An Analog Neural Network that Learns Sudoku-Like Puzzle Rules

Saeid Barzegarjalali
Department of Electrical Engineering
University of Southern California
Los Angeles, California 90089-2560
Email: sbarzega@usc.edu

Alice C. Parker
Department of Electrical Engineering
University of Southern California
Los Angeles, California 90089-2560
Email: parker@usc.edu

***Abstract*—We have designed a fully-connected neural network implemented as an analog circuit consisting of 8 neurons and 64 synapses that can learn rules of 2-by-2 Sudoku or Sudoku-like puzzles and then can solve them. In this circuit, learning is mediated by giving a *dopamine* reward signal to correct actions, which has a biological basis and is known as *reinforcement learning* [1]. Regular architecture of the circuit helps it to generalize learned rules and as a consequence expedites the learning procedure. The circuit receives *dopamine* externally from a *trainer* circuit and therefore learning is supervised. Injected *dopamine* will strengthen some of the existing excitatory synapses similar to biological *synaptic plasticity* [2]. Previously, we had designed a bio-inspired circuit that learns spatiotemporal patterns in an unsupervised mode using *structural plasticity*. In the human brain, learning is mediated by both types of plasticity [3].The long-term goal of our research group is combining these two types of plasticity (*synaptic* and *structural*) to design a network with higher level of learning capabilities and more complex cognitive skills to imitate the biological brain.**

***Keywords*—Bio-inspired electronics; dopamine receptor; Supervised learning; Synaptic plasticity**

I. INTRODUCTION

We have designed a trainable and fully-connected neural network with 8 neurons and 64 excitatory synapses. Every neuron is connected to all other 7 neurons via 7 excitatory synapses (one for each) that are normally silent but can be awakened. Also, every neuron is externally connected to an *excitation input* via an excitatory synapse that is always on. Therefore, a *trainer* circuit can force any of these neurons to fire to set initial conditions for each game. Figure 1 shows the architecture of the circuit. Blue circles represent neurons and black lines connecting them represent two synaptic connections. Black lines that are connected to each neuron individually and externally represent a synaptic connection to the neuron for the external excitation. For example, the bidirectional line between neurons *A1* and *A2* means that there is one synapse that connects the output of neuron *A1* to the input of neuron *A2* and there is another synapse that connects the output of neuron *A2* to the input of neuron *A1*. Both neurons can be excited externally with one synapse each. Figure 2 shows our puzzle with four squares, each of which is labeled with one of the letters A, B, C and D. There are two neurons in each square with index 1 or 2. These neurons fire to show the content of that square resembling *one-hot* encoding. For example, if *A1* fires, it means that there is 1 in square A and if *D2* fires, it means there is 2 in square D. We will show how this network can learn rules of 2-by-2 Sudoku or Sudoku like games with a bio-inspired mechanism and then can solve them. Also, we will show how regularity in the architecture of a neural network can expedite the learning procedure.

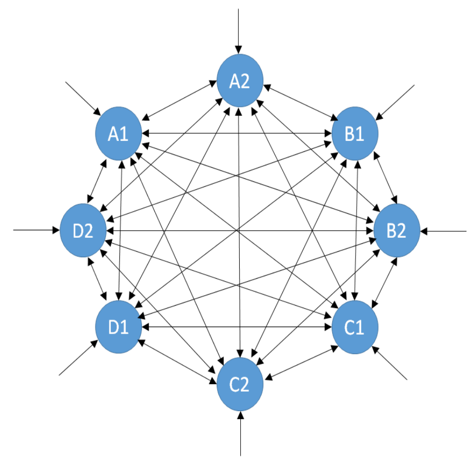

Fig. 1. Fully connected neural network with normally-off (silent) synapses. Every neuron can be excited externally with normally-on synapses.

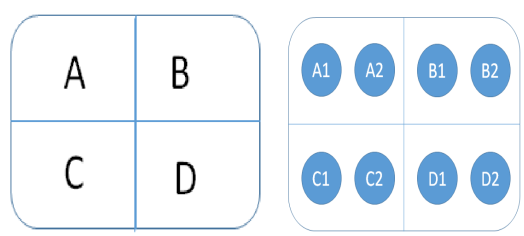

Fig. 2. A 2-by-2 Sudoku-like puzzle

II. THE CIRCUIT AT THE TRANSISTOR LEVEL

A. Synapse

To illustrate the complexity of our circuits, Figure 3 shows our synapse at the transistor level which previously had been designed with CNT transistors [4], but here it is designed with CMOS TSMC45nm technology. Variations of this synapse have been published for some time [5]. Input to this synapse is the *Action Potential* of the pre-synaptic neuron and the output of the synapse is an *Excitatory Post-synaptic Potential*

(EPSP). The *Neurotransmitter Concentration* knob on the synapse modulates the amplitude of the *EPSP* output of the circuit; when it is grounded the synapse will be off and when it is connected to VDD the synapse will be fully on. There are 64 of these synapses in the circuit including 8 synapses to excite every neuron externally and 56 synapses that connect every neuron to all other neurons.

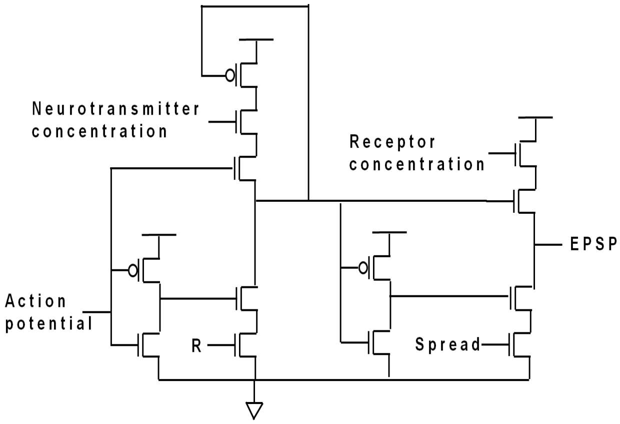

Fig. 3. An excitatory synapse adapted from [4]

B. The neuron

A neuron in our circuit, as shown in Figure 4, consists of two components, the Dendritic Arbor and the Axon Hillock. Inputs to the neuron are 8 *EPSPs* coming from the other 7 neurons and one external excitation input. Output of the neuron is an *Action Potential*. Figure 5 shows the 8-input Dendritic Arbor of the neuron. The Dendritic Arbor is made of seven 2-input adders that previously had been designed and used by others [6], [7]. The Axon Hillock that generates an *Action Potential* has also been designed previously and has been used in some publications [5], [7].

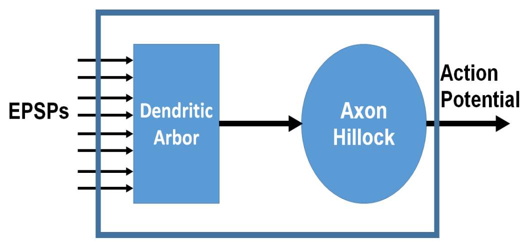

Fig. 4. A neuron in our circuit consists of the dendritic arbor and the axon hillock.

