Towards a Memristor Model Library in Modelica

Kristin Majetta[1] Christoph Clauss[1] Torsten Schmidt[2]
[1] Fraunhofer Institute for Integrated Circuits IIS, Design Automation Division EAS Dresden
Zeunerstrasse 38, D-01069 Dreden, GERMANY
[2] Technische Universität Dresden, Faculty of Electrical and Computer Engineering
Helmholtzstrasse 18, D-01062 Dresden, GERMANY
{kristin.majetta, christoph.clauss}@eas.iis.fraunhofer.de Torsten.Schmidt1@tu-dresden.de

Abstract

The Modelica realization of two memristor modeling approaches is presented which is compatible to the Modelica Electrical Analog Library. Circuit examples of some basic cases of application are simulated. Comparisons with published simulation results show the correctness of the numerical realizations. The models are the base of a Memristor Model Library.

Keywords: Memristor, window function, modeling, pinched hysteresis, resistive switches, numerical simulation of electronic devices

1 Introduction

The memristor is a special kind of resistor with memory. Therefore, the term “memristor” is composed by parts of both words “memory” and “resistor”. The theoretical concept of a memristor was published first in 1971 by Leon O. Chua [3]. After Strukow et alt. [5] observed that certain nanoscale devices with thin semiconductor layers can be described as memristors, an intensive investigation started on both how a memristor works and how it can be utilized in electronic circuits. Since then, the memristor has a wide attention in the research community of electrical engineers, physicists, and biologists. Recent investigations are mainly focused on resistive random access memories. The advantage of memristors (or more general of memristive devices) is to store information without any power soure to be needed. This could open a new paradigm in power saving computation as well as low power storage. Other fields of research are neuromorphic systems, memristor circuits theory, and applied analog memristor circuits. It can be expected that further interesting fields of both research and application will be opened up in future.

Simulation has been applied since the very beginning of integrated circuit development, so it does for memristor circuits. Therefore, memristor modeling became necessary, and simulation models were published which can be used in different simulation tools. E.g. MATLAB models use a state equation based approach, but models for SPICE need a combination of built-in SPICE models which realize the memristor behavior.

This paper deals with the adaption of published memristor models to Modelica. It is the first step towards the general aim to create a library for memristors, and memristive systems (memcapacitors, meminductors, memristive systems with more than one state [6]). The library will allow to investigate memristor applications on the one hand on circuit level, and on the other hand in the context of arbitrary Modelica applications. In section 2 two different models are presented. Simulation results using Dymola are shown in section 3.

2 Model Approaches

Once memristor measurement data were available several model approaches were elaborated, and adapted to the data. Two very basic models are presented in this section.

2.1 A Basic Model Approach on Varying Resistance

Basing on the physical device structure the authors of [1], [2], [5] introduce a memristor model as a resistor with varying resistance R_{MEM}

$$v(t) = R_{MEM}(x)i(t) \qquad (1)$$

which depends on x linearly and changes between R_{ON} and R_{OFF} :

$$R_{MEM}(x) = R_{ON}x + R_{OFF}(1-x) \tag{2}$$

The structure of such a device is depicted in figure 1 and consists of a partly doped TiO_2-layer which is together with the undoped part sandwiched between two Pt-electrodes. With w being the length of the doped region (Figure 1), and D the total length of doped and undoped region, the state x is defined as

$$x = w / D \tag{3}$$

The doped region is highly conducting (R_{ON}) whereas the undoped region is less conducting (R_{OFF}). Taking into account the length of both regions equation (2) represents a series connection of the actual resistances of both the doped and undoped region.

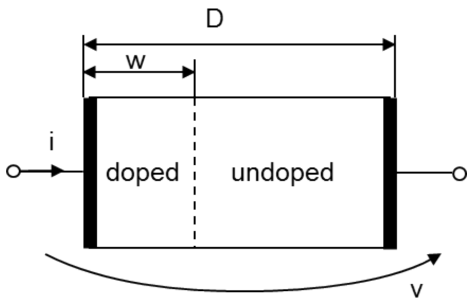

Figure 1 Physical Memristor Schematic

The state equation (4) describes the speed of the boundary movement, which depends on the current $i(t)$, the resistance R_{ON} of the doped region, the dopant mobility μ_v and a window function $f(x)$:

$$\frac{dx}{dt} = \frac{\mu_v R_{ON}}{D^2} i(t) f(x) \tag{4}$$

The so-called window function establishes a nonlinear drift behavior of the device and hence provides a second order approximation to the real device. According [3] one possible window function is

$$f_{Jog}(x) = 1-(2x-1)^{2p} \tag{5}$$

whereas the authors of [1] propose an improved window function f_{Bio} which overcomes sticking of x at 0 or 1 whenever x reaches one of these values. The proposed window function depends on both x and the sign of the current $i(t)$:

$$f_{Bio}(x) = 1-(x - stp(-i(t))^{2p} \tag{6}$$

$$stp(i(t)) = \begin{cases} 1 & i(t) \geq 0 \\ 0 & i(t) < 0 \end{cases} \tag{7}$$

The equations (1) to (7) are the base for writing a simulation model. One possibility is constructing a SPICE subcircuit out of SPICE basic components (macromodeling) according to [1]. This subcircuit could be copied to Modelica using the Modelica.Electrical.Spice3 package. A more convenient way is using the declarative behavioral modeling capability of Modelica. This leads straight forward to the following model, called Memristor_Biolek2009:

```
model Memristor_Biolek2009
import ME = Modelica.Electrical;
import SI = Modelica.SIunits;
extends
      ME.Analog.Interfaces.OnePort;
parameter SI.Resistance RINIT, RON,
                    ROFF;
parameter SI.Length D;
parameter Real muev;
parameter Integer P;
SI.Resistance
      RMEM(start=RINIT,fixed=true);
SI.Length w;
Real x, k, f;
equation
RMEM = RON*x +ROFF*(1-x);
x = w/D;
v = RMEM*i;
der(x) = k*i*fBio;
k = (muev*RON)/(D^2);
//fJog = 1-(2*x-1)^(2*P);
fBio = 1 - (x - stp(-i))^(2*P);
end Memristor_Biolek2009
```

The special sign function is

```
function stp
input Modelica.SIunits.Current i;
output Real value;
algorithm
  value:=if (i<0) then 0 else 1;
end stp;
```


DOI 10.3384/ecp12076507

An icon, default values for parameters as well as assertions complete the model.

2.2 An Improved Approach

The authors of [6] propose an improved approach with more parameters than the model already presented which can be better adapted to given memristor characteristics, e.g. measured data.

Like the Memristor_Biolek2009 model this model has a state variable $x(t)$ for calculating the conductivity according to

$$i(t) = \begin{cases} a_1 x \sinh(bv(t)) & v(t) \geq 0 \\ a_2 x \sinh(bv(t)) & v(t) < 0 \end{cases} \qquad (8)$$

with a_1, a_2, and b being adjustable parameters. The state equation for x is:

$$\frac{dx}{dt} = g(v(t)) f(x) \qquad (9)$$

Whereas $g(t)$ is a threshold function which ensures a state changes only if thresholds are exceeded:

$$g(v(t)) = \begin{cases} A_p(e^{v(t)} - e^{v_p}) & v(t) > v_p \\ -A_n(e^{-v(t)} - e^{v_n}) & v(t) < -v_n \\ 0 & else \end{cases} \qquad (10)$$

The window function $f(x)$ expresses the effect that it is harder to change the state near the boundaries, taking into account the polarity. Parameters are introduced to be able to fit to measured values.

$$f(x) = \begin{cases} e^{\alpha_p(x - x_p)} w_p(x, x_p) & x > x_p \\ e^{\alpha_n(x + x_n - 1)} w_n(x, x_n) & x < 1 - x_n \\ 0 & else \end{cases} \qquad (11)$$

$$w_p(x, x_p) = \frac{x_p - x}{1 - x_p} + 1 \qquad (12)$$

$$w_n(x, x_n) = \frac{x}{1 - x_n} \qquad (13)$$

The equations (8) to (13) can be formulated in Modelica as they are. This leads to the second memristor model, called Memristor_Yakopcic2011:

```
model Memristor_Yakopcic2011
import ME = Modelica.Electrical;
import SI = Modelica.SIunits;
extends
ME.Analog.Interfaces.OnePort;
parameter Real Ap, An;
parameter SI.Voltage Vp, Vn;
parameter Real xp, xn, ap, an;
parameter SI.Current a1, a2;
parameter Real xinit;
parameter SI.InversePotential b;
Real gV, fx, wp, wn;
Real x( start=xinit, fixed=true);
equation
i = if(v>=0)then a1*x*sinh(b*v)
          else a2*x*sinh(b*v);
gV = if(v>Vp)then Ap*(exp(v)-
                  exp(Vp))
   elseif (v<-Vn) then
          -An*(exp(-v)-exp(Vn))
   else 0;
fx = if(v>0 and x>= xp) then
          exp(-ap*(x - xp))*wp
   elseif (v>0 and x<xp) then 1
   elseif (v<0 and x<=1-xn) then
               exp(an*(x+xn-1))*wn
   else 1;
wp = (xp - x)/(1 - xp) + 1;
wn = x/(1 - xn);
der(x) = gV*fx;
end Memristor_Yakopcic2011
```

