Design and Simulation of a Quaternary Memory Cell based on a Physical Memristor

Jonathan Taylor and Alberto Nannarelli[(1)]
[(1)]DTU Compute, Technical University of Denmark

***Abstract*—Memristors were theorized more than fifty years ago, but only recently physical devices with memristor's behavior have been fabricated and shipped. In this work, we experiment on one of these physical memristors by designing a memristor-based memory cell, implementing the cell, and testing it. Our experiments demonstrate that the memristor technology is not yet mature for practical applications, but, nevertheless, when production will provide reliable and dependable devices, memristor-based memory systems may replace CMOS memories with some advantages.**

I. Introduction

Memristors were first theorized by Leon O. Chua in 1971 [1], but their physical realization has been impossible until 2008 when HP Labs presented working prototypes of bipolar devices with characteristics similar to Chua's memristors [2].

Memristors can be used in a number of integrated circuits and applications domains. In particular, memristors can be a viable replacement of DRAMs as an ultra low power non-volatile memory since they can be fabricated in high density arrays (less than 50 nm per cell) and can implement multi-valued logic.

This paper presents a possible implementation of a quaternary memory cell by using memristors.

After presenting the basic characteristics of an ideal memristor in Sec. II, we simulate its behavior, and, based on that, we develop a circuit to implement a 4-level memristor-based memory cell in Sec. III. Next in Sec. IV, we present the measurements obtained by the electrical characterization of a physical memristor produced by Knowm [3], and its use in the memory circuit we simulated in Sec. III. Sec. V gives a brief description of the physical circuit. Finally, in Sec. VI we draw the conclusion and discuss the plausibility of using current memristor technology in such a memory circuit.

The experimental results show that, although promising, the actual technology has several shortcomings that make the physical memristor not yet usable in a memory cell.

II. Memristor Basics

The memristor was first theorized in a 1971 paper by Leon O. Chua [1]. The basic characteristics of the memristor is that it changes its "memristance" (unit is Ohm), as charge flows through it:

$$v = M(q) \cdot i$$

Memristance decreases in value as a positive charge is asserted, and increases as a negative charge is asserted, giving it a distinct figure-of-eight input/output hysteresis loop in the V-I plane. As such, information can be stored as a memristance (resistance) in the component.

A model was acquired from [4], which was used to do Spice simulations of an ideal memristor. The ideal memristor has a perfectly symmetrical hysteresis loop, meaning that for a given positive charge, a corresponding negative charge of same magnitude and duration will completely reverse the effects the original positive charge made on the memristor, essentially deleting whatever information was stored.

This behaviour is the basis for the method of storing and reading information in this paper. A simple example can be obtained by connecting a memristor in series with a resistor and by reading the output voltage in the middle point (voltage divider circuit). The input and output simulated waveforms are shown in Fig. 1, where a simple read/write/erase sequence is shown.

In the simulation of Fig. 1, we execute the following sequence of operations. First we read the state (memristance) by sending two pulses – time window [0,100 ms]. The read signal has a counteracting pulse straight after, so the entire read sequence creates a net memristance change of zero. Second, in time window [100, 200 ms], we send a write signal to change the memristance. Third, in time window [200, 300 ms], we send the read signal again and in the output (Fig. 1, bottom) we can notice that the stored value has changed. Fourth, in time window [300, 400 ms], we reset the memristance by sending a negative pulse to "erase" the memristor. Fifth, is time window [400, 450 ms], we read again the value stored in the memristor.

In this simulation, the read signal is of lower magnitude than the write and erase signals. The difference in read signal output between the write and erase pulses is clearly seen, which signifies the memristors change in memristance. What can also be seen from Fig. 1 is that the output signal, and therefore the change in memristance, depends on the amount of time the input signal is asserted. For two different input signals with same magnitude but different duration, the memristor will have two distinct memristive states, assuming a saturation state is not reached.

