Reconfigurable Analog Electronics using the Memristor*

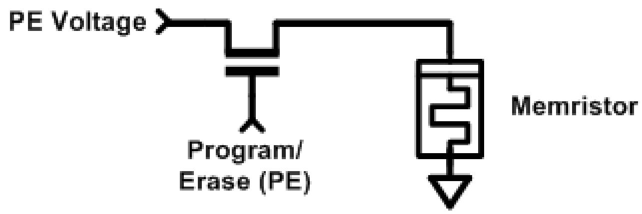
R. Jacob Baker and Kristy A. Campbell
Department of Electrical and Computer Engineering
jbaker@boisestate.edu

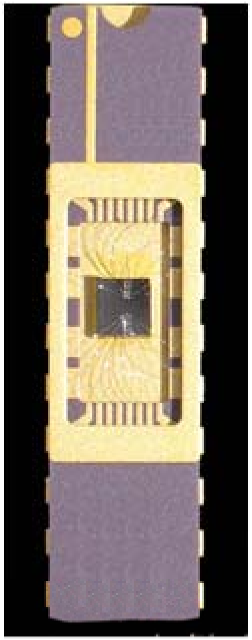
- Practical reconfigurable analog design using the Memristor
  - Minimize the stress across the Memristor device
  - Programming/erasing the Memristor must be simple and reliable
- Biggest potential impact is found in circuits for
  - Analog trimming
  - Data conversion
  - Communications
  - Compensation for physical variations (temperature, sensor conditioning, etc.)

*This work supported by the Air Force Research Laboratory

Programming/Erasing the Memristor

- Drive PE (below) high
  - To erase, connect PE Voltage to a negative potential, for example, < -250 mV
  - To program connect PE to a positive potential > 250 mV



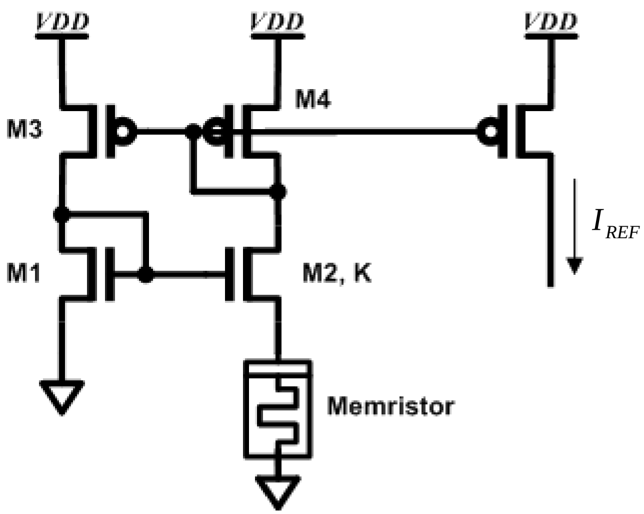


Package containing 12 Memristors fabricated at Boise State University.

Key Points

- Resistance of the Memristor can be scaled downwards by increasing cross-sectional device area
- Larger area results in more consistent devices
- Memory resistance retention improved by minimizing the voltage across the device.
  - Ideally voltage across the device is 0 when using the device!

Basic Beta-Multiplier Reference



M2 is made K-times wider than M1, in other words its Beta is multiplied up hence the name Beta-Multiplier Reference (BMR).

$$I_{D1} = I_{D2} = I_{D3} = I_{D4} = I_{REF}$$

$$V_{GS1} = \sqrt{\frac{I_{D1}}{2\beta}} + V_{THN} = V_{GS2} + I_{D2} R_{memristor}$$

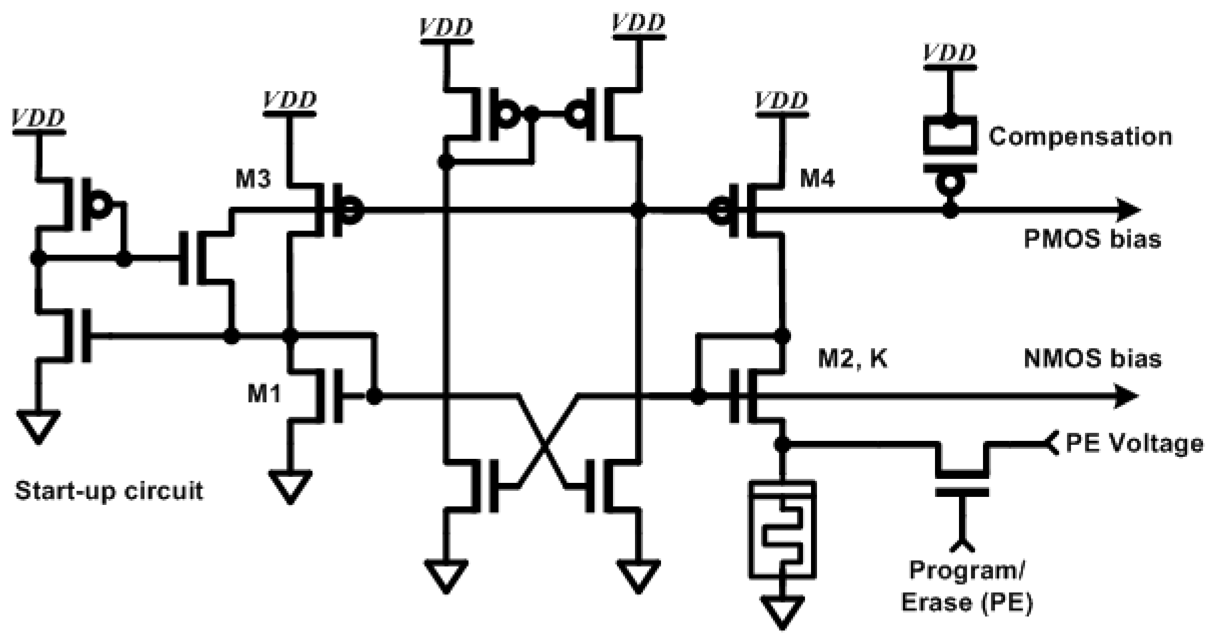
$$I_{REF} = \frac{2}{R_{memristor}^2 KP_n \cdot \frac{W_1}{L_1}} \left(1 - \frac{1}{\sqrt{K}}\right)^2$$

Independent of *VDD*!

