Neural Flip-Flops IV: Lamprey Locomotion

Lane Yoder
Department of Science and Mathematics, retired
University of Hawaii, Kapiolani
Honolulu, Hawaii
LYoder@hawaii.edu
NeuralNanoNetworks.com

Abstract

The lamprey is one of the most ancient of extant vertebrate species. It has changed relatively little in 450 million years and is considered a prototype for all vertebrates. Its primitive nervous system has been studied extensively, and the basic architecture of the central pattern generator (CPG) that produces its anguilliform swimming motion is well known.

Here it is shown that each segmental component of the lamprey's CPG is a JK flip-flop, with additional excitatory inputs and feedback that cause all of the neurons' states to oscillate. The JK flip-flop is the most widely used flip-flop design in electronic computational systems because of its advantageous features. This is apparently the first discovery that a known network of neurons functions as a logic circuit. The lamprey's oscillator can be implemented with electronic hardware, and the design is apparently unknown in engineering.

A simulation based on simple neuron responses to excitation and inhibition illustrates the common period, phase relationships, and burst durations of the segmental cells' oscillations. Simulation software for electronic logic circuits verifies the simulated neuron responses, on vastly different time scales. The simulation methods presented here may aid in further study of CPG neurophysiology. The architecture of the oscillating JK flip-flop may aid in the development of artificial neural network applications such as robotics.

Key words: lamprey, locomotion, central pattern generator, CPG, flip-flop, JK flip-flop, oscillator, bursting neuron, explicit neural model, neural network, neuronal network, robotics.

1. Introduction

Flip-flops are the basic building blocks of sequential logic systems, whose logic gate outputs are functions of both the current inputs and the past sequence of inputs. This article is the fourth in a series of articles that show how neurons are likely to be connected to perform certain Boolean logic functions with networks composed of neural flip-flops (NFFs). The first three articles [1-3] showed that NFFs and NFFs configured as central pattern generators (CPGs) can produce the major phenomena of short-term memory, electroencephalography, and the lobster's stomatogastric ganglion. Three previous articles [4-6] explored the analog properties of neuron signals in combinational logic operations, whose outputs depend only on the current state of the inputs. A family of fuzzy logic decoders was shown to generate the phenomena central to color vision and olfaction. All of these network designs show all of the neurons and their connections explicitly, and their operation depends only on dynamic neuron properties of excitation and inhibition. The networks are logic circuits that can be composed of electronic hardware as well as neurons.

The six previous articles proposed novel networks of neurons that produce known nervous system phenomena. The present article shows that a known network of neurons functions as a logic circuit, and that this logic circuit's design is essentially the same as a standard electronic logic circuit design. Each segmental component of the central pattern generator (CPG) that controls the lamprey's swimming motion is a JK flip-flop with set and reset inputs provided by excitatory neurons, each of which has inhibitory feedback from the flip-flop's opposite memory bit output. This arrangement is unstable, causing all of the cells' states to oscillate with a common period and various phases and burst durations.

Besides the addition of set and reset cells that generate oscillations, the only real difference between the lamprey's segmental component and a standard JK flip-flop is an enabling signal from each input cell to the corresponding output cell. The selective pressure that led to the lamprey's deviation from the modern JK flip-flop design is not clear, but there are at least two possibilities. With excitatory input, the output cells do not need to be spontaneously active. Spontaneously active cells may have been unavailable or uncommon in the early evolution of vertebrate CPGs. Second, electronic JK flip-flops are virtually always enabled by input from a clock. The lamprey's enabling signals from the input cells may substitute for this timing function.

The lamprey's CPG has been studied extensively. Its basic organization of synaptic connections is well known [7-12], although details are uncertain [13, 14]. One of these details is that a group of neurons is commonly represented by a single neuron symbol in simplified CPG figures. A group's size and organization of synaptic connections may affect the periods, phases, and burst durations of the neurons' oscillations. The lamprey's CPG has been simulated mathematically [9, 15], and the CPG has been used as a model for artificial controllers that can produce swimming movements [16].