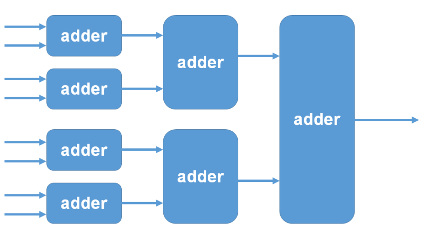

Fig. 5. 8-input Dendritic Arbor

C. Circuits for synaptic plasticity and reward-based learning

Figure 6 shows the block diagram of the *synaptic plasticity* and learning part of the circuit. The circuit monitors firing patterns of the neurons, checks which patterns receive *dopamine*, and based on firing patterns sets *Neurotransmitter Concentrations* for the appropriate synapses. The circuit consists of three types of components which are as follows:

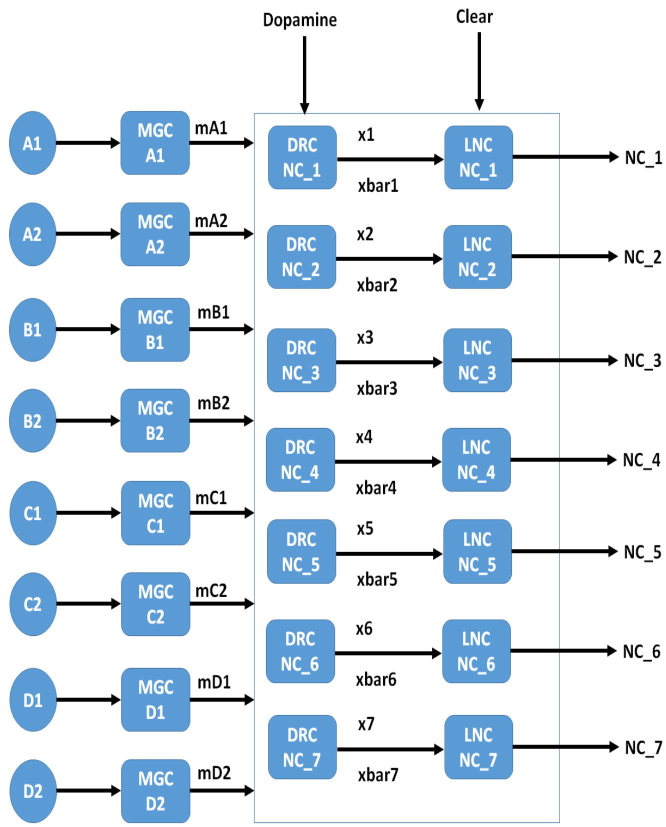

Fig. 6. Block diagram of *synaptic plasticity* and learning mechanism

1) Monostable Generator Circuit (MGC)*:* Figure 7 shows our bioinspired Monostable Generator Circuits *(MGCs)* for all neurons. As a case in point, input to the *MGC* circuit of neuron *A1* is *Action Potential* of neuron *A1* and output is *mA1*. If neuron *A1* is silent for awhile, node *P* will be discharged through transistor 4 and mA1 will be high because of the inverter. If *A1* fires, transistor 2 turns on and its current will be mirrored via transistors 1 and 3 and the current will charge up node *P* and as a consequence node *mA1* will go low. After sometime (determined by *bias2*), node *P* will be discharged again and node *mA1* will go high again. Therefore, node *mA1* is normally high and generates a negative logic pulse when the neuron *A1* fires. Duration of this pulse determines temporal tolerance for detecting synchrony of firing of the neurons in the presence of *dopamine*. There is one *MGC* for each neuron and as a consequence 8 *MGCs* in total.