As such, the memristor can be used for more than just 1-bit (2 state) memory storage, but anywhere in-between as well, depending on how much control one has over the memristor. For this paper, a 2-bit (4 state) memristor memory circuit is designed and simulated. Sending a series of 1V, 10 ms pulses, with 1 ms rise and fall times, through the simulated memristor,

978-1-5090-1095-0/16/$31.00

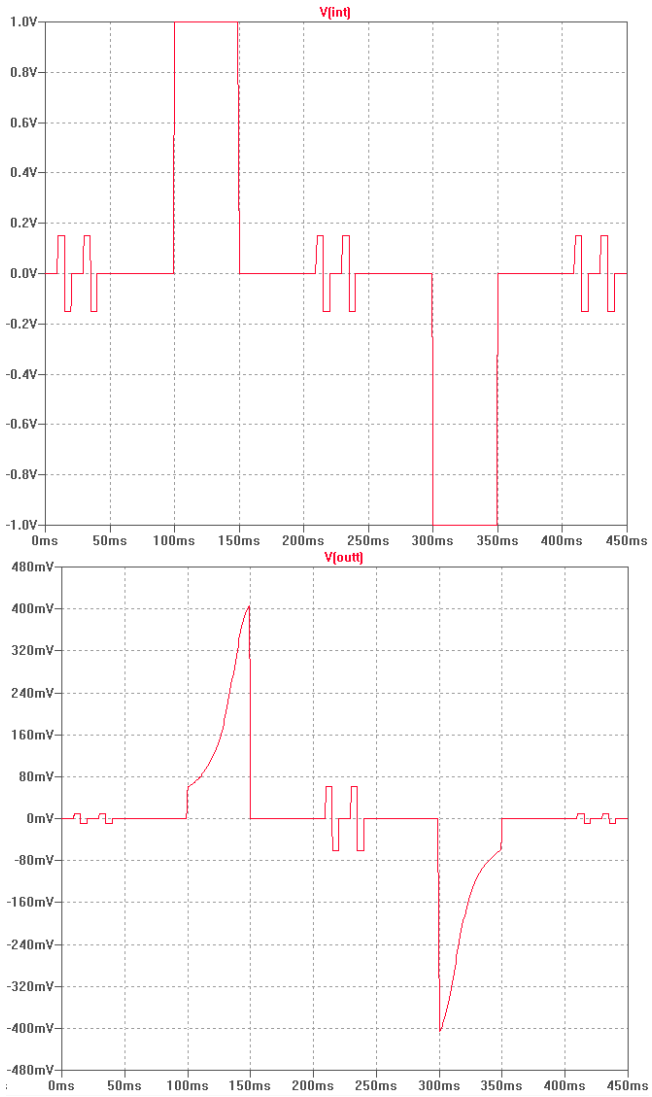

Fig. 1. Spice simulation of read/write sequence. Shows input (top) and output (bottom). The exact sequence is read, read, write, read, read, erase, read, read.

several distinct states were observed, as shown in Fig. 2. Pulses 3 through 5 seem to have the highest impact on memristance, so for the 2-bit memory a 1V, 20 ms initialization pulse is needed before operation, in order to put the memristor in this "optimum write state". To distinguish between these 4 states, three comparators are needed, with their reference voltages in-between the memristive states, also shown in Fig. 2.

III. Usage as a Multi-State Memory Cell

In order to utilize the characteristics of the memristor for memory applications, we design a circuit which is able to translate a 2-bit input into a pulse of corresponding duration (the encoder), and likewise a circuit that can translate the memristance value into a 2-bit value (the decoder).

The circuit, shown in Fig. 3 has the encoder before the memristor, symbol $-\!\!\propto\!\!|\!\!-$, which is surrounded by an H-bridge, so direction of current flow through the memristor can