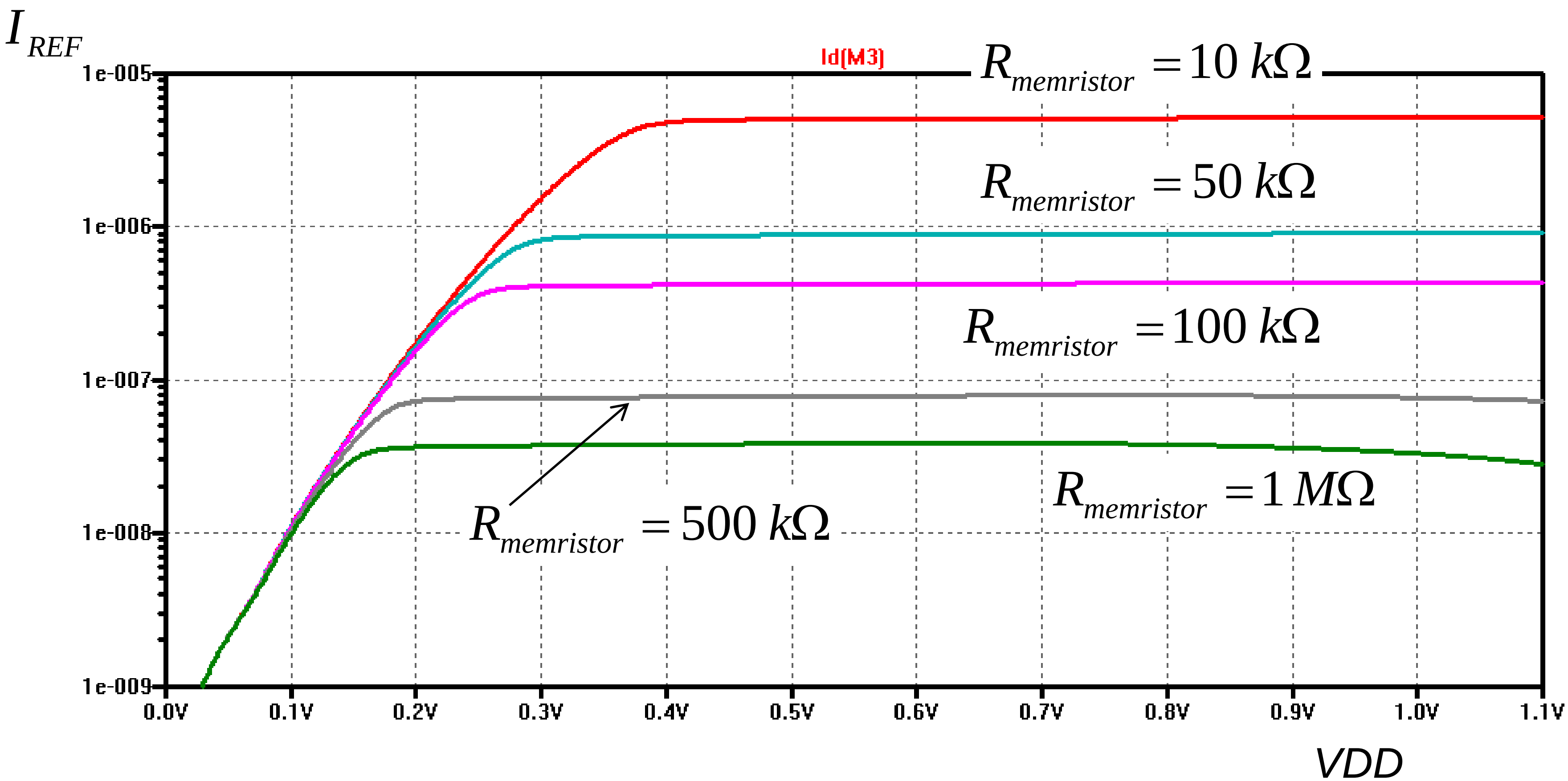
Dependent on $R_{memristor}$

Nanometer CMOS BMR Reference using the Memristor

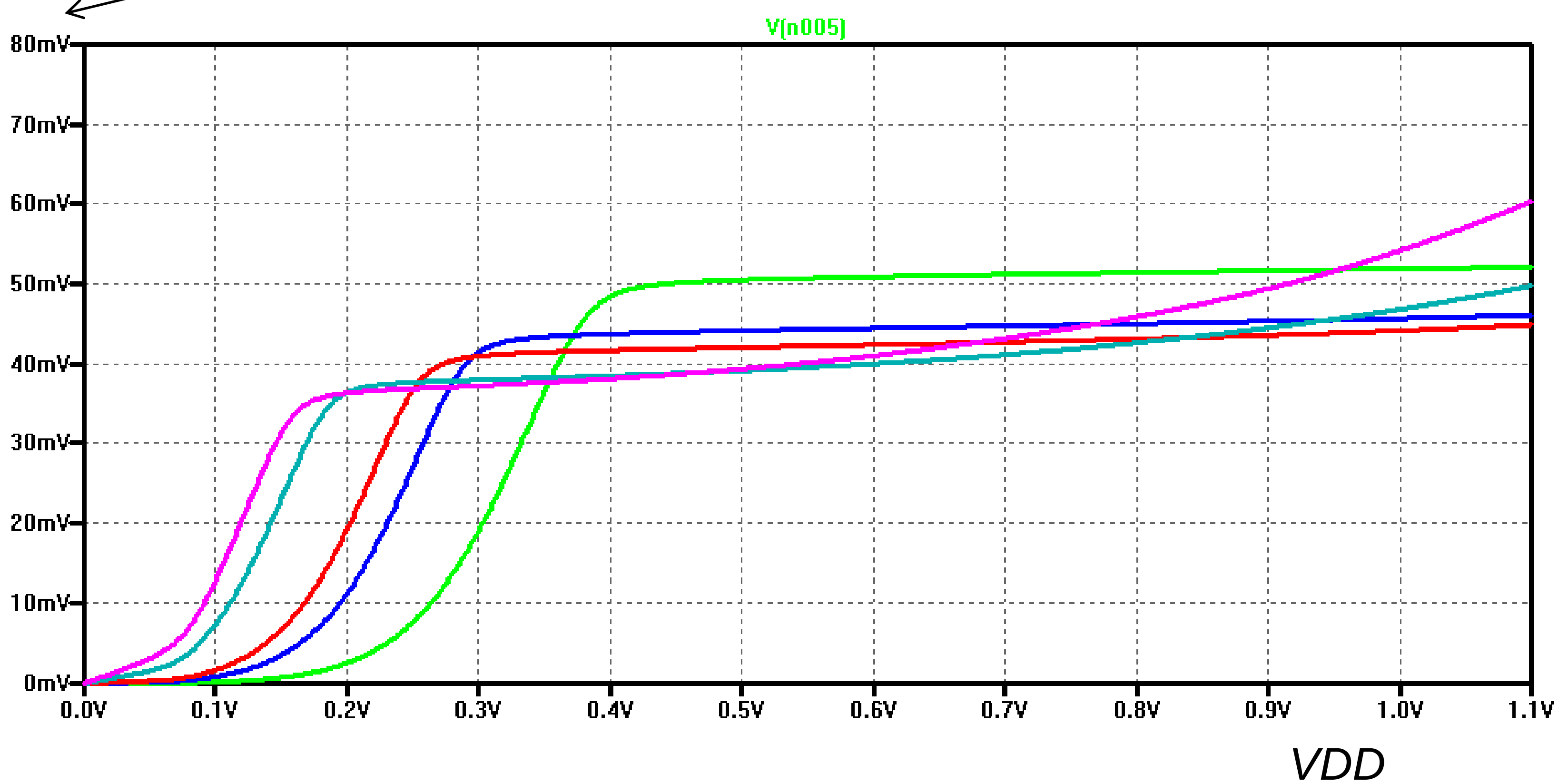
- Add amplifier to ensure better power supply insensitivity
- Add start-up circuit
- Program/Erase by driving PE signal high and applying a "PE Voltage"
- Do we minimize the voltage across the Memristor during non-PE operation?
- Does the resulting reference current vary with changes in *VDD*?



Variation of current with $R_{memristor}$ in 50 nm CMOS BMR



What is the voltage across $R_{memristor}$?