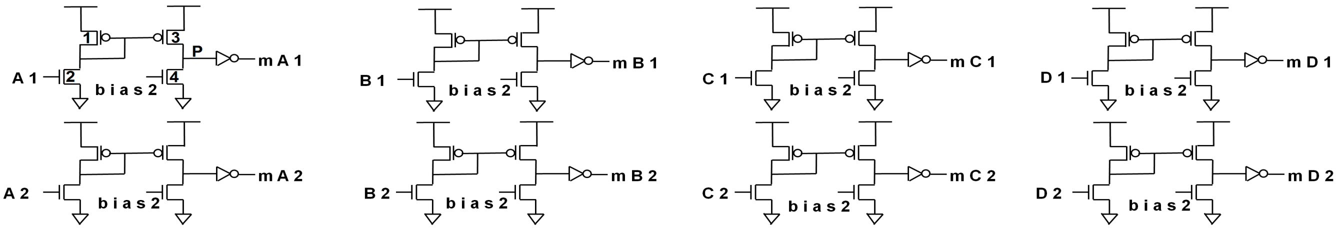

Fig. 7. Monostable Generator Circuits *(MGCs)* that synchronize neural firing

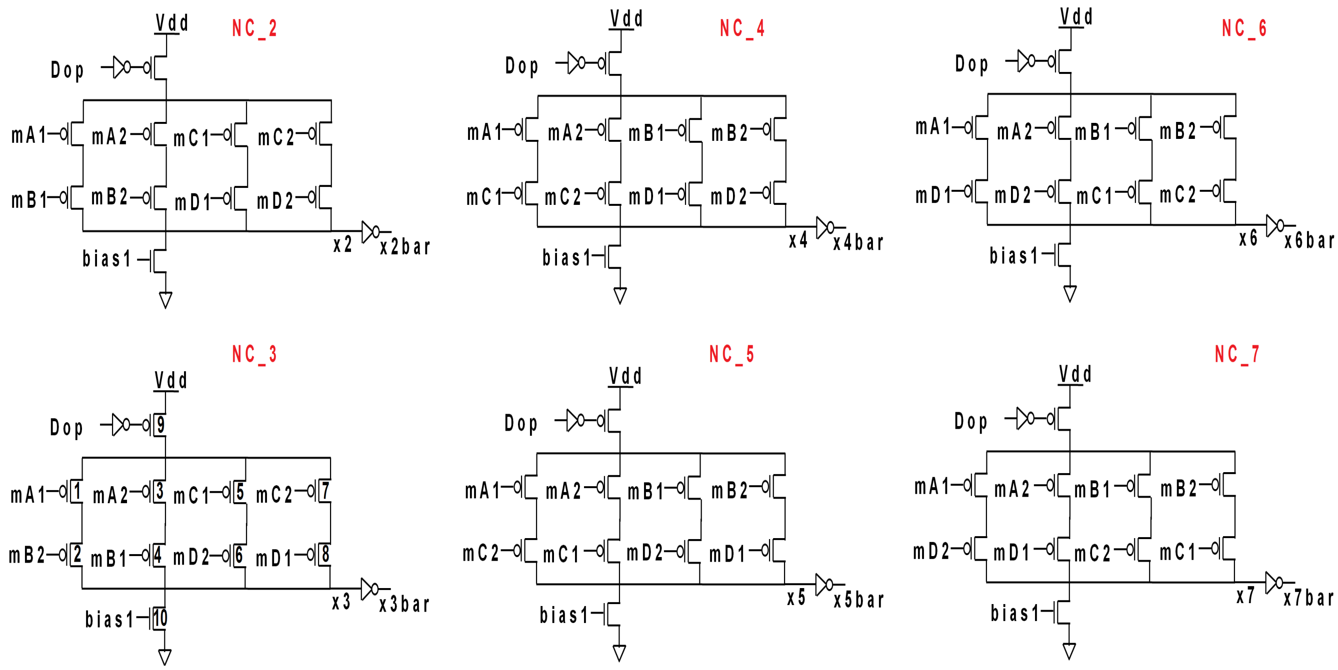

Fig. 8. Dopamine Receptor Circuits (*DRCs*) that detect correct firing

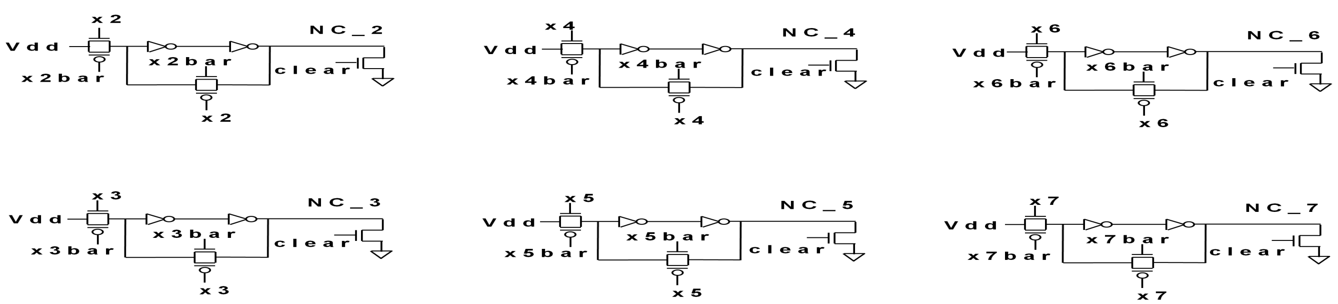

Fig. 9. Latches for Neurotransmitter Concentrations (*LNCs*) to maintain learned synaptic strengths

2) Dopamine Receptor Circuit (DRC): Figure 8 shows the Dopamine Receptor Circuits *(DRCs)*. These circuits detect synchronous firing of neurons leading to a correct puzzle solution. As a case in point, in *NC_3*, Node *x3* is normally discharged through transistor 10 and, because of the inverter, *x3bar* is normally high. If *A1* fires simultaneously with *B2*, *A2* fires simultaneously with *B1*, *C1* fires simultaneously with *D2* or *C2* fires simultaneously with *D1*, and *DOP* signal is high (*dopamine* is given to the circuit) at least for one of these events, then node *x3* will be charged up to VDD via one of the branches (transistors 1-2, 3-4, 5-6, or 7-8) along transistor 9 and *x3bar* will go low. After some time *x3* and *x3bar* will go low and high respectively; this time is determined by *bias1*. There are 7 *DRCs* in the final circuit that will be explained in the next sections. *Dopamine Receptor Circuit* for *NC_1* is not shown in the figure.

3) Latch for Neurotransmitter Concentration (LNC) *:* There is one Latch for each Neurotransmitter Concentration value (Figure 9). These latches maintain learned synaptic strengths. *NC_x* nodes can be reset with *clear* signal and then stay low. The *x* and *xbar* signals from corresponding *DRCs* are normally low and high respectively and as a consequence the latches are normally in holding mode. If *x* and *xbar* signals from the corresponding *DRC* go high and low respectively, each latch will be in writing mode and VDD will be written to *NC_x*. Then *x* and *xbar* signals will return to their normal points and latch will hold the written VDD.

III. Eight Types of Synapses used in the Regular Architecture

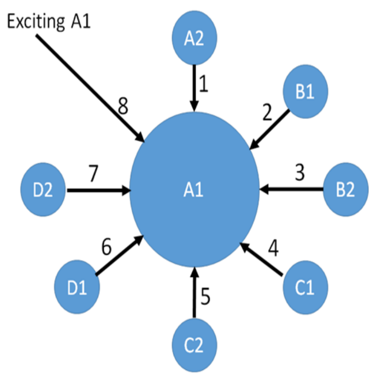

Fig. 10. Inputs to neuron *A1*

Figure 10 shows 8 synaptic connections to neuron *A1* coming from other neurons and external excitation. All the 64 synapses in the circuit can be classified in one of these 8 types as follow:

Type 1: Synapses that connect two neurons inside the same square which are *A1-A2, B1-B2, C1-C2, D1-D2*. These synapses are controlled by *NC_1*. Neurons in the same square are not supposed to fire simultaneously. In other words, content of a square cannot be 1 and 2 at the same time. Therefore, we do not set *NC_1* to VDD in any of the games.
Type 2: Synapses that connect a neuron to a neuron in its horizontal adjacent square with the same number which are *A1-B1, A2-B2, C1-D1, C2-D2*.These synapses are controlled by *NC_2*.
Type 3: Synapses that connect a neuron to a neuron in its horizontal adjacent square with the different number which are *A1-B2, A2-B1, C1-D2, C2-D1*.These synapses are controlled by *NC_3*.
Type 4: Synapses that connect a neuron to a neuron in its vertical adjacent square with the same number which are *A1-C1, A2-C2, B1-D1, B2-D2*.These synapses are controlled by *NC_4*.
Type 5: Synapses that connect a neuron to a neuron in its vertical adjacent square with the different number which are *A1-C2, A2-C1, B1-D2, B2-D1*.These synapses are controlled by *NC_5*.
Type 6: Synapses that connect a neuron to a neuron in its diagonal adjacent square with the same number which are *A1-D1, A2-D2, B1-C1, B2-C2*.These synapses are controlled by *NC_6*.
Type 7: Synapses that connect a neuron to a neuron in its diagonal adjacent square with the different number which are *A1-D2, A2-D1, B1-C2, B2-C1*.These synapses are controlled by *NC_7*.
Type 8: Synapses that connect external excitation to corresponding neurons which are *A1, A2, B1, B2, C1, C2, D1, D2*.These synapses are always on (i.e. Their NC knobs are hardwired to VDD.), so the *trainer* can always cause any of the neurons to fire to train the network or to initiate the process to solve a puzzle after it learns the rules of the games.