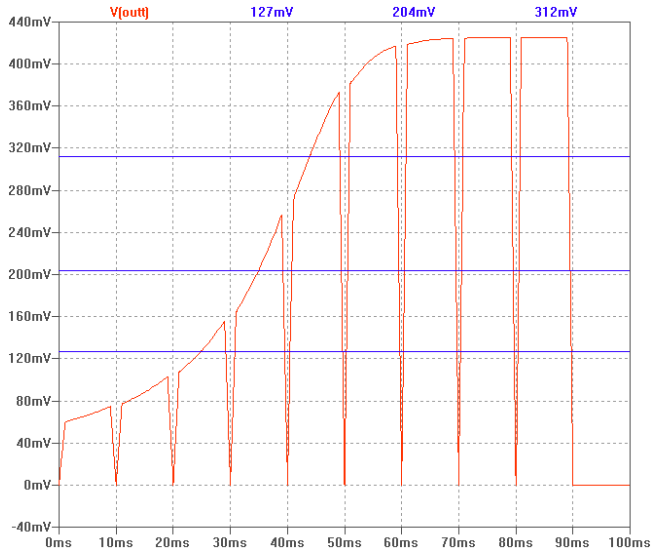

Fig. 2. Spice simulation of memristor output with 10ms, 1V pulses as input. Suggested comparator voltages for 2-bit memory also shown.

be changed. Following is an amplifier and the comparators. Included is a switch which can control when the signal is amplified. Ideally, this switch would only be closed when a read signal is asserted, as the amplifier feeds directly into the comparator stage. Finally is the decoder and two D-flip-flops (DFFs), which are meant to hold the last read value of the memristor.

A feedback loop runs the output values to the input of the encoder, for when data must be erased from the memristor. The switches in the circuit are controlled via a microcontroller, or similar.

In the following, we explain the detail of the different parts of the circuit in Fig. 3.

A. Encoder stage

The encoder takes a 2-bit input and converts it to a pulse of corresponding duration to be fed into the memristor. This is done by first converting the 2-bit input into 000, 001, 011, and 111, respectively, as shown in Table I. Three switches at the output of the encoder are then closed and opened in sequence, so that the correct pulse duration is applied to the memristor. In the simulation, one pulse is 10 ms long.

By Table I, we can derive the encoder logic: the 2-bit value $(I_1 I_2)$ is converted in three pulses P_1 P_2 P_3 as

$$\begin{aligned} P_1 &= I_1 \text{ OR } I_2 \\ P_2 &= I_2 \\ P_3 &= I_1 \text{ AND } I_2 \end{aligned}$$

The same circuit is used when erasing information from the memristor. In this case, the input to the encoder is simply the output from the decoder circuit, reading the memristor (switches S3-S4 closed, and S1-S2 open). The direction through the memristor is reversed, and the encoded signal is applied.

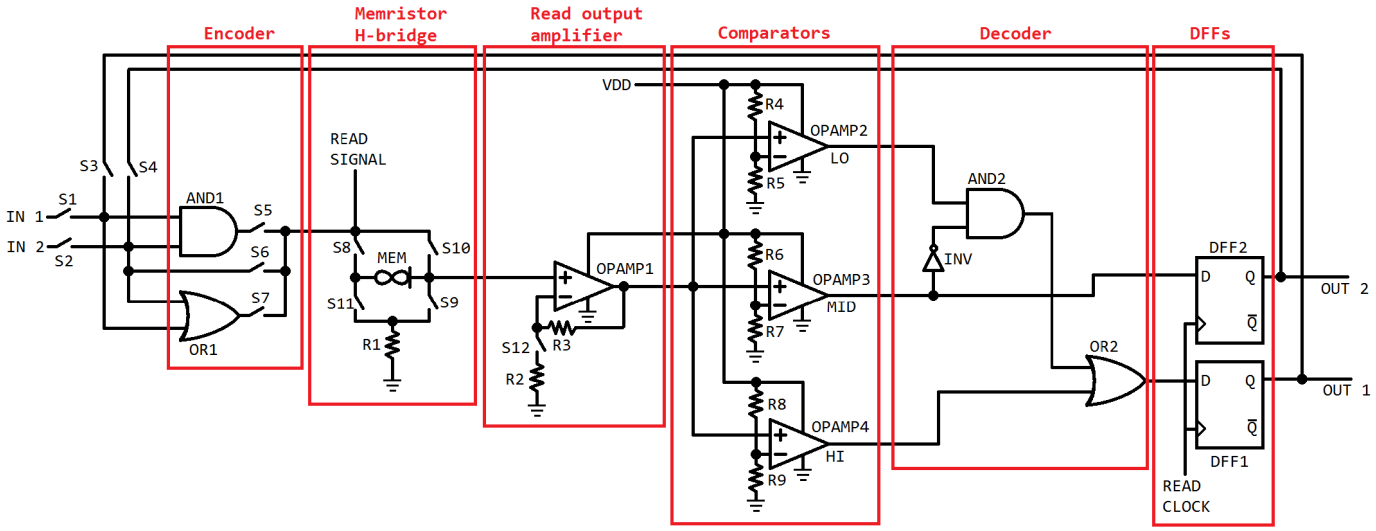

Fig. 3. Memristor —⊲⊲— and surrounding circuit used for quaternary memory.

