable scenario is to obtain a negative result at the output of the network. At the WTA output, the neuron j that is activated is the one with the higher similarity value

$$\text{Category} = j | s_{jr} \geq s_{ir} \quad \forall i \in \{1..L\} \quad (8)$$

where s_{jr} is defined by (7). When two vectors ($\boldsymbol{db_j}$ and $\boldsymbol{r}$) present a similarity such that $s_{\text{jr}} > s_{\min}$, the probability of identifying vector $\boldsymbol{r}$ as belonging to class j is close to 1. From a practical point of view, the system incorporates a *T flip-flop* at each category output to store the most likely category.

III. Experimental Results

A. Similarity Search Platform

A hardware/software platform to perform similarity searches is shown in Fig. 3. We have used a GIDEL *Procstar IV* [22] card that communicates with the software through the PCIe bus. This board incorporates four FPGAs (Altera *Stratix III*), each one drives two DDR2 SO-DIMM memory banks (2 × 4 GB External + 2 × 256 MB

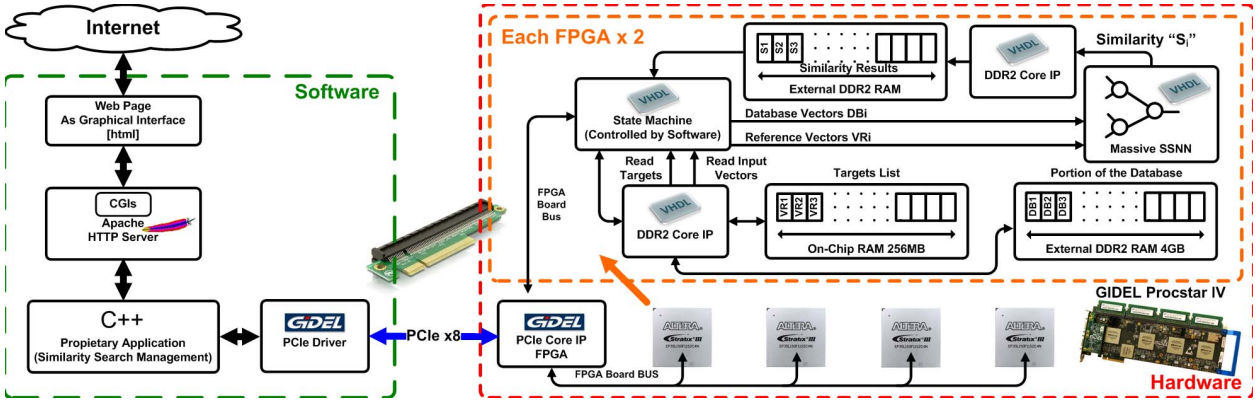

Fig. 3. Similarity search hardware/software platform architecture.