IV. Training Type 3 Synapses

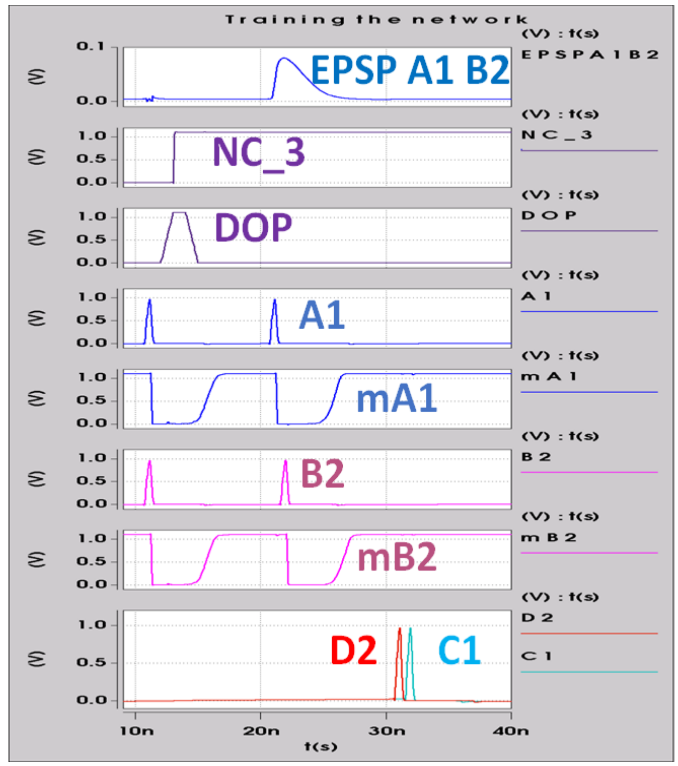

Fig. 11. Setting *NC_3*

Synapses become trained (strengthened) when correct answers are provided by the neural network to the trainer circuit. A simple experiment shows how the NC signals that train the synapses are generated. Figure 11 shows the simulation result for setting *NC_3*, and the results of the training that sets *NC_3*. At t=10 nanoseconds, both *A1* and *B2* neurons are forced to fire externally by the *trainer* and the circuit receives *dopamine* for this firing pattern. This *dopamine* shot sets *NC_3* to VDD. At t=20 nanoseconds, we assume the circuit has been trained. Now we start a game session. Only neuron *A1* is forced to fire but the synapse connecting neuron *A1* to neuron *B2* has been turned on (*EPSP A1 B2*) by *NC_3* and neuron *B2* fires due to firing of neuron *A1*. This shows the direct effect of the previous training. At t=30 nanoseconds, we continue the game session. Neuron *D2* is externally forced to fire and since all type 3 synapses are on, it causes its horizontal adjacent neuron with different number (*C1*) to fire. This means the circuit can generalize rules.

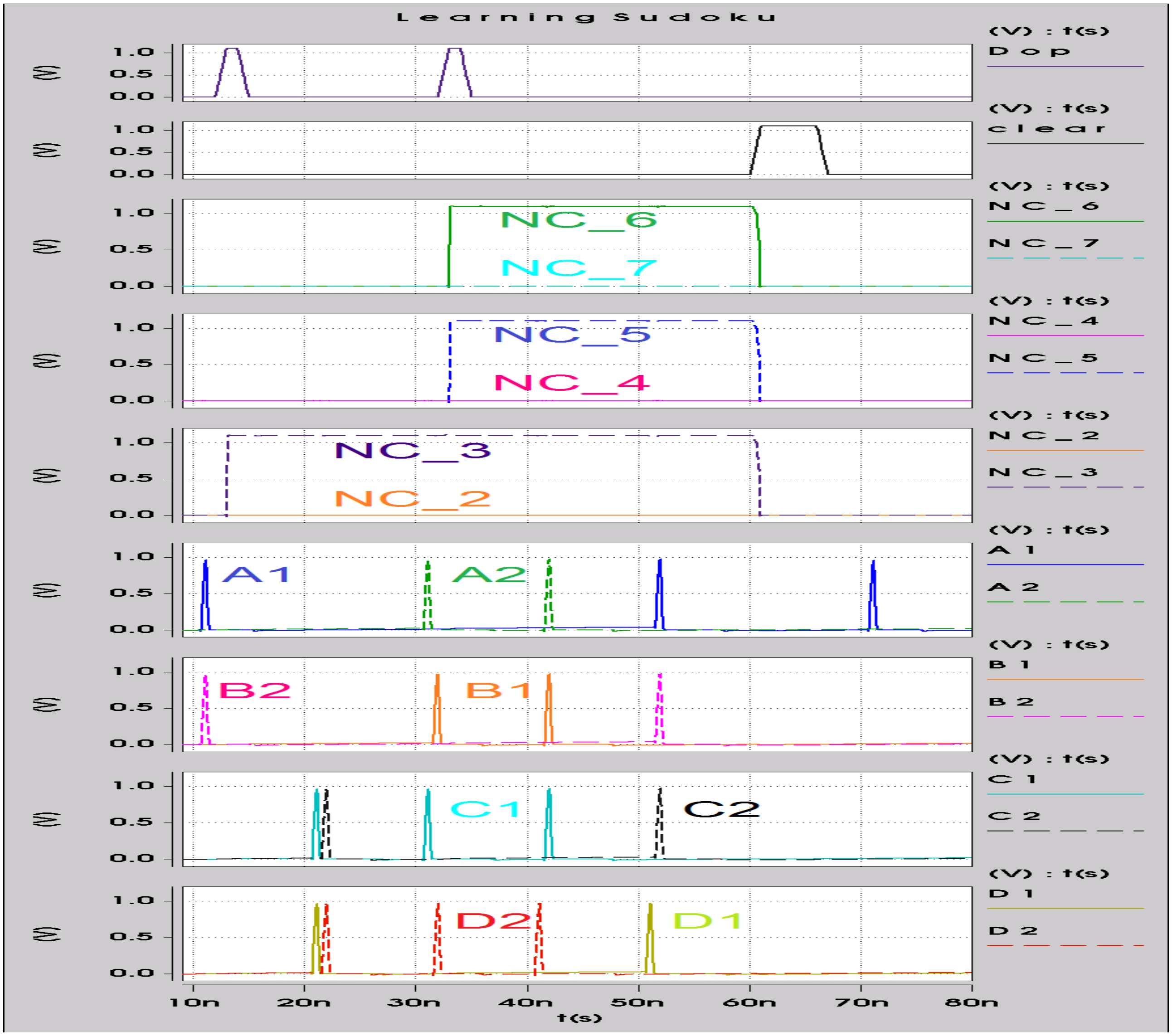

Fig. 12. A supervisor is training the network by giving a *dopamine* reward.

V. The Network Can Get Trained and Then Can Solve a 2-by-2 Sudoku

There are eight different ways of beginning a 2-by-2 Sudoku by setting one of the squares to either 1 or 2 and there are two possible solutions as shown in Figure 14. According to the rules of Sudoku, in every row and in every column all numbers (here 1 and 2) should repeat exactly once. Therefore, *NC_3*, *NC_5* and *NC_6* should set to VDD and all other *NC*s should stay low (off). This is because numbers in the same row and column are different and on the same diagonal are equal. Figure 12 shows how the network can learn to solve the Sudoku by two shots of *dopamine*. We train the network with three different inputs at 10, 20 and 30 ns. At t=10 nanoseconds, both *A1* and *B2* neurons are forced to fire and the circuit gets *dopamine shot* for this firing pattern. As a consequence, *NC_3* goes high. The circuit learns synapses between neurons in the same row with different numbers should turn on.