$R_{memristor}$ = 10k, 50k, 100k, 500k, and 1MEG

50 mV across $R_{memristor}$ is good but can we reduce this stress further?

- Looking at the equation for the reference current (below) notice that if *K* goes to 1 the current goes to zero and thus so does the voltage across $R_{memristor}$
- The result is reducing *K* reduces the stress across the device!
- Dropping *K* from 4 to 2 causes the voltage across $R_{memristor}$ to drop from 50 to 25 mV

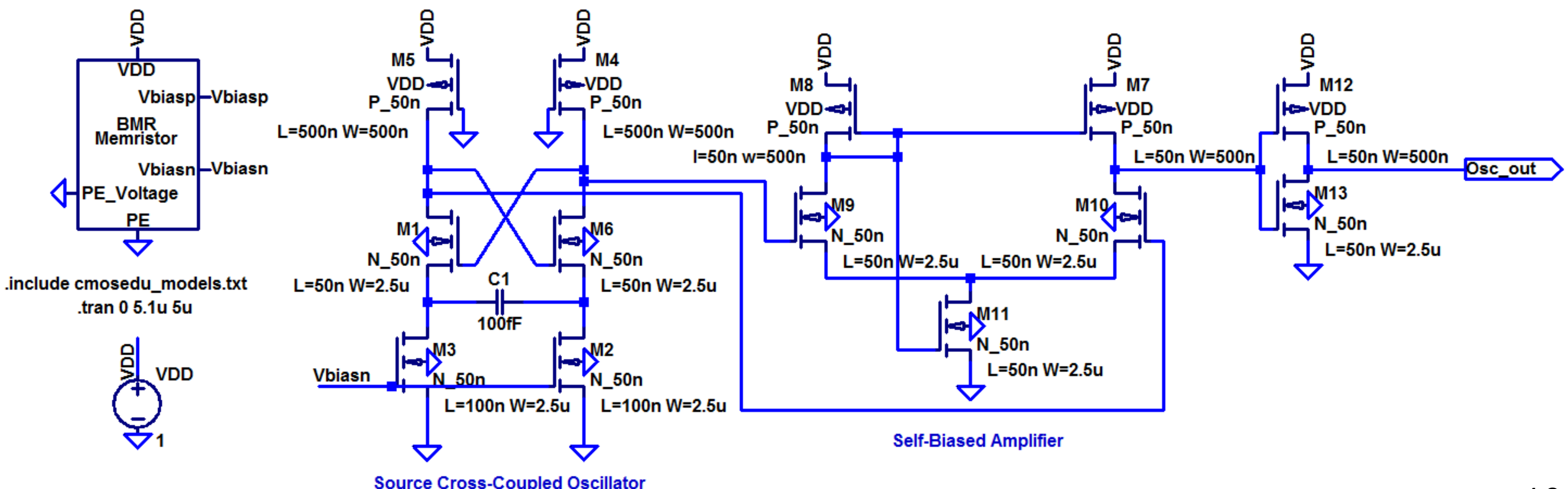
$$I_{REF} = \frac{2}{R_{memristor}^2 KP_n \cdot \frac{W_1}{L_1}} \left(1 - \frac{1}{\sqrt{K}}\right)^2$$