TABLE I
QUERY RESULTS FOR THREE DIFFERENT REFERENCE COMPOUNDS

ZINC13410708_1	Score	ZINC12595305_1	Score	ZINC01994557_1	Score
ZINC13410708_1	1	ZINC12595305_1	1	ZINC1994557_1	1
ZINC48504579_169	0.95471	ZINC43840584_167	0.95501	ZINC4858268_6	0.9736
ZINC46532970_144	0.95279	ZINC43692858_137	0.95142	ZINC6536772_103	0.9692
ZINC42309305_141	0.94814	ZINC48107549_160	0.95046	ZINC9950459_14	0.9604
ZINC3397121_73	0.94803	ZINC46326243_195	0.94817	ZINC3407017_95	0.9603
ZINC49199347_86	0.94546	ZINC48107546_140	0.94648	ZINC29371413_43	0.9544
ZINC42210855_152	0.94457	ZINC25261028_87	0.94617	ZINC21036375_21	0.9543
ZINC1426614_92	0.9421	ZINC48415497_143	0.94564	ZINC15441957_145	0.9533
ZINC4865820_28	0.94169	ZINC45085953_36	0.94352	ZINC47340811_147	0.9517
ZINC22630929_36	0.94096	ZINC43271229_95	0.94265	ZINC4945399_62	0.9514
ZINC13690268_87	0.9408	ZINC1013722_149	0.94208	ZINC42486060_13	0.9499
ZINC25995941_167	0.94076	ZINC47730885_77	0.94183	ZINC12574037_40	0.9498
ZINC19595114_36	0.94071	ZINC42020063_89	0.94118	ZINC16459430_74	0.9498
ZINC2314037_63	0.93985	ZINC29957530_34	0.94105	ZINC7450868_153	0.9496
ZINC11406227_71	0.93946	ZINC41247629_109	0.94075	ZINC28849869_22	0.9492
ZINC1414978_18	0.93921	ZINC29957481_25	0.9407	ZINC41608402_130	0.9488
ZINC48040704_200	0.9383	ZINC46326241_81	0.93917	ZINC28849871_25	0.9485
ZINC8686258_62	0.93823	ZINC46485111_55	0.93878	ZINC49032117_14	0.9483
ZINC41870989_44	0.93813	ZINC44114773_131	0.93696	ZINC49180253_3	0.948
ZINC250216_100	0.93806	ZINC43271247_84	0.93695	ZINC42652216_2	0.9479
ZINC48040707_103	0.93803	ZINC46744477_73	0.93676	ZINC5628818_39	0.9477
ZINC1125794_107	0.93801	ZINC12595303_2	0.93653	ZINC312169_16	0.9473
ZINC49199337_87	0.93731	ZINC1567201_132	0.93644	ZINC32827648_49	0.947
ZINC14776979_23	0.93728	ZINC47786879_28	0.93612	ZINC291424_40	0.9467
ZINC49199295_92	0.93676	ZINC29957532_39	0.93609	ZINC15774821_43	0.9465
ZINC48984236_175	0.93667	ZINC12595305_135	0.93509	ZINC40696620_176	0.9465
ZINC47316350_96	0.9364	ZINC16965782_71	0.93436	ZINC48653383_200	0.9463
ZINC2201000_29	0.93616	ZINC43271248_21	0.93392	ZINC47340811_46	0.9462
ZINC49469411_37	0.93515	ZINC29957483_23	0.93382	ZINC42652220_5	0.9459
ZINC48033446_170	0.93507	ZINC48699174_41	0.93372	ZINC9376981_35	0.9458
ZINC38141915_161	0.93499	ZINC30039243_141	0.9328	ZINC12042794_16	0.9457
ZINC32736486_52	0.93494	ZINC47730718_5	0.93255	ZINC48531002_143	0.9457
ZINC49224939_164	0.93478	ZINC42089877_102	0.93247	ZINC8245049_3	0.9456
ZINC48195870_143	0.9347	ZINC12855931_29	0.93205	ZINC14116623_196	0.9453
ZINC11238859_192	0.93464	ZINC12390472_160	0.93177	ZINC26571595_185	0.9452
ZINC46350341_42	0.93448	ZINC5398753_129	0.93172	ZINC10342308_30	0.9449
ZINC31305889_69	0.93446	ZINC41813471_168	0.9308	ZINC518765_45	0.9448
ZINC43069711_61	0.93441	ZINC31636514_171	0.93048	ZINC6063561_193	0.9447
ZINC46560083_181	0.93419	ZINC48667988_98	0.93025	ZINC16953027_180	0.9447
ZINC31528041_94	0.93406	ZINC29943919_113	0.93014	ZINC518766_44	0.9444

on Board), in order to read/write the database. At each FPGA, we implement 24 800 equivalent neurons working at 87.5 MHz ($v_{\text{th}} = 8$). The setup presents a power of the order of 100 W. A similar platform is used in [23] but using probabilistic logic.

B. Similarity Search Results

We have applied the proposed architecture shown in Fig. 2(a) to perform similarity searches to huge USR-based molecular databases. Each compound is represented by a 12-D vector (so that $m = 12$). Since each component in the SSNN is represented by the activity of one single neuron (bounded between 0 and 1), we normalize the USR descriptors with respect to their higher variation. To compare the performance of hardware and software versions of USR, we used a database with 147.328.000 3-D conformers derived from 3.5 million purchasable molecules retrieved from the ZINC database [24]. In Table I, we show the results of the top 40 similar molecules in three different searches (with respect compounds: ZINC13410708, ZINC12595305, and ZINC01994557). In gray, we identify those compounds also found in the top-200 using software. The time per search is 0.5 s (300 million of comparisons per second). In contrast, the software implementation of USR running on eight cores of an Intel Core i7-2920XM (CPU at 2.50 GHz, 16 GB RAM) completed the same run in 122 s. Finally, in Fig. 4, we show the correlation between Hardware and Software when they are ranked from the best to the worst fit. Only a small deviation from the identity (red line) is observed.

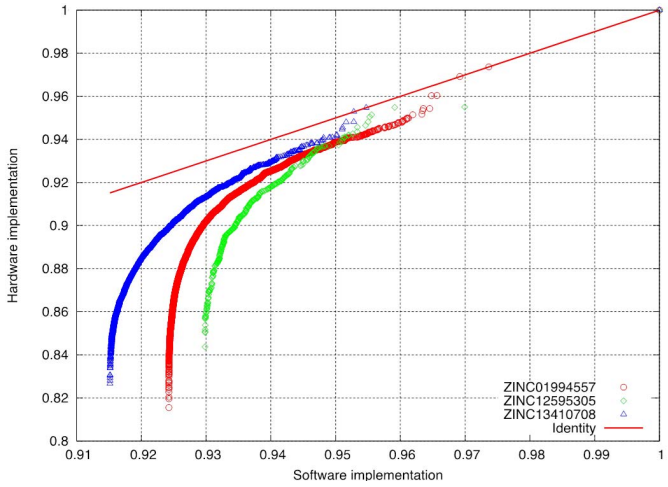

Fig. 4. Relationship between the molecular similarities found with the Hardware implementation in front of the molecules found by software, for three different queries.