3 Test And Application Examples

3.1 Memristor Characteristic Using One Input Voltage Pulse

The first example shows the Memristor_Biolek2009 characteristic using a simple voltage pulse.

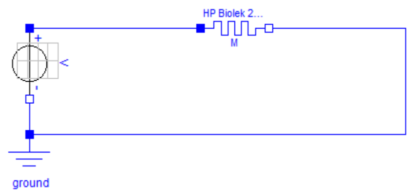

Figure 2 Memristor test circuit

RINIT	RON	ROFF	D	muev	p
11000	100	16000	1e-8	1e-14	10

If the above mentioned parameters are used, the voltage pulse of Figure 3 causes the current-voltage hysteresis of Figure 4. The reason for the hysteresis is increasing of the doped region length as long as a positive current is flowing which increases the overall resistance. Figure 5 shows the change of the state x which influences the resistance. This illustrates that the state is the "memory" of the memristor. The initial state is caused by the initial value RINIT. No special hysteresis model is used, only changing the state causes the hysteresis.

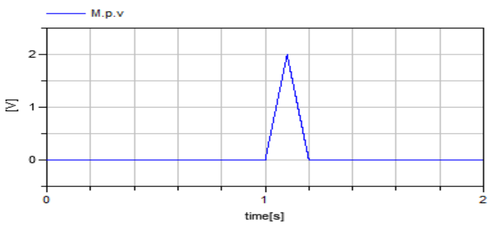

Figure 3 Single voltage pulse

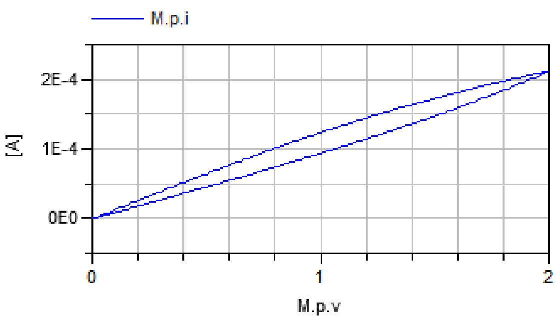

Figure 4 Current-Voltage Hysteresis

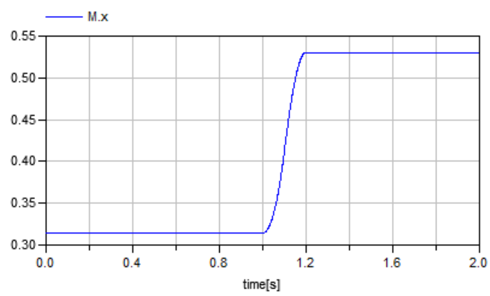

Figure 5 State change due to the voltage pulse

3.2 Memristor Characteristic Using Sinusoidal Input Voltage

To compare the Memristor_Biolek2009 characteristic with the results published in [1] a circuit like in Figure 2 is simulated using a sinusoidal voltage input (1.2 V amplitude, 1 Hz). The memristor parameters are the same as in the example 3.1. Figure 6 shows both the input voltage, and the resulting current.

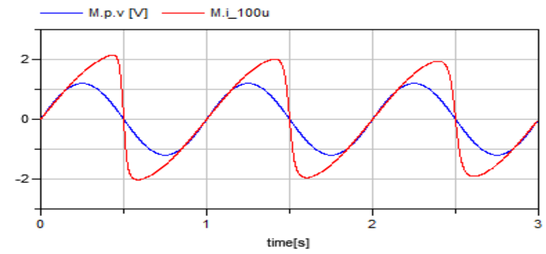

Figure 6 Input voltage and current (raised by 100), with RINIT=11000 Ohm

This result differs slightly from the result published in [1] depicted in Figure 7 due to initial transient effects. In the steady state which can be reached by changing the initial resistance to RINIT=11500 or by simulating over a long time period the published results are reached (Figure 8, Figure 9). The reason for these differences in initialization can depend on different numerical algorithms, and on different error bounds. This has to be investigated in future.