2. Methods

The lamprey's CPG was simulated in MS Excel and CircuitLab. For the implementation with neurons, the number t_i represents the time after i neuron delay times. All neuron delay times are assumed to be equal, but minor differences would not have a significant effect on the simulations. A JK flip-flop's outputs must be opposite, so at time $t_0 = 0$, the neuron response of one of the outputs was initialized at 1 and the rest were initialized at 0. (If all responses are initialized at 0, the simulated responses oscillate erratically.) For $i > 0$, the output of each neuron at time t_i was

computed as a function of the inputs at time t_{i-1} according to the neuron responses derived below and indicated in the first figure. After a few neuron delay times, the responses fall into periodic patterns. To verify the simulated neural implementation, an electronic implementation was simulated in CircuitLab. The simulation software initializes the states.

3. Neural Boolean logic

All Boolean logic results for the networks presented here follow from the neuron responses to binary (high and low) input signals and the algebra of Boolean logic applied to the networks' connections. Analog signals (intermediate strengths between high and low) were considered in [1, 2] only to show how NFFs can generate robust binary signals in the presence of moderate levels of additive noise in binary inputs. That discussion will not be repeated here.

3.1. Binary neuron signals

Neuron signal strength, or intensity, is normalized here by dividing it by the maximum possible intensity for the given level of adaptation. This puts intensities in the interval from 0 to 1, with 0 meaning no signal and 1 meaning the maximum intensity. The normalized number is called the *response intensity* or simply the *response* of the neuron. Normalization is only for convenience. Non-normalized signal strengths, with the highest and lowest values labeled Max & Min rather than 1 and 0, would do as well.

The responses 1 and 0 are called binary signals collectively and high and low signals separately. If 1 and 0 stand for the Boolean truth values TRUE and FALSE, neurons can process information contained in binary signals by functioning as logic operators.

The strength of a signal consisting of action potentials, or spikes, can be measured by spike frequency. A high signal consists of a burst of spikes at the maximum spiking rate. For a signal that oscillates between high and low, the frequency of the oscillation is the frequency of bursts, not the frequency of spikes.

For binary signals, the response of a neuron with one excitatory and one inhibitory input is assumed to be as shown in Table 1. Of the 16 possible binary functions of two variables, this table represents the only one that is consistent with the customary meanings of "excitation" and "inhibition." Table 1 is also a logic truth table, with the last column representing the truth values of the statement X AND NOT Y. In simplest terms, the neuron performs this logic function because it is active when it has excitatory input *and* does *not* have inhibitory input.

Excitatory X	Inhibitory Y	Response
0	0	0
0	1	0
1	0	1
1	1	0

Table 1. Neuron response to two binary inputs, one excitatory and one inhibitory. The response is the logical truth value of X AND NOT Y.

The AND-NOT gate is virtually never used as a building block in logic circuit design. Its significance for the networks presented here is that it can be implemented with a single neuron. It was shown in [5] that the AND-NOT gate, with access to high input, is functionally complete, meaning any logic function can be performed by a network of such components.

Some neurons are active spontaneously and continuously without excitatory input [17, 18]. In the figures, neurons with no excitatory input are spontaneously

active. The behavior of a spontaneously active neuron is assumed to be the same as a neuron with a high excitatory input.

3.2. Single neuron logic primitives

Fig 1 shows symbols for a few logic primitives. For several reasons that were detailed in [1], networks are illustrated with standard (ANSI/IEEE) logic symbols rather than symbols commonly used in neuroscience schematic diagrams. One of the reasons is that the symbols can be interpreted in two ways. As a logic symbol, the rectangle with one rounded side in Fig 1A represents the AND logic function, and the circle represents negation. The input variables X and Y represent truth values TRUE or FALSE, and the output represents the truth value X AND NOT Y. Second, Fig 1A can also represent a single neuron, with a circle representing inhibitory input and no circle representing excitatory input. If X and Y are binary inputs, the output is X AND NOT Y by Table 1. For the rest of the symbols in Fig 1 and the networks in subsequent figures, the outputs follow from straightforward Boolean logic.