TABLE I
2-BIT VALUES AND CORRESPONDING MEMRISTOR OUTPUT STATES

2-bit value	Corresponding memristor read state
00	000
01	001
10	011
11	111

B. H-bridge

Surrounding the memristor are four switches (S8–S11 in Fig. 3) in an H-bridge configuration. In order to bring the memristor into a lower memristive state, we have to assert a positive voltage across the memristor, and for a higher memristive state, a negative voltage, however, we cannot simply apply a negative voltage, so instead an H-bridge is used to change direction of flow through the memristor.

C. Comparator stage

The comparator stage consists of three comparators, which read the memristance state. When reading from the memristor, a short read pulse is sent through the memristor. The corresponding output from the memristor is then amplified for the short period the read signal is asserted, and the three comparators determine the state. For the 2-bit case, the different output signals possible are 000, 001, 011, and 111, respectively (Table I). This is then fed into the decoder.

D. Decoder stage

The decoder works similarly to the encoder, but just in reverse. The signals $L_1L_2L_3$ (000, 001, etc.) are converted to a 2-bit value D_1D_2. Looking again at Table I, but going right-to-left, we get the boolean expressions:

$$\begin{aligned} D_1 &= (L_1 \text{ AND } (\text{NOT } L_2)) \text{ OR } L_3 \\ D_2 &= L_2 \end{aligned}$$

This 2-bit output is then stored in two DFFs.

E. Simulation of circuit

To test the circuit, a simple test procedure was set up in Spice. Each time the memristor is written to, a read signal must first be applied, so whatever data may be on the memristor already can be erased. After existing data is erased, the input is encoded and written to the memristor. After this, the value was read and stored in the DFFs. This procedure is repeated 5 times, for the inputs 00, 01, 10, 11, and 00, respectively.

Fig. 4 shows the results. The red line shows the output of the memristor, as seen by the comparator stage. Note how the read signal (spikes above 0.3V in the figure) is amplified, so it is easily distinguished from the write and erase signals. The green and blue lines are the outputs from the DFFs. The signals have been scaled so they all fit within a single plot.

The DFF outputs follow the 00, 01, 10, 11, 00 input, which confirms that the circuit is working as intended.

The method presented here, although quite accurate, is not easily implemented in physical circuits, as it requires the memristor to be 100% consistent for it to work. This is mostly due to how writes work by erasing the exact data already on the memristor.

An alternate way of doing this, which would be safer for physical circuits implementation, is for every write to be preceded by a complete erase and re-initialization of the memristor. Doing this eliminates the necessity for reading the memristor state. Instead, an erase signal is simply applied until the memristor reaches a saturated high-resistance state.

Worth noting also, is that any number of memristance states can theoretically be utilized, however as the number of states increase, so does the complexity of the surrounding circuitry.

IV. THE PHYSICAL MEMRISTOR

An actual memristor is currently being manufactured by Knowm Inc. [3]. In the following we describe the testing of BS-AF-W memristor IC and comment on its performance.

