Non-Silicon Non-Binary Computing: Why Not?

Elena Dubrova, Yusuf Jamal, Jimson Mathew
Royal Institute of Technology
Stockholm, Sweden

Overview

- Motivation
- Introduction to multiple-valued computing
- Implementation of multiple-valued functions using chemically assembled electronic nanotechnology
- Conclusion and open problems

Motivation

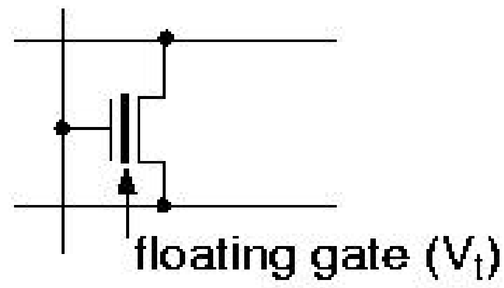
- Current silicon-based technologies use binary bits to represent data
  - transistors have two stable states: on/off
  - they are cheap, reliable and efficient
- It is possible to use current technologies for non-binary computing, but the theoretical advantage is lost, except for some applications
- The situation might be different for non-silicon based technologies

Theoretical advantages

- On- and off-line interconnect can be reduced if signals in the circuit assume four or more levels rather than two
- Storing two instead of one bit of information per cell doubles the density of the memory in the same die size
  - Intel StartaFlash, NEC 4-Gbit DRAM
- Arithmetic circuits often benefit from using other than binary number systems
  - ripple-through carries can be reduced or eliminated if redundant or residue number systems are used

Example: 4-valued flash memory

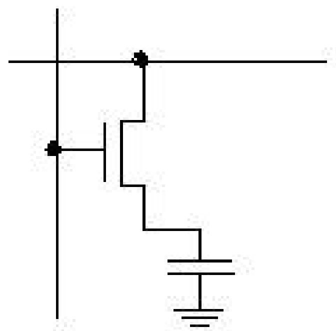
- Non-volatile multiple-write memory
- Found in over 90% PCs, over 90% cellular phones and over 50% modems
- Each cell consists of a single transistor



- Transistors can have one of four different threshold voltages V_t , controlled by the amount of charge stored on the floating gate

Dymanic RAM (DRAM)

- Volatile general purpose memory
- Used in main processing units, operating systems, video and audio data processing
- Each cell consists of a single capacitor and a transistor



- capacitor stores a quantity of charge that corresponds to the logic value of the signal

Higher levels of abstraction

- Using multiple-valued logic at higher levels of abstractions gives us a more compact and natural description of the problem
  - for example, a traffic light controller can be described using 3 values, representing “green”, “yellow” and “red”

Practical problems

- The attempts to build multiple-valued circuits can be traced back to 1960
  - 3-valued SETUN computer
  - bipolar I^2L, ECL
  - CMOS
- Except Flash and DRAM memory applications, no mulitple-valued design survived the competition with binary designs
- Silicon-based technologies do not seem to be suitable for implementinig mulitple-valued circuits

Goal of this paper

- To show that it is possible to implement multiple-valued circuits with non-silicon based technologies
  - chemically assembled electronic nanotechnology is taken as an example
  - the idea can be used in other technologies
- To show that we can benefit from using mvultiple-valued circuits instead of binary

Multiple-valued computing

- Instead of Boolean functions $\{0,1\}^n \rightarrow \{0,1\}$ we implement multiple-valued functions $\{0,1,..,m-1\}^n \rightarrow \{0,1,..,m-1\}$

x_2 \ x_1	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

an example of 3-valued function $MIN(x_1,x_2)$

Functionally complete sets

- A set of functions is called functionally complete if any other function can be composed from the functions in this set
  - {AND, OR, NOT} is functionally complete for Boolean functions $\{0,1\}^n \rightarrow \{0,1\}$
- Boolean algebras are not functionally complete for functions over the sets other than {0,1}

Chain-based Post algebra

- A generalization of Boolean algebra
  - corresponds to the first multiple-valued logic developed in 1921 by Emil Post
- $P := < M;+,\cdot,L>$
  - $M := \{0, 1, ..., m - 1\}$ set whose elements form totally ordered chain $0<1<...<m-1$
  - "+" is the binary operation maximum
  - "·" is the binary operation minimum
  - $L := \{ x^0, x^1,..., x^{m-1} \}$ is the set of unary literal operators