Why is this approach to reconfigurable analog integrated circuits significant and how is it used?

- Programmability is non-volatile
- The circuit is small
- Currents can be used for power supply independent voltage generation
  - Dynamically scale data converter operating range
- Control oscillator frequency
  - Useful in PLLs, charge pumps, wake-up circuits, etc.

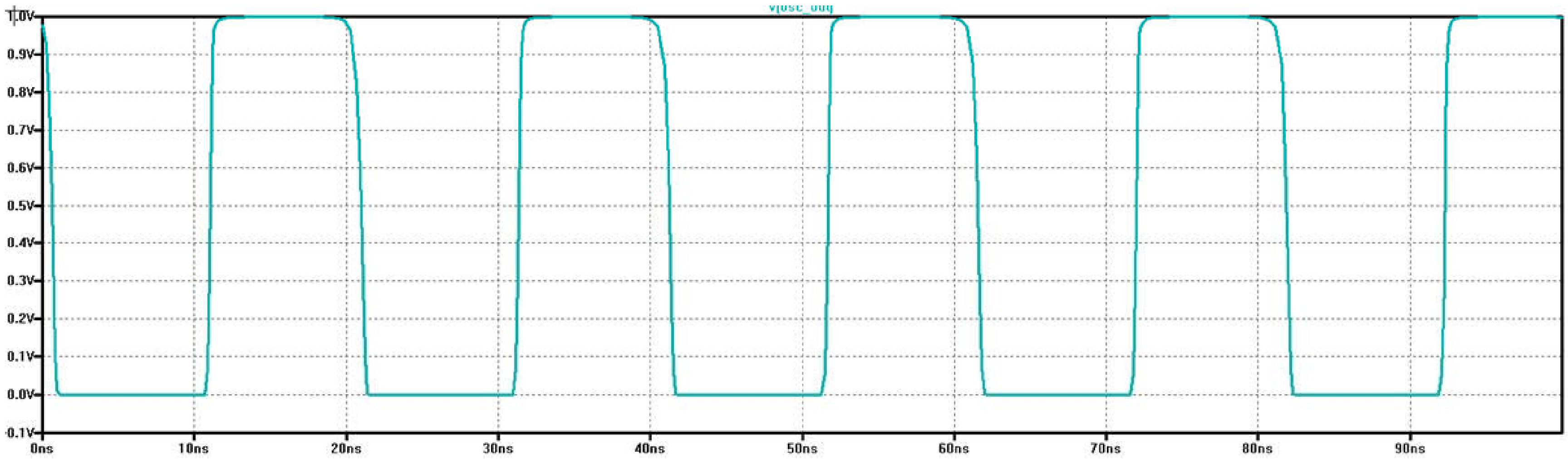
A Memristor-Controlled Oscillator using a Source-Coupled Topology

- Use the Memristor-programmed BMR to set, or control, the frequency of an oscillator
- Potential for very low power operation at high-frequencies



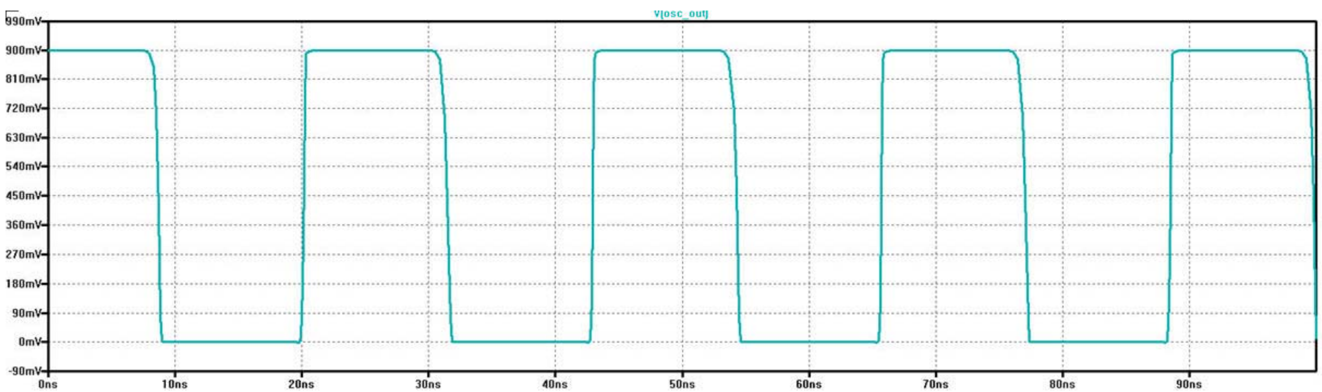
Simulation Results

- Oscillation frequency is near *I*/*C*
  - Here *I* = 5 μA and *C* = 100 fF so the oscillation frequency is close to 50 MHz
- Note that this oscillator is non-volatile, that is, on power-up the oscillation frequency remains on changed from power-down.