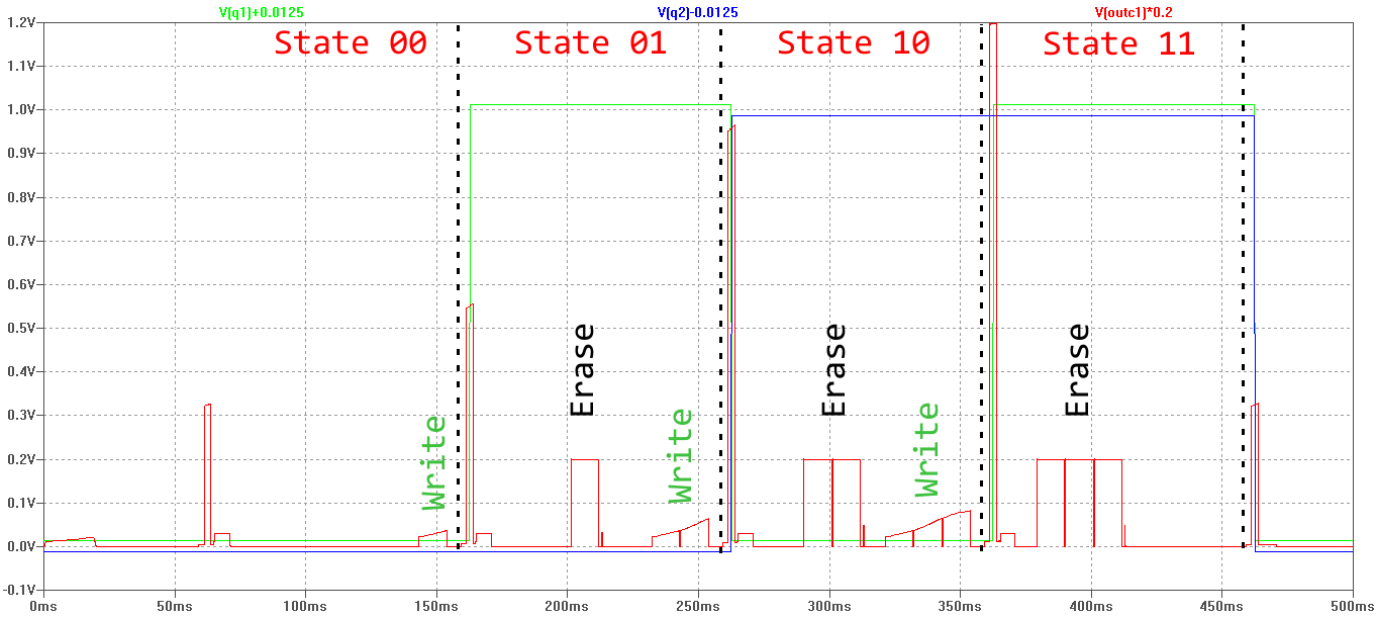

Fig. 4. Simulation results from quaternary memory circuit. Red line shows comparator input, or memristor output, while the green and blue lines shows the DFF outputs (state of the memory cell). Signals are scaled so they fit within a single plot.

Fig. 5. The 16 DIP package containing the 8 discrete Knowm BS-AF-W memristors (source [3]).

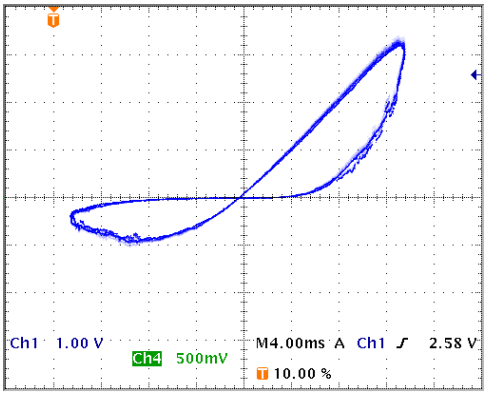

Fig. 6. Hysteresis loop of physical memristor for 3.5V, 50.1kHz input signal. Input on x-axis and output on y-axis.

Eight discrete memristors are packed in a 16-pin DIP package (Fig. 5).

The memristor was set up in a voltage division circuit (in series with a 51kOhm resistor) similar to the one used in the simulations. Fig. 6 shows the hysteresis loop for the physical memristor, for a 3.5V, 50.1kHz input signal, with the input signal on the x-axis and output signal on the y-axis.

The Knowm memristor shows obvious memristor characteristics, with its figure-of-eight hysteresis loop, however it is far from the hysteresis loop of the theoretical ideal memristor.

Most notably, the positive and negative loops are not symmetrical, meaning that applying a positive charge and an equal negative charge will not result in the memristor being in the same state as before. Instead, the corresponding negative charge required to counteract the effects of a positive charge will have to be of different duration. The exact relation between the positive and negative loops should be possible to find through analysis of the two loops given a consistent memristor output, however, as it turns out, the tested physical memristor is very inconsistent.