Representation of multiple-valued functions

$x_2 \backslash x_1$	0	1	2
0	0	1	0
1	0	0	0
2	2	0	2

$$f(x_1,x_2) = 1\cdot x_1^1 \cdot x_2^0 + 2\cdot x_1^0 \cdot x_2^2 + 2\cdot x_1^2 \cdot x_2^2$$

$x_2 \backslash x_1$	0	1	2
0	0	1	0
1	0	1	0
2	2	2	2

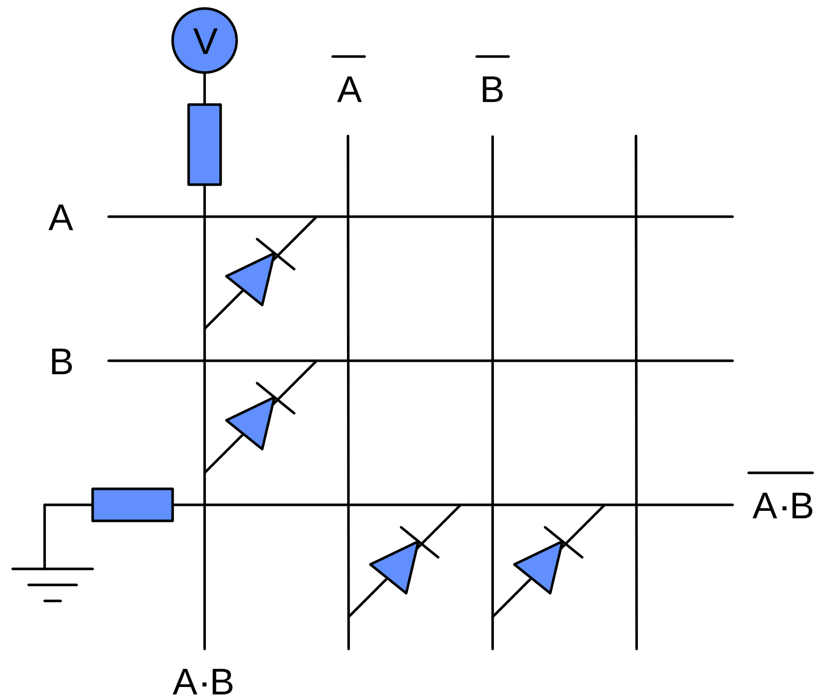
$$f(x_1,x_2) = 1\cdot x_1^1 + 2\cdot x_2^2$$

1 + 2 = 2 since “+” = MAX

Chemically assembled electronic nanotechnology (CAEN)

- Dense regular architecture: *nanoFabric* composed of *nanoBlocks*
- *nanoBlock* is a molecular logic array that can be programmed to implement a three-input three-output Boolean function and its complement
- Active elements are molecular switches
  - two-terminal devices (cheaper chemical assembley)
  - no inverters can be built
  - all signals should be available is both, complemented and non-complemented form

nanoBlock of an AND



{AND,NOT} is a functionally complete set for Boolean functions, so any Boolean function can be composed from nanoBlocks

Implementation of multiple-valued functions using CAEN

- We would like to preserve the following features of nanoFabric:
  - the architecture is a two-dimensional array
  - it can be configured to implement any multiple-valued function
  - only two-terminal devices are used
- The last point implies that, as in binary case, no ”inverters” are available

What is an ”inverter” in MV-case?

- In binary case, the complement is defined by

$$x' = 1 - x$$

- In m-valued case, it can be defined as

$$x' = (m-1) - x$$

x	x'
0	2
1	1
2	0

- However, it will not result a functionally complete set in combination with AND and OR

Multiple-valued ”inverter”

- To get a functionally complete set, we have to extend NOT to the literal operator, defined by

$$x^i = \begin{cases} 1, & if\ x = i \\ 0, & otherwise \end{cases}$$

x	x^0	x^1	x^2
0	1	0	0
1	0	1	0
2	0	0	1

- in combination with AND and OR, literals give us a functionally complete set for $\{0,1,\ldots,m\text{-}1\}^n \to \{0,1\}$ functions

Implmentation of MV functions

- In binary case, the absence of inverters implies that all the signals should be availible in complemented and non-complemented form
- In multiple-valued case, all the signals should be availible as literals
  - literals are functions of type $\{0,1,..,m-1\}^n \rightarrow \{0,1\}$
  - AND and OR operations can be applied to literals
    - $x^i \cdot x^j$ is of type $\{0,1\}^2 \rightarrow \{0,1\}$
  - same diode-resistor logic as in binary case can be used for multiple-valued functions