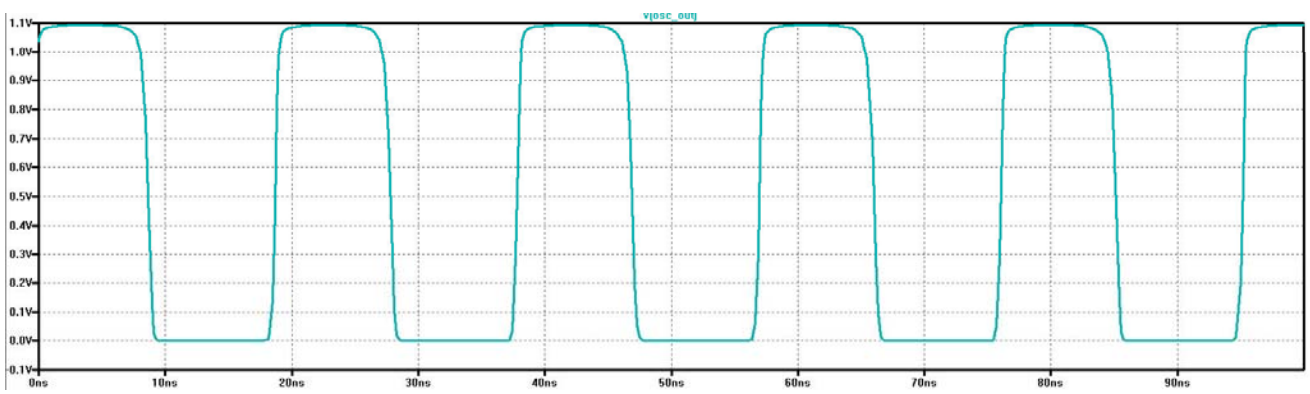


More Simulation Results

- Reducing, to 900 mV and increasing, to 1.1 V, shows that the oscillation frequency doesn't change much (not an exponential relationship as in many integrated oscillators).