The frequency response of the physical memristor is consistent with theory, in that the loop gets ”narrower” as frequency is increased. This is directly related to how pulses of different duration will set the memristor in a unique memristive state. Fig. 7 shows memristor hysteresis loops of varying frequency over a longer period of time. The frequencies are, starting in the top left and ending in the bottom right, 5, 10, 20, 50, 100, and 150Hz. What should be immediately obvious is the inconsistency shown in each of these oscilloscope screenshots, which feature several memristor hysteresis loops with the same input signal applied. This is worst at 5Hz, and gets better as frequency is increased.

Disregarding these fluctuations, we see how, in general, the hysteresis loop becomes narrower and of lower magnitude as frequency increases, which is consistent with the ideal memristor model.

An attractive quality of the memristor with regards to

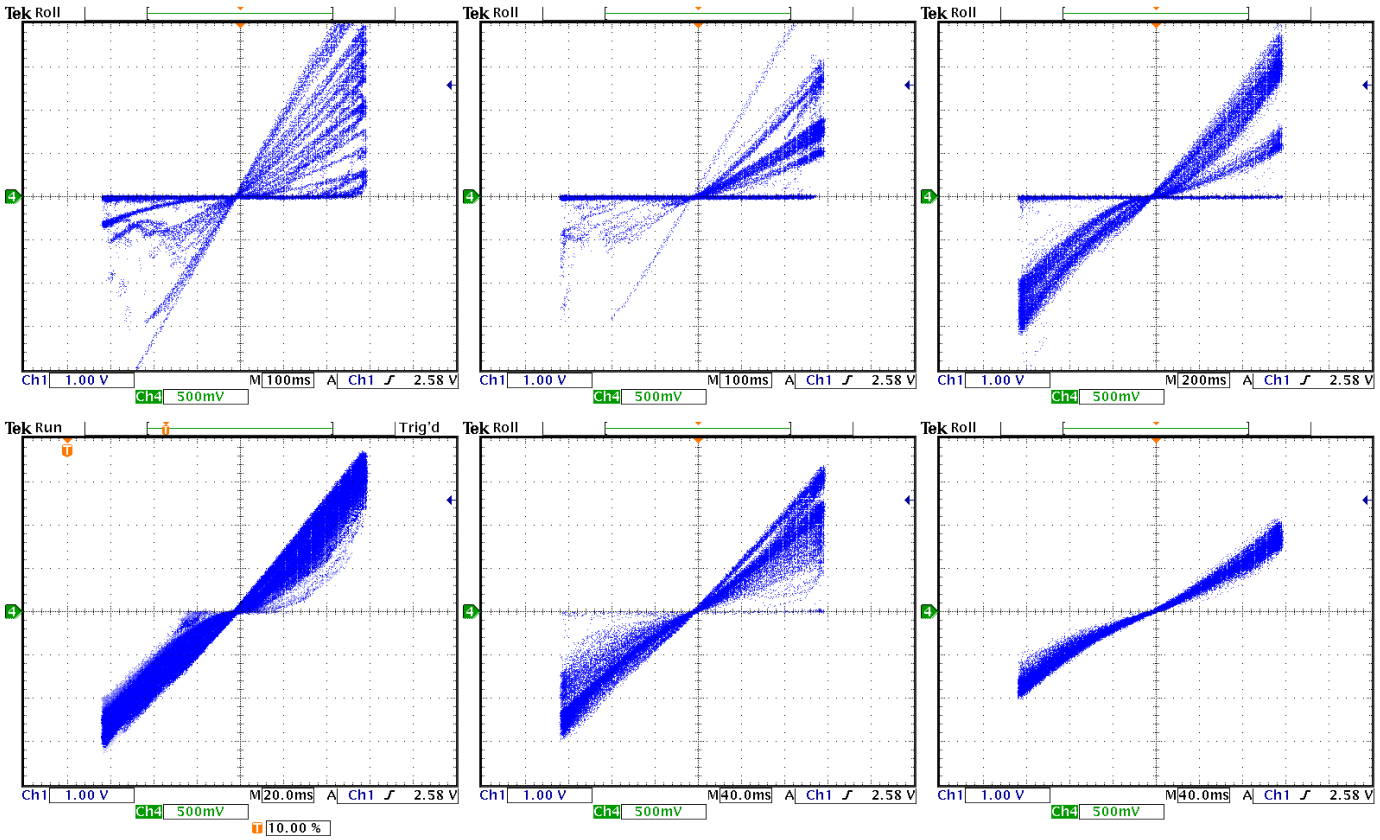

Fig. 7. Several hysteresis loops for a memristor at 5Hz (top left), 10Hz, 20Hz, 50Hz, 100Hz, and 150Hz (bottom right). Input on x-axis and output on y-axis.

memory is its non-volatility. A quick test was set up to test the data retention of the physical memristor. A 2.6V write pulse was asserted to each of the 8 memristors in the IC, followed by a 0.4V read pulse to confirm that the data had been written to them. Of the eight memristors, one failed to retain any data and another two seemingly operated at different voltage levels from the rest, switching to the low resistance state simply from having a read pulse asserted

The five remaining memristors were powered off for approximately half an hour, after which another read was asserted. Of the five, only a single memristor managed to retain data. The one successful memristor had an initial output read voltage of about 0.35V, which dropped very slightly (less than 10%) over the course of the 30 minutes. The rest had varying initial output voltages, but dropped down to 0V when they were read again. The successful memristor was read again after approximately another hour, but had lost its data.