Example of literals f^0, f^1, f^2

f:

y \ x	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

f^0:

y \ x	0	1	2
0	1	1	1
1	1	0	0
2	1	0	0

f^1:

y \ x	0	1	2
0	0	0	0
1	0	1	1
2	0	1	0

f^2:

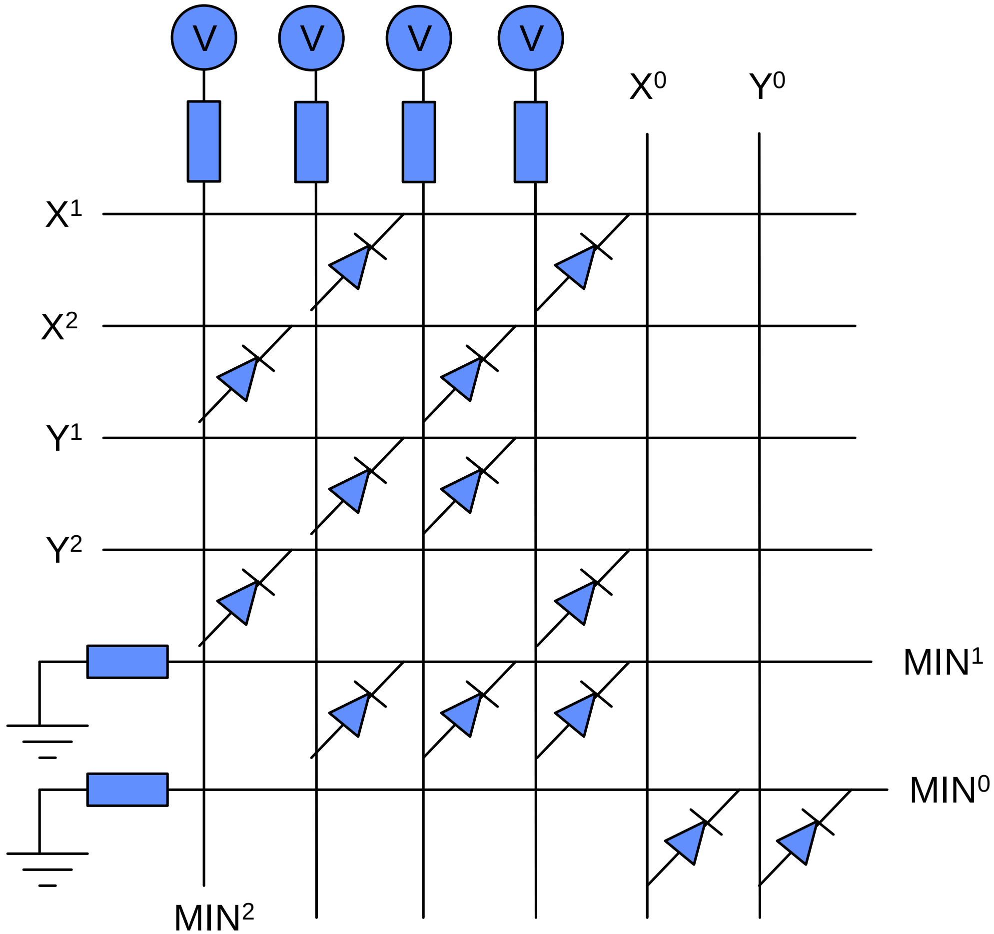
y \ x	0	1	2
0	0	0	0
1	0	0	0
2	0	0	1

$f^0 = x^0 + y^0$

$f^1 = x^1y^1 + x^1y^2 + x^2y^1$

$f^2 = x^2y^2$

3-valued MIN(x,y) gate



4-valued adder

- In the paper, we show a design of a 4-valued adder implemented in the molecular logic array using 16 x 16 grid
- It is smaller than a 2-bit binary adder implemented in the molecular logic array

Conclusion

- It is possible to implement multiple-valued functions with chemically assembled electronic nanotechnology using the same diode-resistor logic as in binary case
- Multiple-valued designs can be more compacts than binary ones implementing the same functionality
- The idea can be extended to other non-silicon technologies

Open problems

- In general, mapping a multiple-valued function in a molecular array is a harder problem
  – a complete suite of associated algorithms is still not available
- Some multiple-valued synthesis and optimization tools exist, but they are not as mature as binary tools
  – MVSIS, University of California at Berkeley