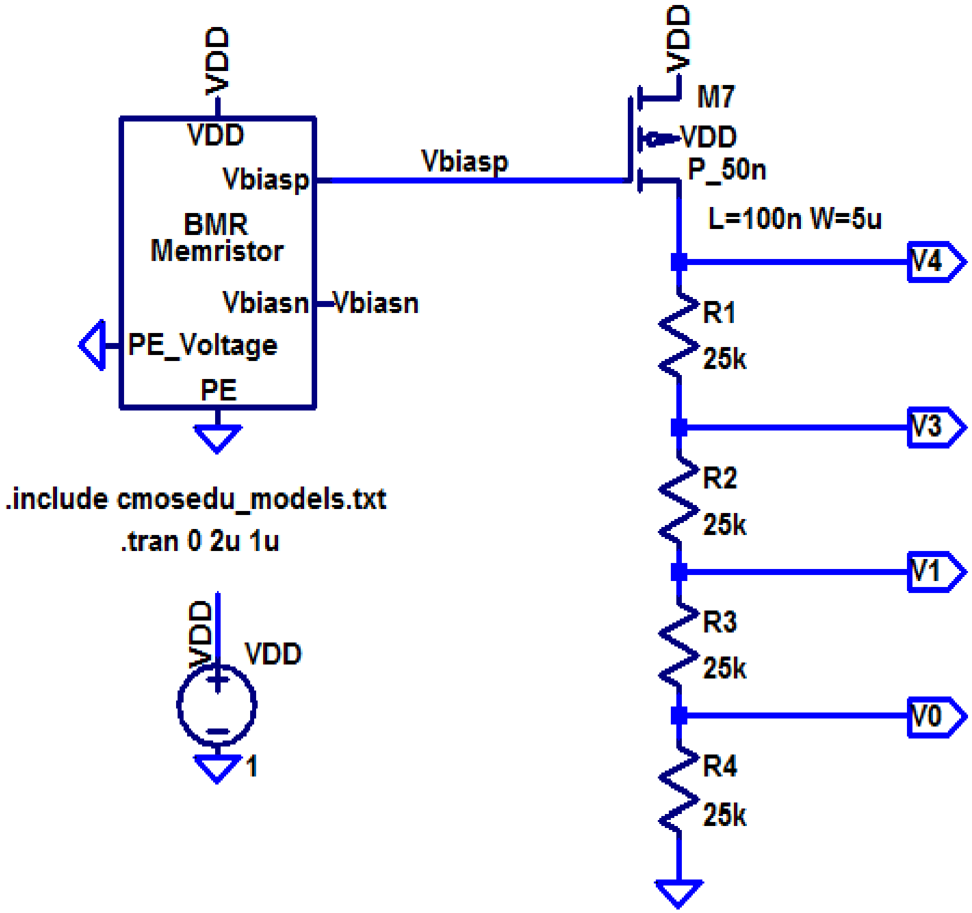
VDD = 900 mV



VDD = 1.1 V

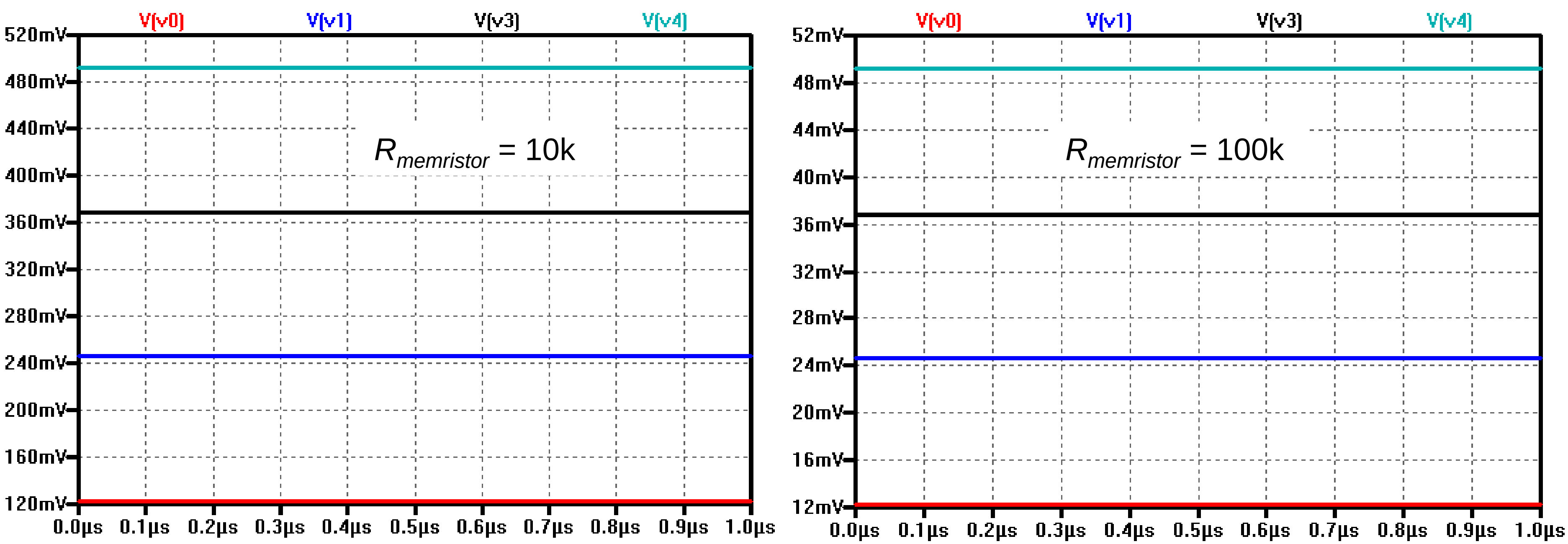
Something simpler: A Voltage Divider

- Use the Memristor-controlled current to generate Memristor -controlled voltages
- Voltages tolerant to changes in the power supply voltage, *VDD*
- Of course they are also non-volatile meaning power can be removed and then re-applied without losing the programmed voltage values