In summary, our test of the physical memristor showed some positive aspects and several shortcomings.

A. The Good

The physical memristor showed a number of characteristics typical of the theoretical memristor: hysteresis loop, frequency dependence, and, to a limited degree, data retention. One of the most important characteristics of the memristor for use in memory cells is its ability to change between memristance states as charge flows through it, and, although unpredictable at times, the physical memristor did show this characteristic.

B. The Bad

The physical memristor is still not reliable in terms of data retention, however, this is acceptable for volatile memories. One memristor retained data for at least half an hour, which is still better than conventional RAM (though not as good as flash memory). Like conventional RAM, this could likely be compensated for with additional refresh circuitry.

Likewise, the hysteresis loop is not a perfect figure-of-eight, but this could also be compensated for, so long the relationship between the positive and negative half-cycle is known.

C. The Ugly

What currently makes the physical memristor unusable is that it is simply unpredictable. Unlike more mature technologies that have well-defined operating behaviour, the memristors tested here behaved differently every time measurements were taken. As a result of a single memristor being inconsistent, all eight memristors, working under same conditions and "history" (charge/discharge cycles), would give a different result. This simply is not usable in the memory circuit described, as very well-defined outputs are needed for each state. Using the memristors for even a simple 1-bit memory could prove difficult.

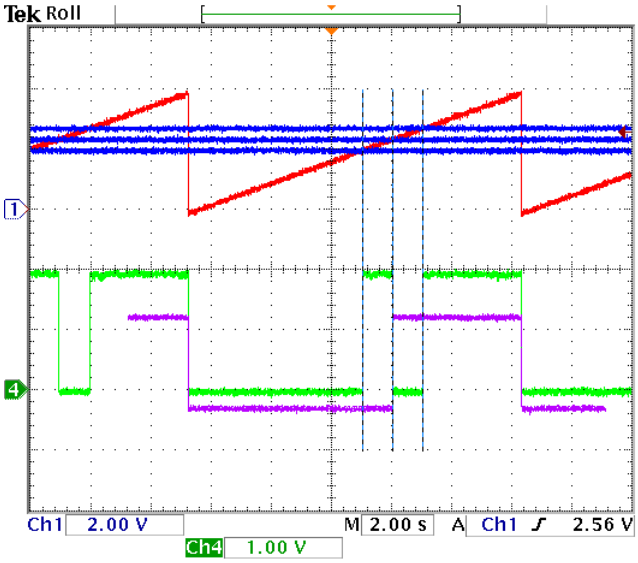

Fig. 8. Input ramp, comparator threshold voltages, and decoder stage output shown in the same image. Vertical lines have been added to show how the intersection of the ramp with the threshold voltages corresponds with the output of the decoder.