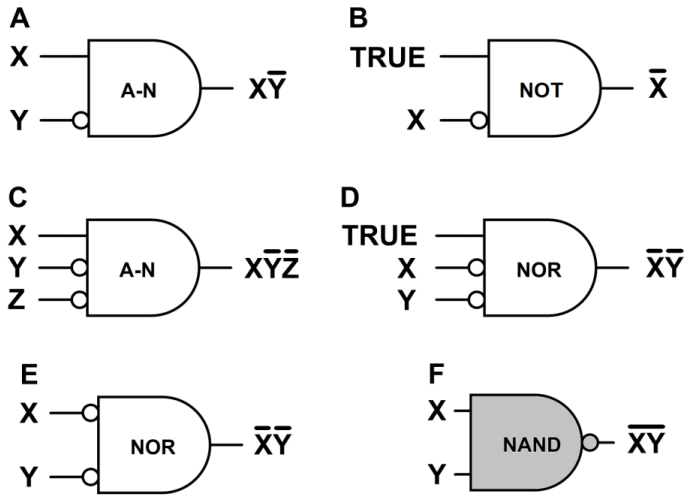

Fig 1. Logic primitives: AND-NOT, NOT, NOR, NAND. Each white symbol can be implemented with a single neuron or with electronic hardware. The gray symbol is

commonly used in electronic logic circuit design. **A.** A logic symbol for an AND-NOT gate, or a neuron with one excitatory input and one inhibitory input. **B.** An AND-NOT gate configured as an inverter, i.e., a NOT gate. **C.** A three-input AND-NOT gate, or a neuron with one excitatory input and two inhibitory inputs. **D.** A three-input AND-NOT gate configured as a NOR gate (NOT OR). **E.** The NOR function implemented with an AND gate, or with a spontaneously active neuron and two inhibitory inputs. **F.** The most common symbol for a NAND gate (NOT AND).

If the first input to an AND-NOT gate is continuously TRUE, as shown in Fig 1B, the output value is NOT X, i.e., the opposite of the input X. Figs 1C and 1D are extensions of Figs 1A and 1B, respectively. The neuron logic for Fig 1C follows from Table 1: if one inhibitory input can suppress one excitatory input, then either one of two inhibitory inputs can suppress the excitatory input. The output for Fig 1D follows from Fig 1C. The neuron output for Fig 1E follows from Fig 1D and the properties that a figure that shows only inhibitory input is assumed to be spontaneously active, and the behavior of a spontaneously active neuron is the same as a neuron with high excitatory input.

The logic primitive NOT(X OR Y) is called the NOR operator (for "NOT OR"). By De Morgan's law, NOT(X OR Y) is logically equivalent to (NOT X) AND (NOT Y). The latter is the output of Figs 1D and 1E. In simplest terms, these neurons are NOR gates because each neuron is active when it has inhibitory input from neither X *nor* Y.

The logic primitive NOT(X AND Y) is called the NAND operator (for "NOT AND"). Fig 1F is the most commonly used symbol for a NAND gate. Like the AND-NOT operator with access to high input, the NOR and NAND operators are

functionally complete. They are also two of the most commonly used building blocks in electronic computational systems.

3.3. JK flip-flops

A flip-flop is a memory mechanism that stores one bit of information in an output that is either 0 or 1. This output is the flip-flop *state* or *memory bit*. A change in the state *inverts* the state. The information is stored by means of a brief input signal that determines the output. A distinction is sometimes made between a "flip-flop" and a "latch," with the latter term reserved for asynchronous memory mechanisms that are not controlled by a clock. The more familiar "flip-flop" is used here.