At t=20 nanoseconds, neurons *C1* and *D1* are forced to fire and they cause *D2* and *C2* to fire respectively because of what had been learned previously with *NC_3* high. Since there is no reward for this firing pattern, *NC_2* does not go high. At t=30 nanoseconds, both *A2* and *C1* neurons are forced to fire and they cause neurons *B1* and *D2* to fire respectively. The *trainer* gives *dopamine* shot for this firing pattern. As a consequence, *NC_5* and *NC_6* go high. This means the circuit has learned that synapses between neurons in the same column with different numbers and synapses between diagonal neurons with the same number should turn on. At this point, the circuit has learned all the rules of the Sudoku game and can solve it. At t=40 nanoseconds, we begin a new game with the rules learned from training. Neuron *D2* is forced to fire. Solution to this input is solution b shown in Figure 14 and therefore neurons *A2*, *B1* and *C1* fire in response. At t=50 nanoseconds, we begin a new game and neuron *D1* is forced to fire. Solution to this input is solution a and therefore neurons *A1*, *B2* and *C2* fire in response. At t=60 nanoseconds, the *trainer* activate *clear* signal and all *NC* signals go low. This means the circuit forgets the rules in response to *clear* signal and as a consequence

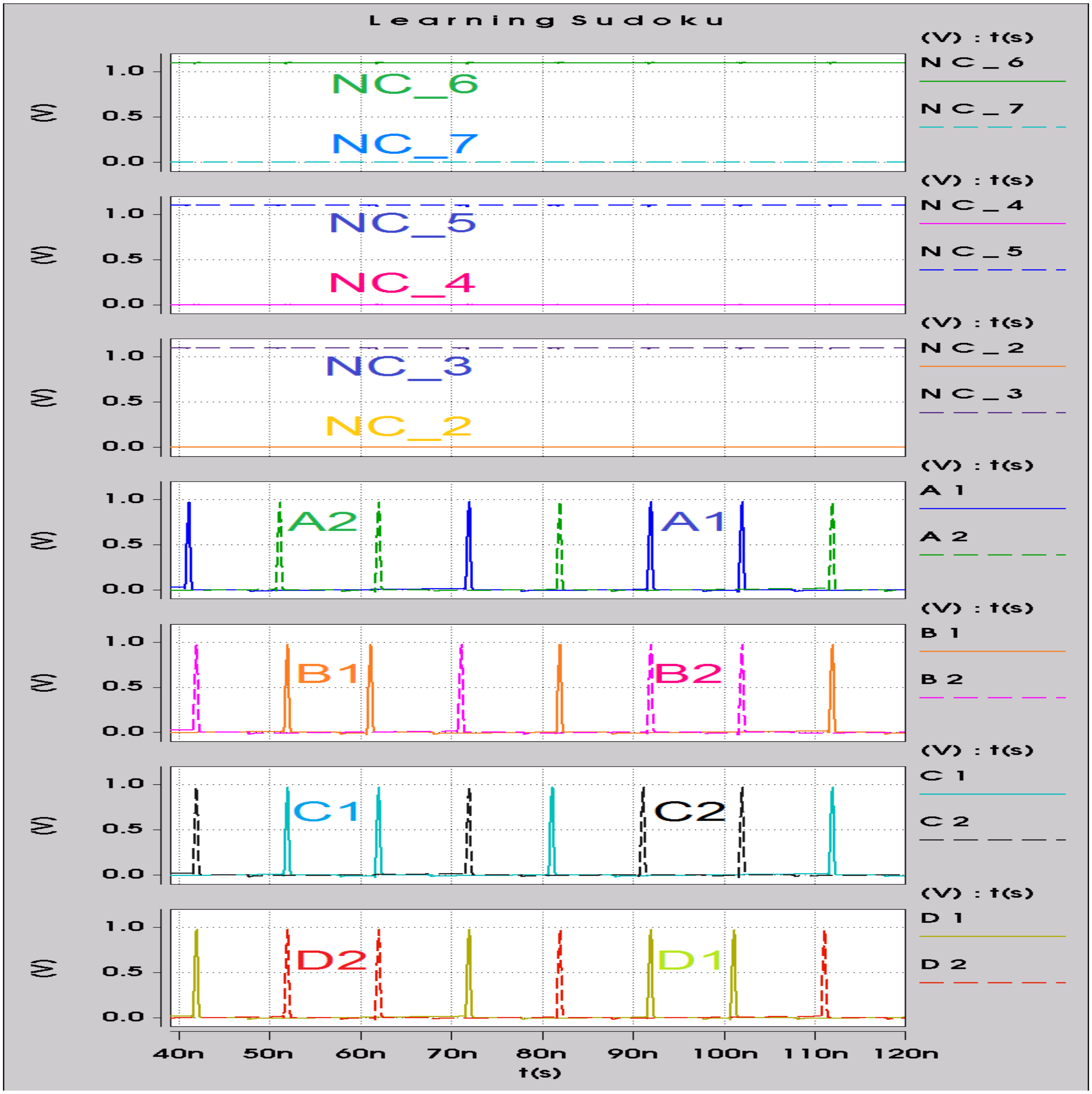

Fig. 13. Network is ready to solve 8 different possibilities.

at t=70 nanoseconds, when neuron *A1* is forced to fire, the circuit does not show response.

In Figure 13 the network has learned rules of Sudoku. *NC_3*, *NC_5*, and *NC_6* have been set to VDD. *NC_2*, *NC_4*, and *NC_7* are low. *NC_1* (not shown in the figure) is also low, because we do not want to have 1 and 2 in the same square at the same time. At t=40 nanoseconds, *A1* is forced to fire. Response of the circuit is solution a. At t=50 nanoseconds, *A2* is forced to fire. Response of the circuit is solution b. At t=60 nanoseconds, *B1* is forced to fire. Response of the circuit is solution b. At t=70 nanoseconds, *B2* is forced to fire. Response of the circuit is solution a. At t=80 nanoseconds, *C1* is forced to fire. Response of the circuit is solution b. At t=90 nanoseconds, *C2* is forced to fire. Response of the circuit is solution a. At t=100 nanoseconds, *D1* is forced to fire. Response of the circuit is solution a. At t=110 nanoseconds, *D2* is forced to fire. Response of the circuit is solution b.

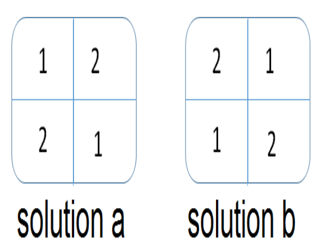

Fig. 14. Any solution to a 2-by-2 Sudoku will be one of these two possibilities.