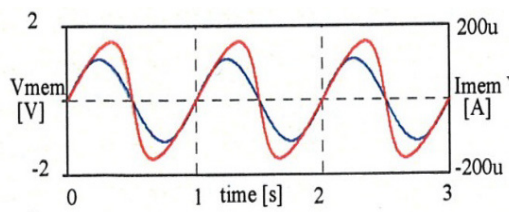

Figure 7 Reference result according to Figure 6, published in [1] (input voltage and current)

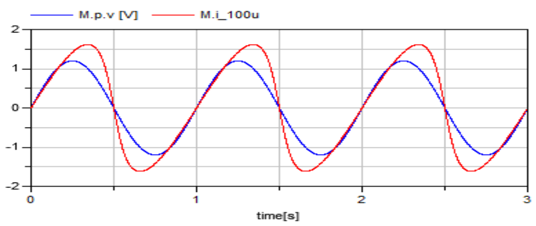

Figure 8 Input voltage and current (raised by 100), with RINIT=11500 Ohm

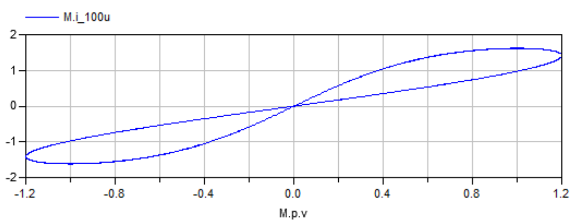

Figure 9 Current-voltage characteristics (current raised by 100), with RINIT=11500 Ohm

DOI 10.3384/ecp12076507

Due to the Memristor_Biolek2009 model, formula (7), the window function $f(x)$ is discontinuous. According to (4) jumping of $f(x)$ influences the derivative of x but not x itself. Therefore, such discontinuities did not yet lead to simulation difficulties in the investigated examples.

3.3 Memristor Characteristic Using Multiple Input Voltage Pulses

The simple circuit according to Figure 2 is used to check the characteristic of the Memristor_Jakopcic2011 model. The results are compared with Fig. 4 in [5]. The memristor model parameters are:

ap	an	Ap	An	Vp	Vn
1	4	0.1	10	0.9	0.2
xp	xn	a1	a2	b	xinit
0.15	0.25	0.076	0.06	3	0.001

The following simulation results (Figure 10, Figure 11) achieved with Dymola are the same as in the reference simulation. The memory effect can be seen. According to this first check the model seems to be correct.

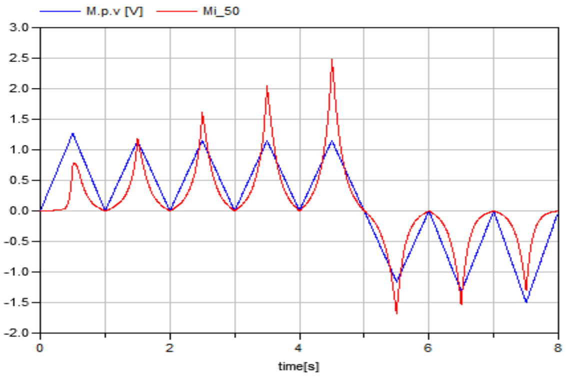

Figure 10 Current-voltage characteristic (current raised by 50), Jakopcic2011 model

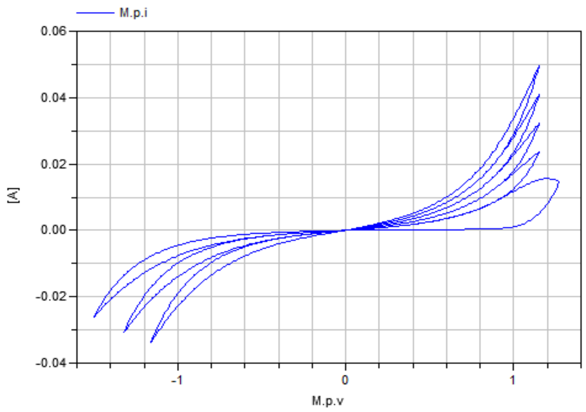

Figure 11 Current-voltage hysteresis, Jakopcic model

3.4 Graetz Rectifier Circuit

Figure 12 shows a Graetz rectifier circuit which uses memristors instead of diodes. It is easily combined using the presented memristor_Yakopcic2011 model as well as MSL components. The memristor parameters are the same as in the previous section.