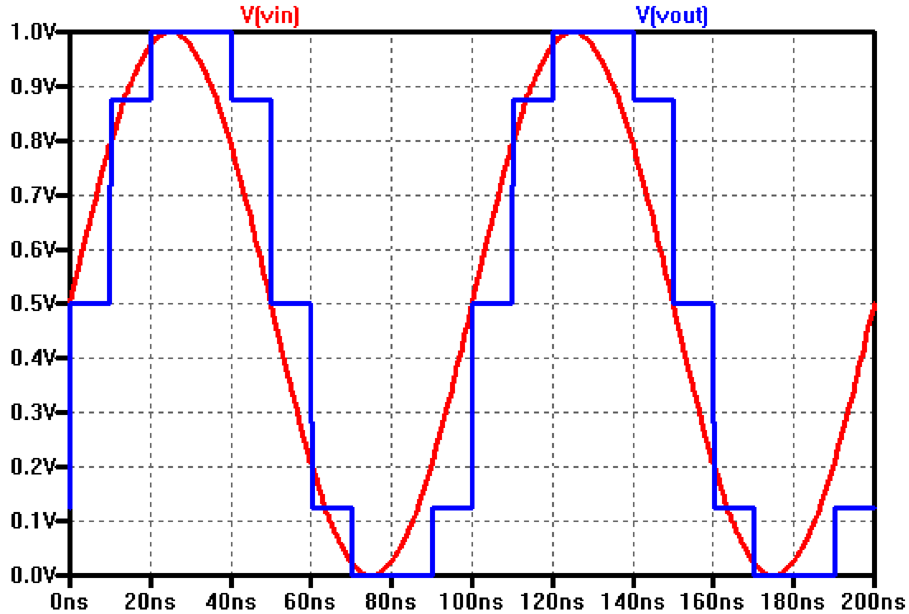
Simulation Results

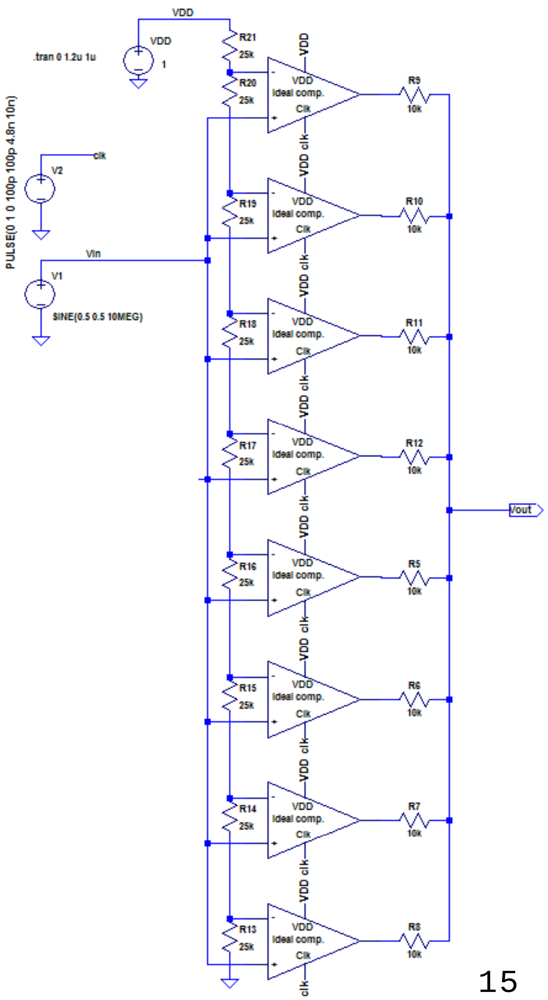
- Below shows how various voltages can be generated by programming the Memristor
- Note! These voltages are independent of *VDD*!
- Again, the programmed voltages are non-volatile



How Can Use this Simple Voltage Divider in a Complex Circuit?

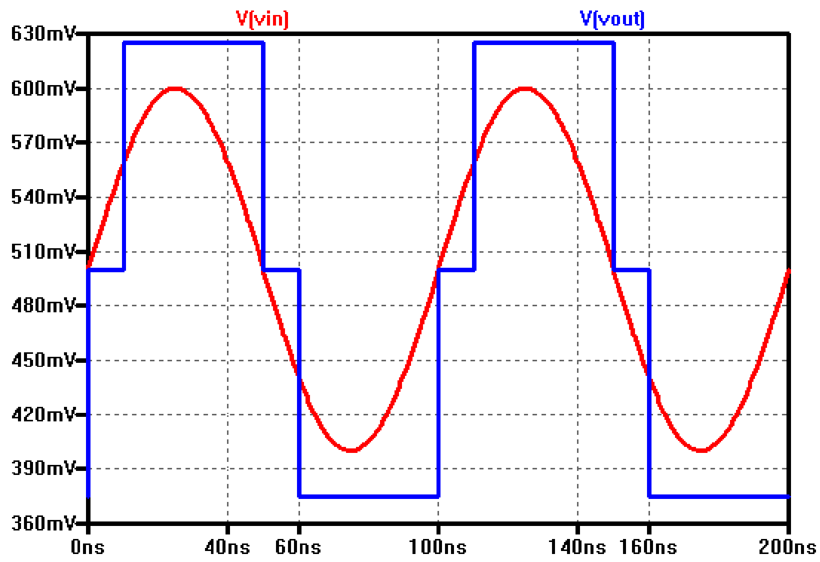
- Consider the Flash ADC seen at the left with simulation results shown below





Reconfiguring the ADC's input range

- What happens when the input signal amplitude shrinks?
- We get fewer output codes thus the noise added to the input signal increases



Quantization Noise

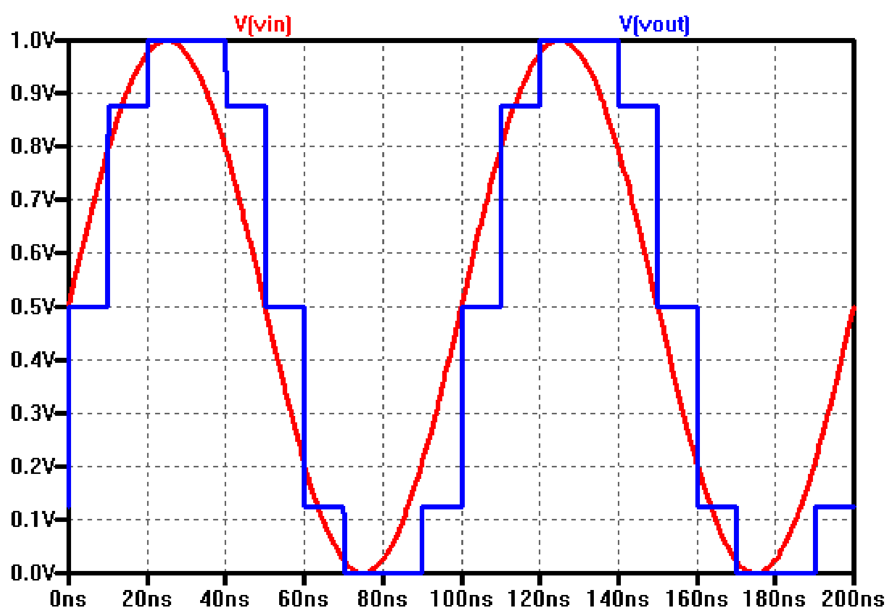
- The voltage dropped across each resistor in our simple 3-bit ADC is $\frac{VDD}{8} = 125\,mV = V_{LSB}$
- This voltage, V_{LSB}, is also the resolution of the ADC
- The RMS value of the quantization noise is give by