The JK flip-flop is the most widely used flip-flop design in electronic computational systems because of its advantageous features. It is called a universal flip-flop because it can be configured to function as an SR (set reset) flip-flop, a D flip-flop, or a T (toggle) flip-flop. It is faster than some other flip-flop designs and does not have propagation delay problems. If the inputs are both high simultaneously, rather than causing an error as in a simpler SR flip-flop, the flip-flop state is inverted because one of the inputs is suppressed by one of the outputs.

Fig 2 shows two implementations of the JK flip-flop, which were also given in [3]. The flip-flop's memory bit is labeled M in the diagrams. The inputs S and R are normally low. A brief high input S *sets* the state to M = 1, and a brief high input R *resets* it to M = 0. Feedback to the output gates maintains a stable state after the input returns to low.

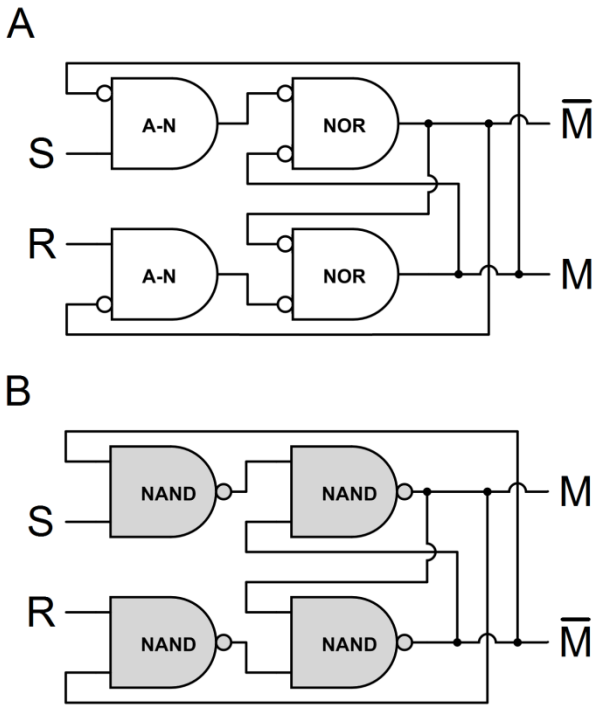

Fig 2. JK flip-flops. A. A JK flip-flop composed of logic gates from Fig 1 that can be implemented with neurons or with electronic components. **B.** One of the standard designs for an electronic JK flip-flop. Each design can be derived from the other by moving each negation circle from one end of a connection to the other. If a circle is moved past a branch point to an output, the output is inverted. Moving the negation circles does not change the logic of the network, but each logic gate in Fig 2A can be implemented with a single neuron from Fig 1.

4. Lamprey central pattern generator

Fig 3A shows a schematic of the CPG that controls the undulatory swimming motion of the lamprey. Fig 3B illustrates the segmental component in standard engineering form with symbols from Fig 1, which can be implemented by neurons or electronic hardware. This figure is a modification of the JK flip-flop in Fig 2A.