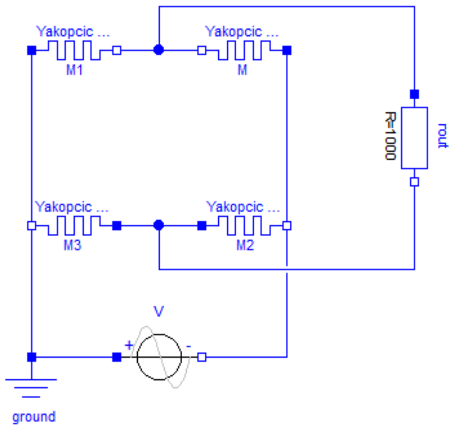

Figure 12 Graetz circuit using memristors

Both the rectified and the original voltage can be seen in Figure 13. Deeper investigation shows that the amplitude of the rectified voltage depends on the amplitude of the input voltage as well as on the frequency. Higher frequency causes smaller rectified voltage amplitudes. If the input voltage is too high the simulation fails. The reasons of that seems to be extremely increasing of exponential functions. Therefore, the model must be improved in future to become more stable.

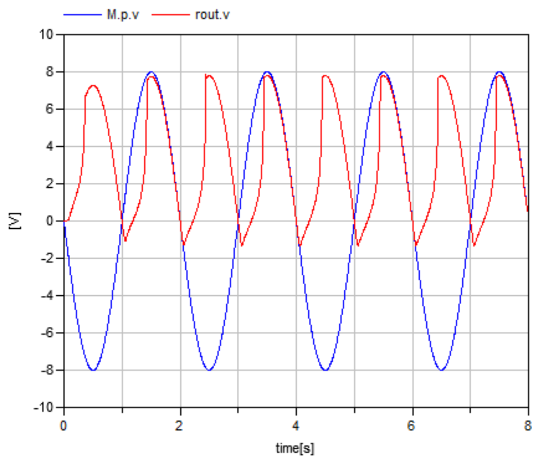

Figure 13 Input voltage, and rectified voltage

4 Conclusions

Two memristor models developed from given memristor model equations are presented. Simple tests show the correctness of the models compared with published simulation results. In some cases small differences occur that have to be investigated in future. Tests with different simulation tools are still necessary, which cover extreme application scenarios.

The Modelica approach of memristor modeling is promising. The memristor models can easily be combined to existing models of the Modelica Standard Library. It is planned to develop a package with numerical stable, and well tested models of memristors, and of other memristive systems like memcapacitors, meminductors, or systems with more than one states. This will allow to study memristors and memristor application circuits in a convenient way.

References

[1] Biolek, Z.; Biolek, D.; Biolkova, V.: Spice Model of Memristor With Nonlinear Dopant Drift. Radioengineering, vol. 18, no. 2, pp. 210-214, 2009.

[2] Biolkova, V.; Kolka, V.; Biolek, Z.; Biolek, D.: Memristor modeling based on its constitutive relation. Proc. of the European Conference of Circuits Technology and Devices (ECCTD). pp. 261-261, 2010.

[3] Chua, L. O.: Memristor – the missing circuit element. IEEE Trans. Circuit Theory vol. 18, no. 5, pp. 507-519, 1971.

[4] Joglekar, Y.; Wolf, S.: The elusive memristor: properties of basic electrical circuits. Eur. J. Phy., vol. 30, pp. 661-675, 2009.

[5] Strukow, D.; Snider, D.; Stewart, D.; Williams, R.: The missing memristor found. Nature, vol. 453, pp. 80-83, Mai 2008.

[6] Ventra, M. D.; Pershin. Y. V.; Chua, L. O.: Circuit elements with memory: memristors, memcapacitors and meminductors. Proceedings of the IEEE 97, 1717 (2009), über arXiv:0901.3682v1 [cond-mat.mes-hall]

[7] Yakopcic, C.; Taha, T. M.; Subramanyam, G.; Pino, R. E.; Rogers, S.: A Memristor Device Model. IEEE ELECTRON DEVICE LETTERS, vol. 32, pp 1436-38, Oct. 2011.

DOI
10.3384/ecp12076507