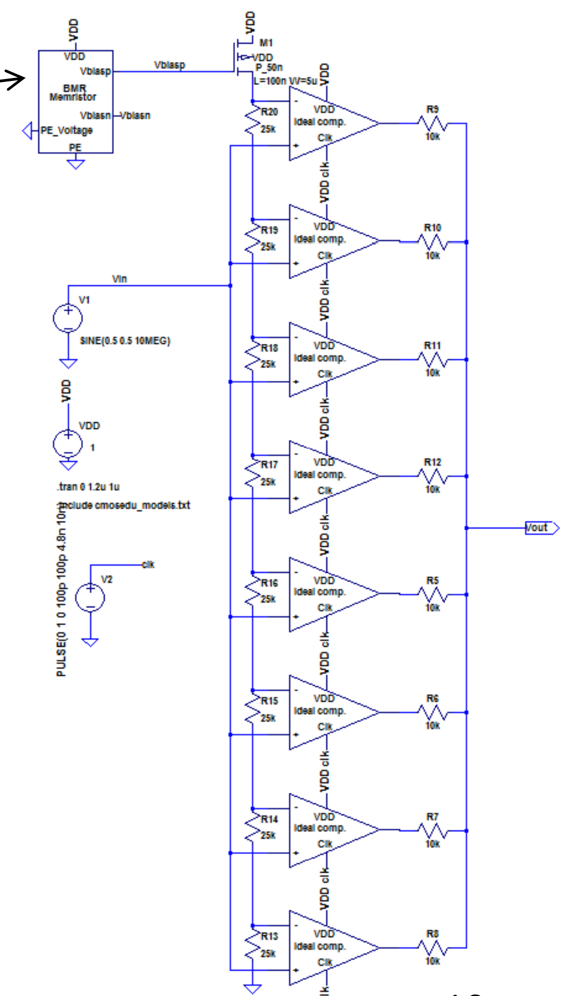
$$V_{Qe,RMS} = \frac{V_{LSB}}{\sqrt{12}}$$

- The key point is that if we can reduce V_{LSB} we can reduce the quantization noise added to the input signal
- Why not simply reduce the resistors or supply voltage driving the resistors to reduce the quantization noise?
  - Answer: then our input signal range is reduced!
  - We want to be able to reconfigure the design for low noise and wide input signal range

Using the Memristor to reconfigure the input signal range - 1

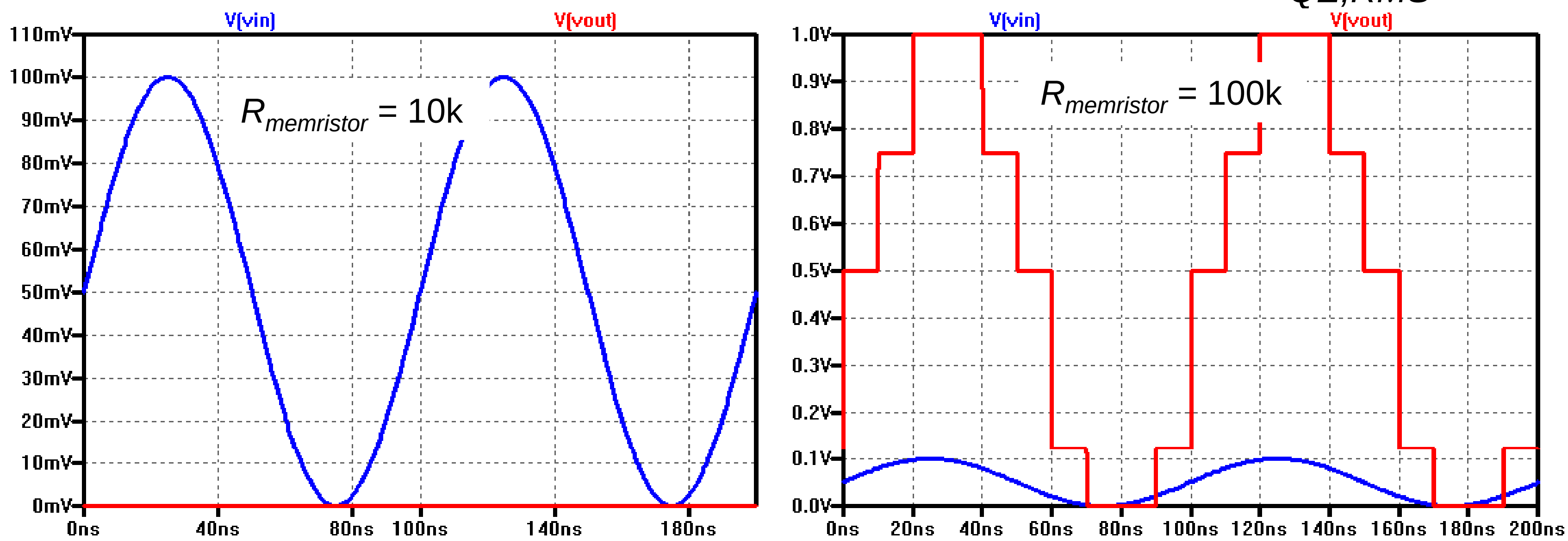
- Adding the Memristor-programmed BMR, $R_{memristor}$ = 10k
- Output seen below for large input signal swings (same as before)





Using the Memristor to reconfigure the input signal range - 2

- The left trace, again using $R_{memristor}$ = 10k, shows how a reduction in the input signal results in no change in the ADC's outputs!
- Reconfiguring the input range allows the ADC's output to swing through all of its codes reducing the added $V_{QE,RMS}$



Summary

- By incorporating the Memristor in the Beta-Multiplier Reference (BMR) we showed that we can
  - Minimize the stress (voltage) across the device
  - Use the Memristor to generate a non-volatile current that is independent of *VDD*
- The Memristor-controlled BMR can then be used to
  - Implement re-configurable voltage references, ADCs, or any circuit that uses reference currents or voltages to control a characteristic of operation
  - Trimming currents or voltages for precision analog design, especially useful in nanometer CMOS where matching is poor (e.g., in a current steering DAC, removing the offset in an op-amp, etc.)