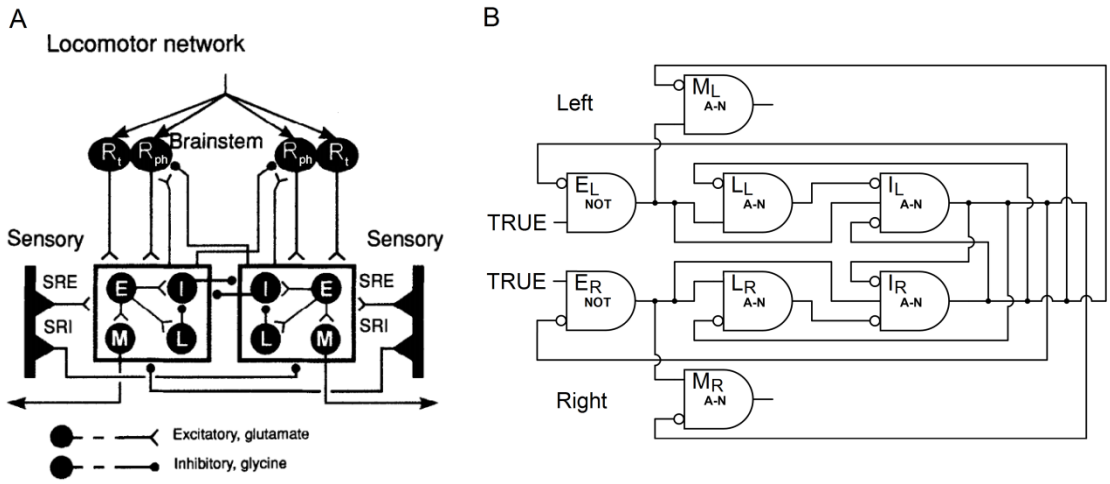

Fig 3. Lamprey CPG that coordinates locomotion. A. The brainstem, sensory, and segmental components that generate bursts (adapted from [11]). A single segmental component is shown in the left and right squares. Each neuron symbol represents a group of neurons. Synapses that terminate at a square affect all neurons in the square. The excitatory neuron (E) in each square excites all of the other neurons in that square. The motor neuron (M) controls the muscles on one side of the segment. The commissural inhibitory neuron (I) inhibits all of the neurons in the contralateral square. The lateral neuron (L) inhibits the ipsilateral commissural neuron. The reticulospinal brainstem neurons are phasic (R_{ph}) and tonic (R_t). The sensory neurons include stretch-receptor neurons that are excitatory (SRE) and inhibitory (SRI). **B.** The segmental component in the squares in Fig 3A illustrated in standard engineering form with symbols from Fig 1. Subscripts indicate the left and right sides.

This study assumes a somewhat simplified segmental component. As stated above in Fig 3A, single neurons represent groups of neurons. The groups' sizes and organizations of synaptic connections may affect the periods, phases, and burst durations of the neurons. These are some of the details that are uncertain [13, 14].

It is also assumed here that only the segmental excitatory neurons (E) receive input from tonic reticulospinal brainstem neurons (R_t), as shown in Fig 3B. Other input to segmental neurons from the brainstem and sensory neurons affect the neurons' oscillation frequency [11], which is proportional to the speed of the lamprey's undulatory wave [11], but it will be shown in the simulations that such external input to the segmental neurons is not necessary for the neurons' states to oscillate.

The network composed of four lateral (L) and inhibitory (I) cells in Fig 3 is a modified JK flip-flop. The only difference between this network and the JK flip-flop of Fig 2A is that the inhibitory cells (I) are AND-NOT cells from Fig 1C rather than the NOR cells of Fig 1E that are shown in Fig 2A. When the excitatory input to Fig 1C is high, as shown in Fig 1D, the cell is enabled to function the same as Fig 1E. Possible reasons for this deviation from the JK flip-flop design in Fig 2A were discussed in the introduction.

This difference from the standard JK flip-flop design makes the burst durations of the I cells one neuron delay time shorter due to the wait for them to be activated by the E cells, and the burst durations of all the other cells one delay time longer due to the wait for the I cells to inhibit them. The modification does not affect the common period of six delay times shown in the next section.

5. Simulations of the lamprey CPG segmental component

Two simulations of the CPG segmental component in Fig 3B were carried out as described in the Methods section.

5.1. Implementation with neurons

Fig 4 shows an Excel simulation of the CPG segmental component implemented with neurons.