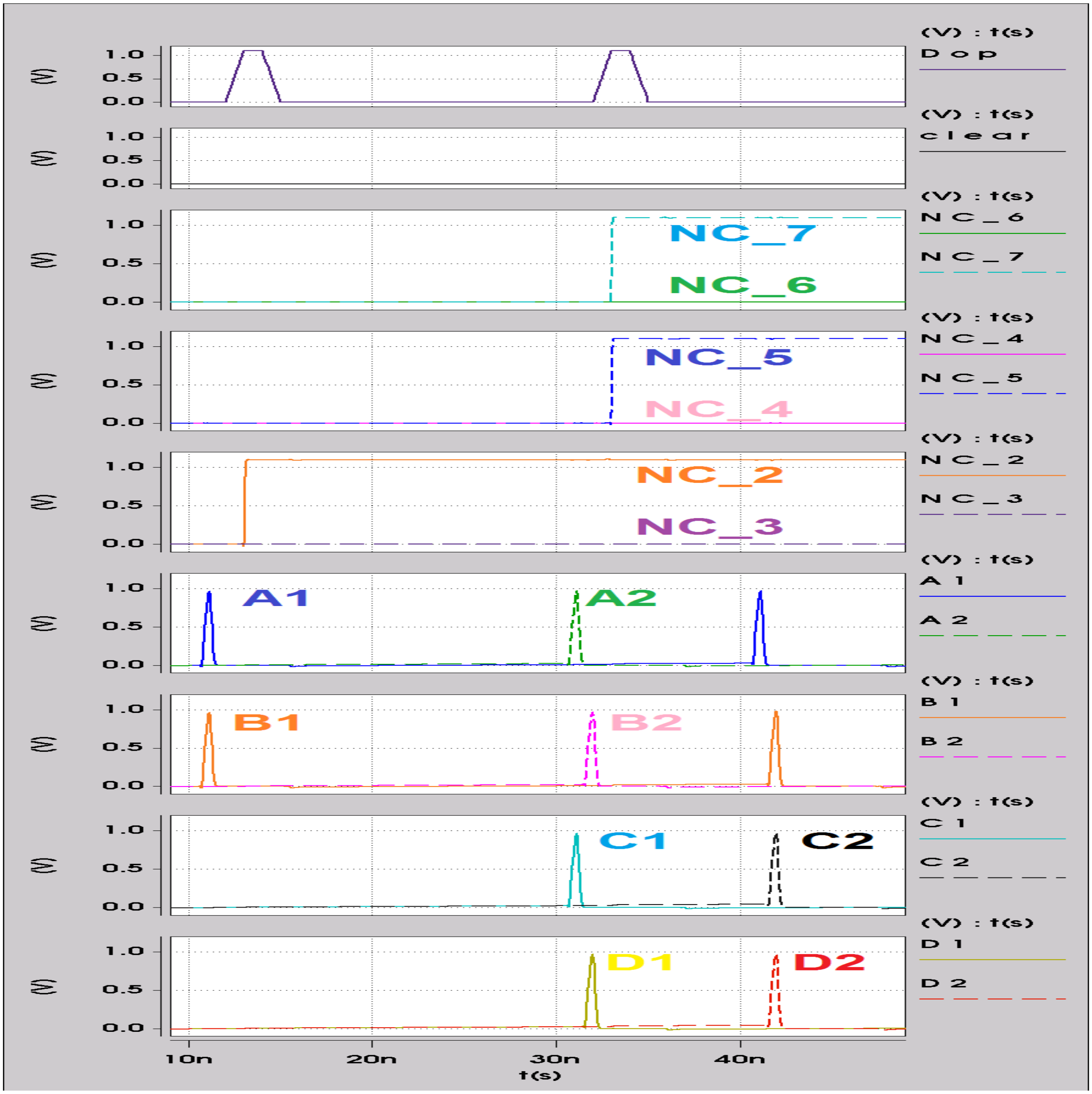

Fig. 15. Network is learning Pseudo Sudoku type I.

VI. PSEUDO SUDOKU TYPE I

Now, we define a new game in which numbers in the same row are equal but in the same column are different. We use a different trainer circuit for this game. There are two possible solutions to this game that are shown in Figure 17. Like Sudoku, the network can learn the rules by two *dopamine* shots. Figure 15 shows how the network can learn rules of Pseudo Sudoku type I by two shots of *dopamine*. At t=10 nanoseconds, both of *A1* and *B1* neurons are forced to fire and the circuit gets *dopamine shot* for this firing pattern. As a consequence, *NC_2* goes high. The circuit learns synapses between neurons in the same row with the same numbers should turn on.

At t=30 nanoseconds, both of *A2* and *C1* neurons are forced to fire and they cause neurons *B2* and *D1* to fire respectively. The *trainer* gives *dopamine* shot for this firing pattern. As a consequence, *NC_5* and *NC_7* go high. This is because numbers on the same column and on the same diagonal are different in this new game. At this point, the network has learned the rules of the new game. Therefore, at t=40 nanoseconds, when neuron *A1* is forced to fire, neurons *B1*, *C2*, and *D2* fire as response which is Solution C.