V. The Physical Memory Cell

The previously simulated circuit was physically built, to test if it worked.

The comparators used Texas Instruments' LM358 OpAmps, while the encoder/decoder used TI's 4000-series (CMOS) logic gates. An Arduino Uno was used for supplying the power to the circuit, and to control the switches (Fairchild Semiconductors 2N7000 MOSFETs). Of course, the Knowm BS-AF-W was the memristor used.

As already stated, the memristor did not work within the requirements of the circuit, though the rest of the circuit worked as intended. As an example, the output from the decoder is shown in Fig. 8, given a ramped input (red). The three blue lines are the comparator reference voltages, and the green and purple lines are the decoder outputs, which can be seen to follow the 00, 01, 10, 11 sequence. Vertical lines have been added so the comparator crossover points can be more easily recognized.

Comparator, encoder, and decoder circuits worked as expected. The memory cell may one day be realizable if the memristor reaches an acceptable level of reliability.

VI. Conclusions

The memristor seems like a promising technology. The simulated circuit worked as hoped, and showed that the memristor can indeed be used as an alternative to conventional data storage methods.

Unfortunately, the physical memristor tested is not reliable enough to implement the proposed memory cell. However, the physical memristor did show most of the characteristics one would expect from a memristor: a pinched, figure-of-eight hysteresis loop, along with the ability to *"store"* resistance. These qualities were not very defined for the physical memristor though, and would seemingly always be slightly different whenever they were measured, which made actually using it in the physical circuit almost impossible.

The memristor technology seems not to be ready yet for actual products. The Knowm website did warn that the IC is for *"preliminary feasibility evaluation"* only, so perhaps the results of the tests are not too unexpected. This is not new considering that when Intel introduced MOS technology in memory chips in 1970, the yield was less than 10% and the device suffered from voltage sensitivity [5].

According to the news, even HP Labs are having troubles with the technology. In a 2010 article [6], it is mentioned that HP believed to revolutionize the memory market by using memristors within "three years time" (some day in 2013). A more recent article, from 2015, says that HP still are not able to produce their memristors "in commercial quantities" [7]. HP are reportedly still developing their memristors, however their application is nowhere near what HP originally imagined, in the 2010 article.

In conclusion, the inconsistencies in the memristor makes it very difficult to implement the physical memory cell. Given a more predictable component, it would have been possible, however, in the current state, the memristor is not suitable.

Instead, it would be worth trying for something less aggressive. A simple 1-bit application using the memristor has been tested, and worked to some degree, but still with inconsistencies. The next step would be to try for 3-state memory, although the cell may result unpractical in a power-of-two dominated world.

The non-volatile quality of the ideal memristor is not quite present in the tested physical memristor, but it still showed signs of information retention.

References

[1] L. Chua, "Memristor - The missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep. 1971.

[2] J. Beckett. (2008, Apr.) "Demystifying the memristor: Proof of fourth basic circuit element could transform computing". [Online]. Available: http://www.hpl.hp.com/news/2008/apr-jun/memristor.html

[3] Knowm Inc. (2015) "BS-AF-W 16DIP Packaged Memristors". [Online]. Available: http://knowm.org/product/bs-af-w-memristors/

[4] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable SPICE Simulations of Memristors, Memcapacitors and Meminductors," *Radioengineering*, vol. 22, no. 945, Jul. 2013. [Online]. Available: http://arxiv.org/pdf/1307.2717.pdf

[5] G. Sideris, "The Intel 1103: The MOS memory that defied cores," *Electronics*, pp. 108–112, 23 Apr. 1973.

[6] J. Markoff, "H.P. Sees a Revolution in Memory Chip," *The New York Times*, 7 Jul. 2010. [Online]. Available: http://www.nytimes.com/2010/04/08/science/08chips.html

[7] Q. Hardy, "HP Destroys a Dream Computer to Save It," *The New York Times*, 3 Jun. 2015. [Online]. Available: http://bits.blogs.nytimes.com/2015/06/03/hp-destroys-a-dream-computer-to-save-it/