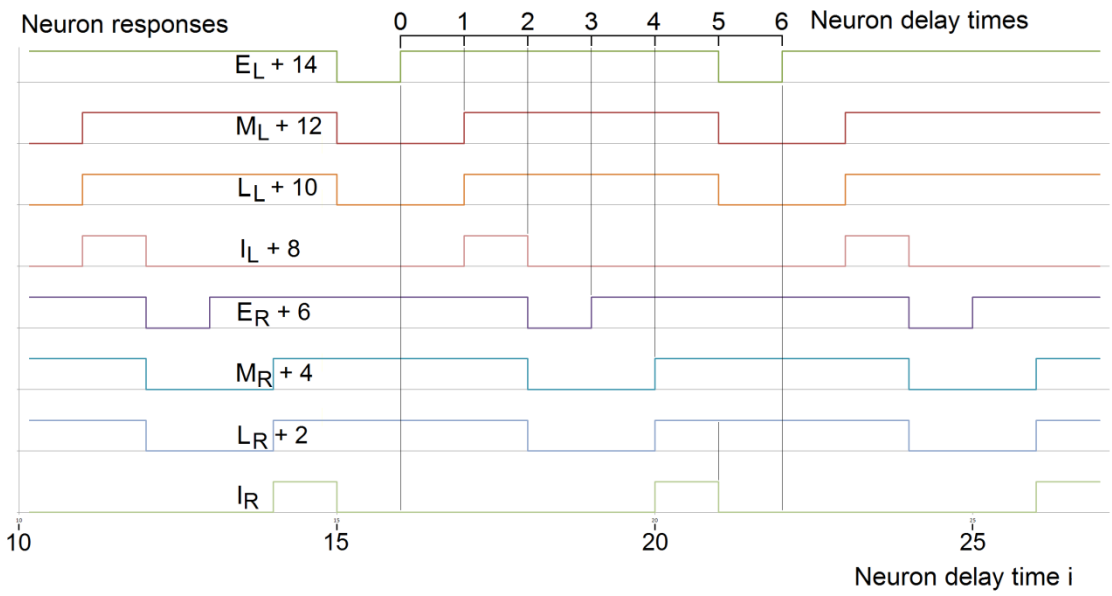

Fig 4. Simulation of the lamprey segmental network in Fig 3B implemented with neurons.

The graphs in Fig 4 show that the common period of the neurons' simulated oscillations is six neuron delay times. The two sets of neuron responses in the left and right sides of the segment have the same oscillations, 180 degrees out of phase. The motor (M) and lateral (L) neurons on each side have identical graphs because they have the same inputs, as shown in Fig 3. The graphs show the state changes after each delay time (as numbered at the top of Fig 4):

0. Uninhibited by I_R, E_L becomes active. I_R remains inhibited by L_R until L_R becomes inhibited and E_R becomes active.

1. E_L excites the other three left cells, which are all unhibited.

2. I_L inhibits the right cells, and L_L inhibits I_L. I_L remains inhibited by L_L until L_L becomes inhibited and E_L becomes active.

3. Uninhibited by I_L, E_R becomes active.

4. E_R excites the other three right cells, which are all unhibited.

5. I_R inhibits the left cells, and L_R inhibits I_R.

6. Uninhibited by I_R, E_L becomes active.

5.2. Implementation with electronic components

To verify the simulated neural implementation in Fig 4, a CircuitLab simulation of an electronic implementation of Fig 3B is shown in Fig 5. Because the AND-NOT gate (Fig 1A) is virtually never used as a building block in logic circuit design, it is normally not provided in simulation software. The AND-NOT gate can be implemented with an AND gate and a NOT gate, as shown in Fig 5A. The simulation software assumes a 10 nanosecond delay for each gate, so buffer gates were added wherever Fig 3B indicates excitatory inputs to make a consistent 20 ns delay time for each simulated neuron. The simulated oscillation period of six neuron delay times is therefore 120 ns, which is shown on the Time axis.