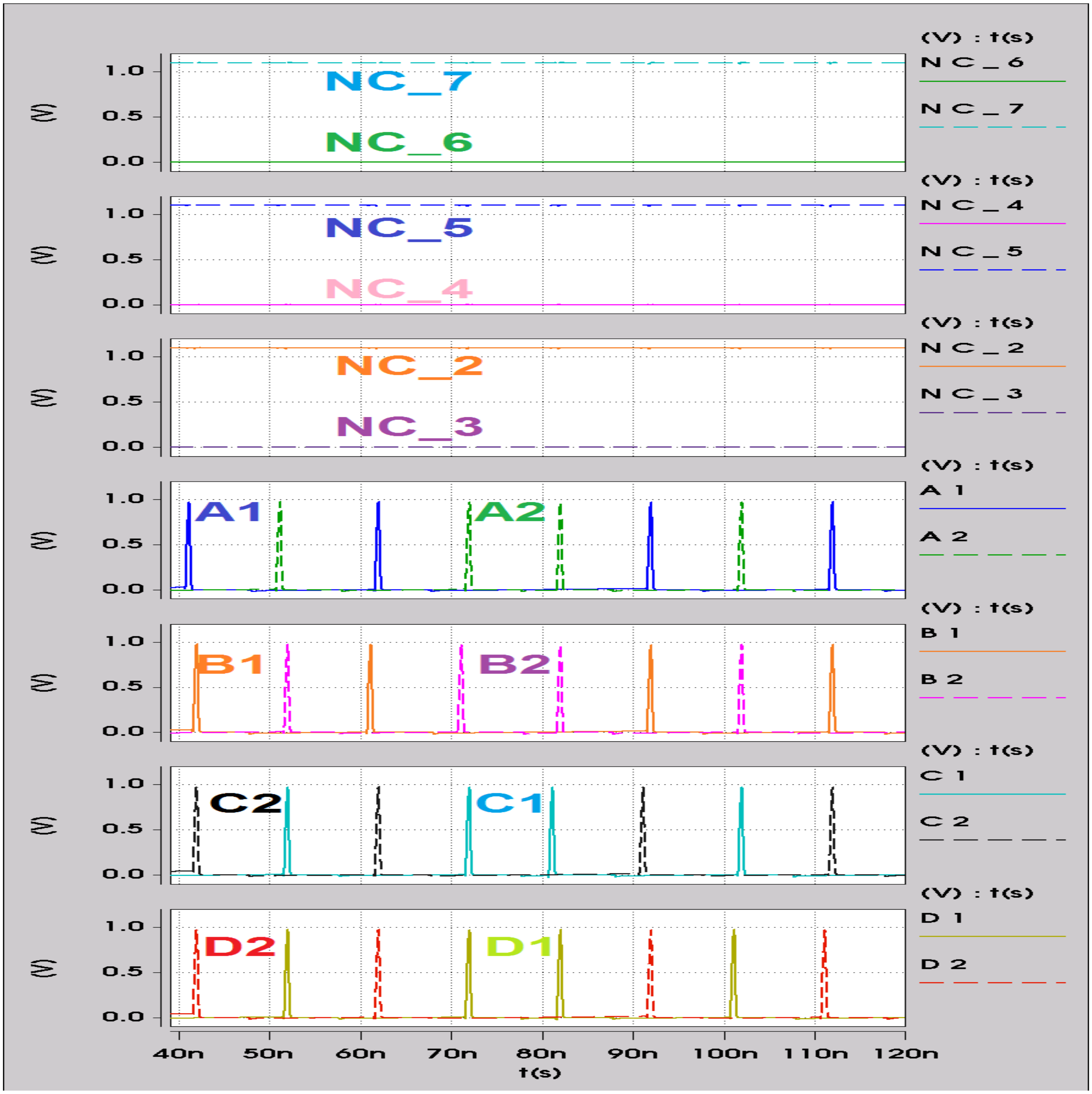

Fig. 16. Network solves 8 different possibilities of Pseudo Sudoku type I.

Figure 16 shows the network has learned rules of Pseudo Sudoku type I. *NC_2*, *NC_5*, and *NC_7* have been set to VDD. *NC_3*, *NC_4*, and *NC_6* are low. *NC_1* (not shown in the figure) is also low, because we do not want to have 1 and 2 in the same square at the same time. At t=40 nanoseconds, *A1* is forced to fire. Response of the circuit is solution c. At t=50 nanoseconds, *A2* is forced to fire. Response of the circuit is solution d. At t=60 nanoseconds, *B1* is forced to fire. Response of the circuit is solution c. At t=70 nanoseconds, *B2* is forced to fire. Response of the circuit is solution d. At t=80 nanoseconds, *C1* is forced to fire. Response of the circuit is solution d. At t=90 nanoseconds, *C2* is forced to fire. Response of the circuit is solution c. At t=100 nanoseconds, *D1* is forced to fire. Response of the circuit is solution d. At t=110 nanoseconds, *D2* is forced to fire. Response of the circuit is solution c.

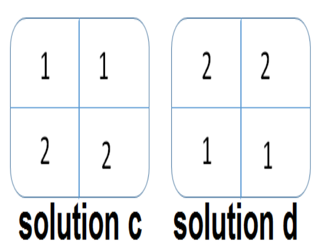

Fig. 17. Pseudo Sudoku type I with two possible solutions

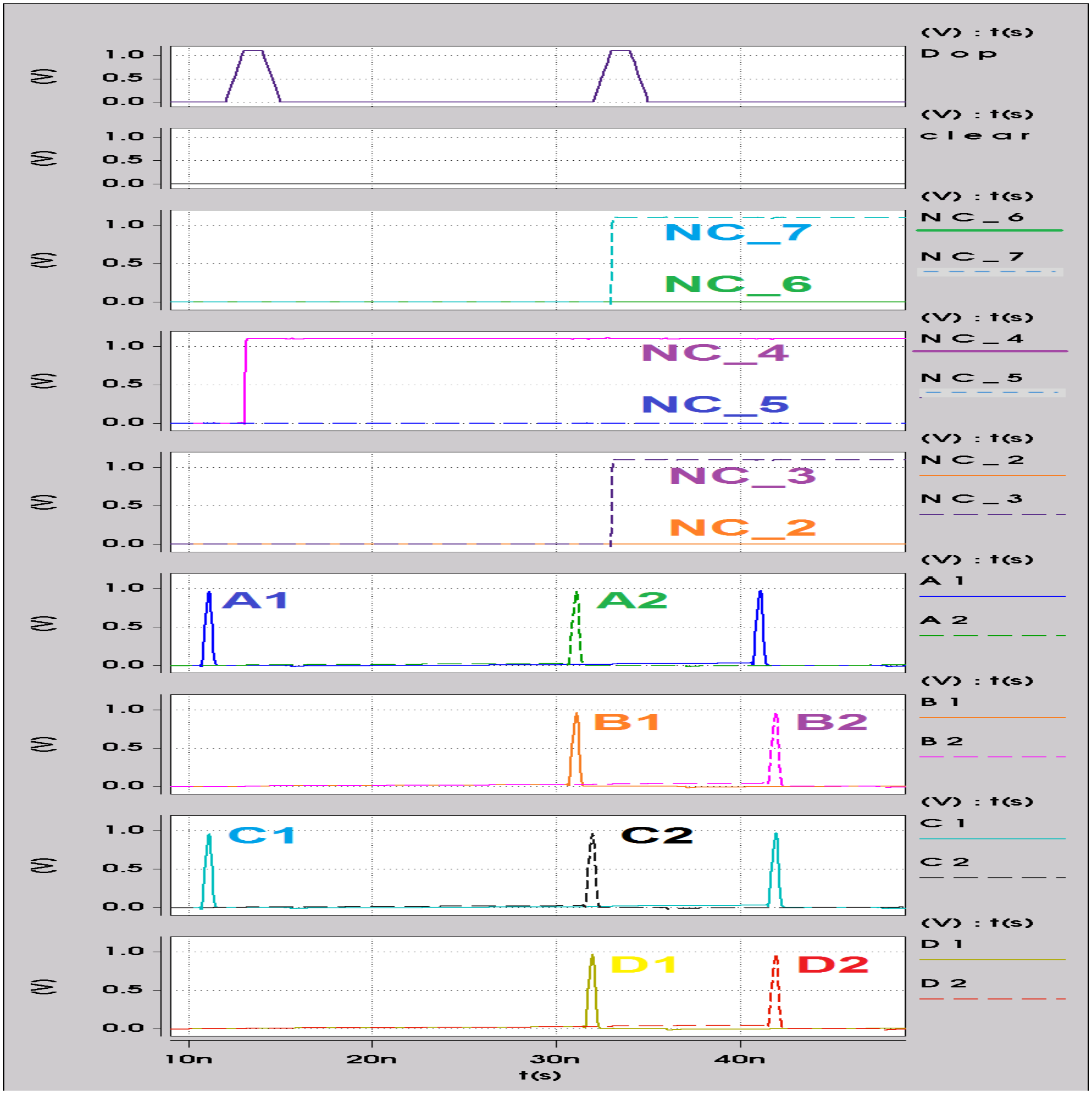

Fig. 18. Network is learning Pseudo Sudoku type II.