C. Comparison With a Purely Digital Implementation

In Fig. 2(b), we show the classical digital circuitry that implements the comparative metric (7). In Fig. 5, we compare both implementations in terms of FPGA resources using an Altera *Stratix III* EP3SE110 (*ProcStar IV 110E-4B* FPGA board welded devices). As can be appreciated, the ratio of logic elements needed by the conventional and SSNN implementation increases as the number of vectors to be compared grows. The comparison shows that the proposed architecture is occupying up to 60 times less circuit area. The speed improvement of the proposed architecture is of the order of 8.7 times in terms of memory screening speed with respect to the classical.

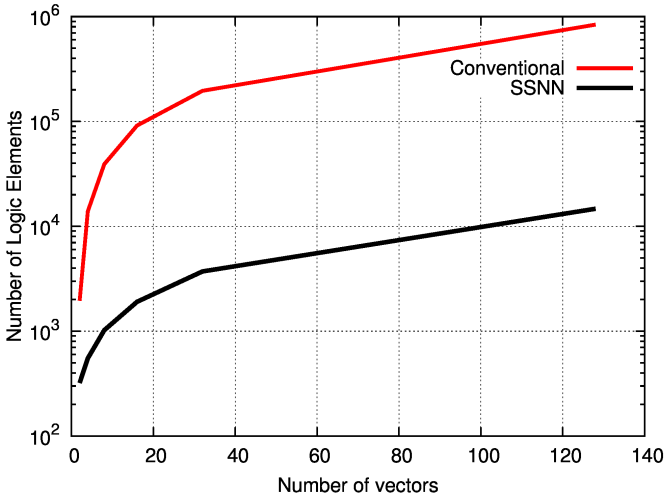

Fig. 5. Comparison between the conventional implementation and the proposed SSNN, in terms of FPGA hardware resources.

IV. Conclusion

Pattern recognition is probably the most fundamental brain process. Some works suggest that the fast pattern recognition processing observed in mammalian brains can only be explained using a delay-based codification of action potential timing [25] since the pattern match in the brain is in the order of 100–200 ms, while biological neurons oscillates with a typical period of the order of 10 ms. This means that the process is completed typically in 10–20 time steps. Therefore, a coding scheme based on firing rates is traditionally considered to be unfeasible for pattern recognition. Nevertheless, a complex delay-based codification seems to require a complex spatiotemporal wiring and learning. In this brief, we proposed a bioinspired neural architecture in which synchronization is able to provide fast pattern recognition by simply using a firing rate coding. This process is quick enough to explain the fast pattern recognition observed in biological experiments. The proposed model is almost based in the use of parallelism and the combination of synchronized and uncorrelated signals.

We also show how a bioinspired spiking neural model implemented in digital hardware (that we define as SSNNs) can be applied to fast pattern matching problems. We also provide an explanation in terms of synchronized/uncorrelated signals and use a specific metric to estimate the degree of synchronization between neural signals. We applied the proposed architecture to the drug discovery field in which huge molecular databases must be screened continuously in the research of new compounds that may present biological activity and therefore can be considered as new drug candidates. The results show an important increase of the database screening speed compared with standard CPU-based systems (about two orders of magnitude) and of classical hardware implementations (a factor 8.7). This speed improvement is due to the high degree of parallelization achieved by the neural network and also due to the fact that an SSNN is performing an approximated processing rather than exact binary computations (that requires more resources in the classical implementation). Note that high precision is not a mandatory requirement to perform an efficient pattern-matching search inside huge databases. The main disadvantage of the system is the precision limitation, since a set of false positives and negatives is obtained due to the random nature of the SSNN. Nevertheless, the results show that the loss of accuracy is compensated by speed gain.

References

[1] A. Szalay and J. Gray, "2020 computing: Science in an exponential world," *Nature*, vol. 440, no. 7083, pp. 413–414, Mar. 2006.

[2] A. Osman, M. El-Refaey, and A. Elnaggar, "Towards real-time analytics in the cloud," in *Proc. IEEE 9th World Congr. Services*, Santa Clara, CA, USA, Jun. 2013, pp. 428–435.