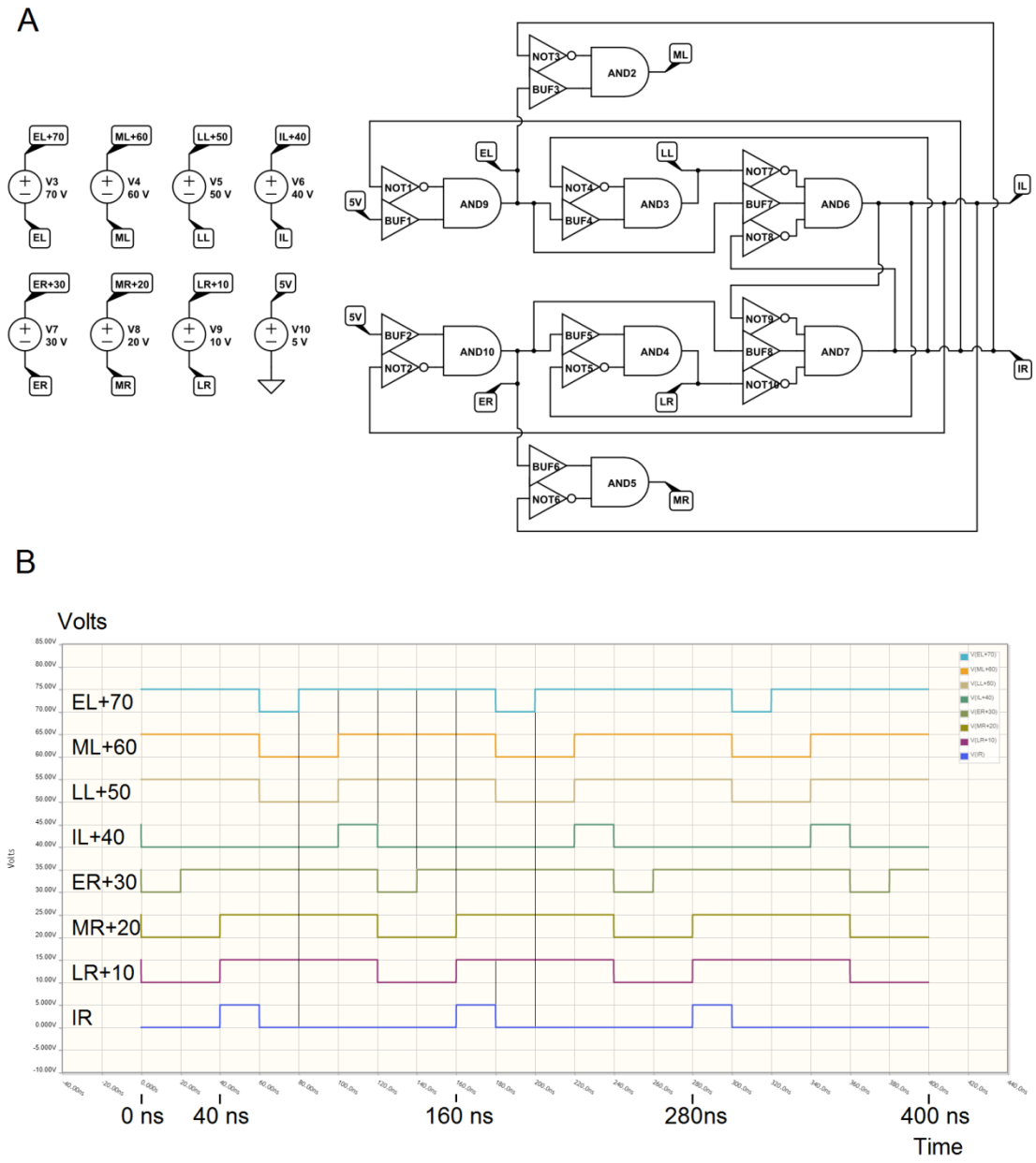

Fig 5. Simulation of the lamprey segmental network in Fig 3B implemented with electronic hardware. **A.** The segmental network in Fig 3B implemented with electronic hardware. **B.** A CircuitLab simulation of the electronic implementation. Logic truth values 0 and 1 are customarily represented by 0V and 5V, respectively. The results are identical to the simulated neural implementation shown in Fig. 4.