VII. Pseudo Sudoku Type II

In Pseudo Sudoku type II, numbers on the same row are different and numbers on the same column are equal as shown in Figure 19. Like previous games, *trainer* can train the network with two *dopamine* shots. In Figure 18, *A1* and *C1* are forced to fire and network gets *dopamine*. This sets *NC_4* to VDD. Then *A2* and *B1* are forced to fire and they cause neurons *C2* and *D1* to fire respectively. Then circuit gets *dopamine* and *NC_3* and *NC_7* get set to VDD. At this point the network knows how to play the game and when *A1* is forced to fire, response of the circuit is solution e. Figure 20 shows how the network can solve eight different possibilities of the pseudo-sudoku type II game.

At 40, 50, 60, 70, 80, 90, 100, and 110 nanoseconds neurons *A1, A2, B1, B2, C1, C2 ,D1, and D2* are forced to fire respectively and response of the network are solutions e, f, f, e, e, f, f, and e respectively.

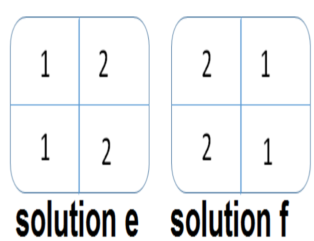

Fig. 19. Pseudo Sudoku type II with two possible solutions

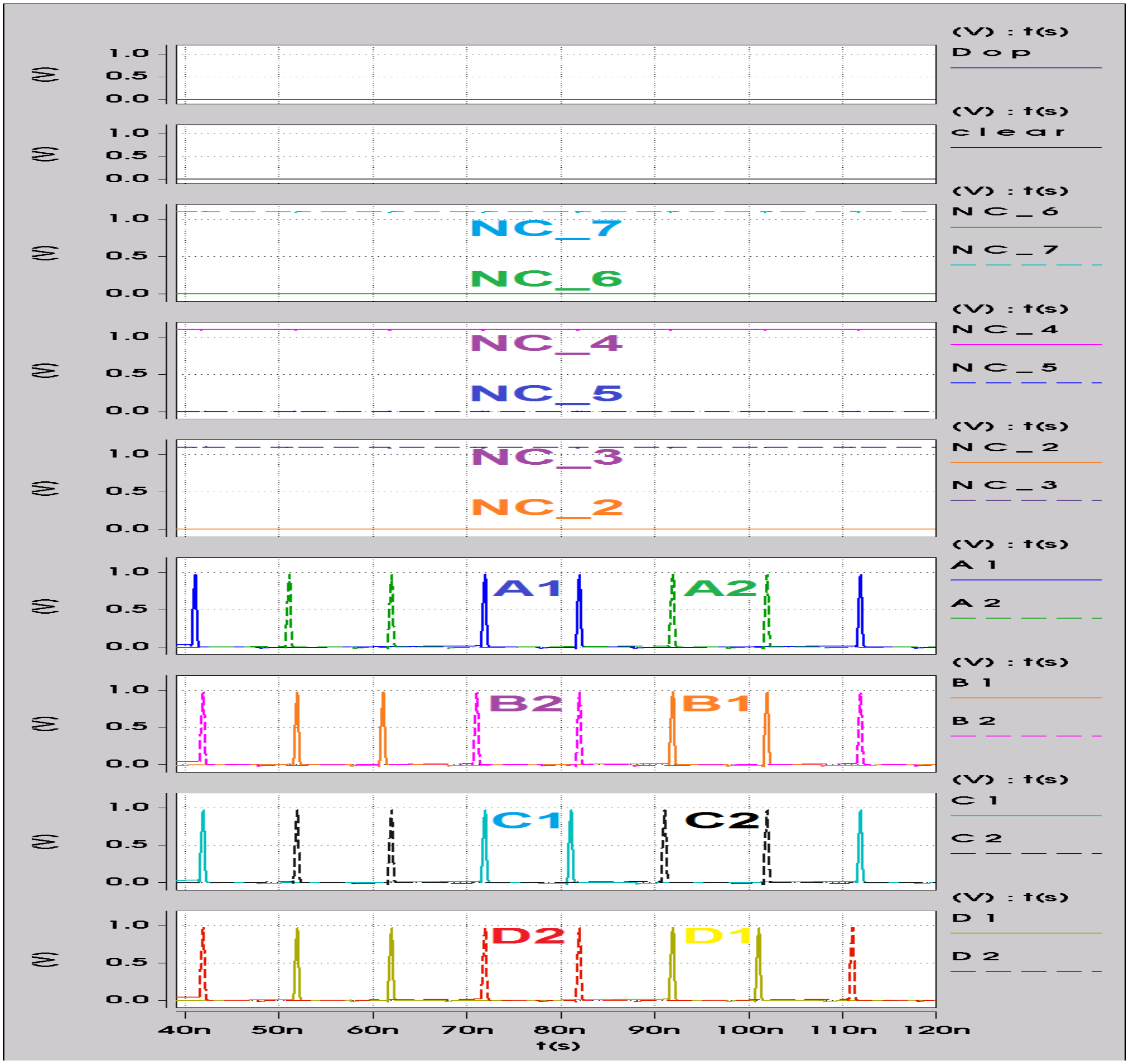

Fig. 20. Network solves 8 different possibilities of Pseudo Sudoku type II.

VIII. Conclusion

We have designed a neural network that can learn in a supervised mode using *synaptic plasticity* with a reward-based mechanism. Regular architecture of the neural network helps it to learn the rules of 2-by-2 Sudoku or Sudoku like games with only two shots of *dopamine*. Neurons and synapses are bio-inspired and imitate their biological counterparts. The learning mechanism is also bio-inspired. In our previous work, we had designed a circuit that learns in an unsupervised mode using *structural* plasticity. Combining these mechanisms and scaling up to more complex examples is the ultimate goal of our group. The neural network itself will increase in size linearly as the problem size increases, and the training circuits will scale as well.

References

[1] Alexxai V Kravitz, Lynne D Tye, and Anatol C Kreitzer. “Distinct roles for direct and indirect pathway striatal neurons in reinforcement”. *Nat Neurosci*, 15:816–818, 2012.

[2] Stefano Fusi. “Hebbian spike-driven synaptic plasticity for learning patterns of mean firing rates”. *Biological Cybernetics*.

[3] G. Indiveri and Shih-Chii Liu. “Memory and information processing in neuromorphic systems”. *Proceedings of the IEEE*, 103(8):1379–1397, Aug 2015.

[4] J. Joshi, A.C. Parker, and Chih-Chieh Hsu. “A carbon nanotube cortical neuron with spike-timing-dependent plasticity”. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 1651–1654, Sept 2009.

[5] M. Mahvash and A. C. Parker. “Modeling intrinsic ion-channel and synaptic variability in a cortical neuromorphic circuit”. In *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 69–72, Nov 2011.

[6] H. Chaoui. “CMOS analogue adder”. *Electronics Letters*, 31(3):180–181, Feb 1995.

[7] S. Barzegarjalali and A.C. Parker. “A hybrid neuromorphic circuit demonstrating schizophrenic symptoms”. In *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*, pages 1–4, Oct 2015.