[3] J. Zou, Y. Han, and S.-S. So, "Overview of artificial neural networks," in *Artificial Neural Networks* (Methods in Molecular Biology), vol. 458, D. J. Livingstone, Ed. Totowa, NJ, USA: Humana Press, 2009, pp. 14–22.

[4] F. Pan Wei Wang, A. K. H. Tung, and J. Yang, "Finding representative set from massive data," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Sep. 2005, pp. 8–15.

[5] G. Schneider, "Virtual screening: An endless staircase?" *Nature Rev. Drug Discovery*, vol. 9, no. 4, pp. 273–276, Apr. 2010.

[6] B. Hoeger, M. Diether, P. J. Ballester, and M. Köhn, "Biochemical evaluation of virtual screening methods reveals a cell-active inhibitor of the cancer-promoting phosphatases of regenerating liver," *Eur. J. Med. Chem.*, vol. 88, pp. 89–100, Dec. 2014.

[7] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "enumeration of 166 billion organic small molecules in the chemical universe database GDB-17," *J. Chem. Inf. Model.*, vol. 52, no. 11, pp. 2864–2875, Nov. 2012.

[8] P. J. Ballester and W. G. Richards, "Ultrafast shape recognition for similarity search in molecular databases," *Proc. Roy. Soc. London A, Math. Phys. Eng. Sci.*, vol. 463, no. 2081, pp. 1307–1321, May 2007.

[9] P. J. Ballester, "Ultrafast shape recognition: Method and applications," *Future Med. Chem.*, vol. 3, no. 1, pp. 65–78, Jan. 2011.

[10] P. J. Ballester, I. Westwood, N. Laurieri, E. Sim, and W. G. Richards, "Prospective virtual screening with ultrafast shape recognition: The identification of novel inhibitors of arylamine *N*-acetyltransferases," *J. Roy. Soc. Interface*, vol. 7, no. 43, pp. 335–342, Dec. 2009.

[11] P. J. Ballester *et al.*, "Hierarchical virtual screening for the discovery of new molecular scaffolds in antibacterial hit identification.," *J. Roy. Soc. Interface*, vol. 9, no. 77, pp. 3196–3207, Dec. 2012.

[12] C. Y. Teo *et al.*, "Discovery of a new class of inhibitors for the protein arginine deiminase type 4 (PAD4) by structure-based virtual screening," in *Proc. 11th Int. Conf. Bioinformat. (InCoB)*, Bangkok, Thailand, Oct. 2012, pp. S4.

[13] S. P. Patil, P. J. Ballester, and C. R. Kerezsi, "Prospective virtual screening for novel p53–MDM2 inhibitors using ultrafast shape recognition," *J. Comput. Aided Molecular Design*, vol. 28, no. 2, pp. 89–97, Feb. 2014.

[14] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Burlington, MA, USA: Academic, 2009, pp. 151–249.

[15] R. Tawel, "Learning in analog neural network hardware," *Comput. Electr. Eng.*, vol. 19, no. 6, pp. 453–467, Nov. 1993.

[16] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.

[17] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, Aug. 2009.

[18] J. L. Rossello, V. Canals, A. Morro, and A. Oliver, "Hardware implementation of stochastic spiking neural networks," *Int. J. Neural Syst.*, vol. 22, no. 4, p. 1250014, Aug. 2012.

[19] J. L. Rossello, V. Canals, A. Oliver, and A. Morro, "Studying the role of synchronized and chaotic spiking neural ensembles in neural information processing," *Int. J. Neural Syst.*, vol. 24, no. 5, p. 1430003, Aug. 2014.

[20] P. N. Steinmetz, A. Manwani, C. Koch, M. London, and I. Segev, "Subthreshold voltage noise due to channel fluctuations in active neuronal membranes," *J. Comput. Neurosci.*, vol. 9, no. 2, pp. 133–148, Sep. 2000.

[21] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*. New York, NY, USA: Oxford Univ. Press, 1999.

[22] *ProcStar IV Internet*. (Jan. 2017). [Online]. Available: http://www.gidel.com/PROCstar%20IV.htm

[23] A. Morro, V. Canals, A. Oliver, M. L. Alomar, and J. L. Rossello, "Ultra-fast data-mining hardware architecture based on stochastic computing," *PLoS ONE*, vol. 10, no. 5, p. e0124176, May 2015.

[24] J. J. Irwin and B. K. Shoichet, "ZINC—A free database of commercially available compounds for virtual screening," *J. Chem. Inf. Model.*, vol. 45, no. 1, pp. 177–182, 2005.

[25] J. J. Hopfield, "Pattern recognition computation using action potential timing for stimulus representation," *Nature*, vol. 376, no. 6535, pp. 33–36, 1995.