6. Acknowledgements

Simulations were done in Excel and CircuitLab. Figures were created in CircuitLab and MS Paint. The author would like to thank David Garmire, Paul Higashi, Anna Yoder Higashi, Sheila Yoder, and especially Ernest Greene and David Burress for their support and many helpful comments.

7. References

1. Yoder L. Neural Flip-Flops I: Short-Term Memory. bioRxiv. 2020 May 24:403196.

2. Yoder L. Neural Flip-Flops II: Short-Term Memory and Electroencephalography. bioRxiv. 2020 June 24:168419.

3. Yoder L. Neural Flip-Flops III: Stomatogastric Ganglion. bioRxiv. 2020 Dec 1: 403154.

4. Yoder L. Relative absorption model of color vision. Color Research & Application. 2005 Aug 1;30(4):252-64.

5. Yoder L. Explicit Logic Circuits Discriminate Neural States. PloS one. 2009 Jan 7;4(1):e4154.

6. Yoder L. Explicit logic circuits predict local properties of the neocortex's physiology and anatomy. PloS one. 2010 Feb 16;5(2):e9227.

7. Ayers J, Carpenter GA, Currie S, Kinch J. Which behavior does the lamprey central motor program mediate?. Science. 1983 Sep 23;221(4617):1312-4.

8. Grillner S, Wallen P, Dale N, Brodin L, Buchanan J, Hill R. Transmitters, membrane properties and network circuitry in the control of locomotion in lamprey. Trends in Neurosciences. 1987 Jan 1;10(1):34-41.

9. Grillner S, Wallen P, Brodin L, Lansner A. Neuronal network generating locomotor behavior in lamprey: circuitry, transmitters, membrane properties, and simulation. Annual review of neuroscience. 1991 Mar;14(1):169-99.

10. Grillner S, Matsushima T. The neural network underlying locomotion in lamprey-synaptic and cellular mechanisms. Neuron. 1991 Jul 1;7(1):1-5.

11. Grillner S, Deliagina T, El Manira A, Hill RH, Orlovsky GN, Wallén P, Ekeberg Ö, Lansner A. Neural networks that co-ordinate locomotion and body orientation in lamprey. Trends in neurosciences. 1995 Jan 1;18(6):270-9.

12. Boyd MR, McClellan AD. Changes in locomotor activity parameters with variations in cycle time in larval lamprey. Journal of experimental biology. 2002 Dec 1;205(23):3707-16.

13. Parker D. Complexities and uncertainties of neuronal network function. Philosophical Transactions of the Royal Society B: Biological Sciences. 2006 Jan 29;361(1465):81-99.

14. Parker D. Neuronal network analyses: premises, promises and uncertainties. Philosophical Transactions of the Royal Society B: Biological Sciences. 2010 Aug 12;365(1551):2315-28.

15. Ekeberg Ö, Grillner S. Simulations of neuromuscular control in lamprey swimming. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences. 1999 May 29;354(1385):895-902.

16. Ijspeert AJ, Kodjabachian J. Evolution and development of a central pattern generator for the swimming of a lamprey. Artificial life. 1999 Jul;5(3):247-69.

17. Kandel E, Schwartz J, Jessell T, Siegelbaum SA, Hudspeth AJ. Principles of neural science. McGraw-Hill Professional. New York, NY. 2013:160.

18. Eggermann E, Bayer L, Serafin M, Saint-Mleux B, Bernheim L, Machard D, Jones BE, Mühlethaler M. The wake-promoting hypocretin–orexin neurons are in an intrinsic state of membrane depolarization. Journal of Neuroscience. 2003 Mar 1;23(5):1557-62.