ROYAL INSTITUTE OF TECHNOLOGY

Multiple-Valued Logic in VLSI: Challenges and Opportunities

Elena Dubrova
Electronic System Design Lab
Department of Electronics
Royal Institute of Technology
Stockholm, Sweden

Outline

- Introduction to MVL theory
- Applications of MVL
  - applications to binary problems
  - design of MVL circuits
    - arithmetic circuits
    - memories
- Conclusions & future prospectives

Definition of chain-based Post algebra

- $P := < M;\ +, \bullet, J; 0, m\text{-}1>$
  - $M := \{0, 1, ..., m - 1\}$ - set whose elements form a totally ordered chain $0 < 1 < ... < m\text{-}1$
  - $+$ - binary operation **maximum**
  - $\bullet$ - binary operation **minimum**
  - $J := \{ \overset{0}{x}, \overset{1}{x}, ..., \overset{m-1}{x} \}$ - set of **literal** operators defined by

$$\overset{i}{x} = \begin{cases} m-1, x = i \\ 0, otherwise \end{cases}$$

MIN, MAX, literals for m=3

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

$MIN(x_1,x_2)$

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

$MAX(x_1,x_2)$

x	${}^{0}x$	${}^{1}x$	${}^{2}x$
0	2	0	0
1	0	2	0
2	0	0	2

literals

Multiple-valued functions

- An *n*-variable *m*-valued function $f(x_1, x_2, ..., x_n)$ is a mapping $f: \{0,1, ..., m-1\}^n \rightarrow \{0,1, ..., m-1\}$

x_1	x_2	f
0	0	0
0	1	0
0	2	2
1	0	1
1	1	2
1	2	0
2	0	0
2	1	1
2	2	0

2-variable
3-valued
function

Functionally complete sets

- A set of functions over M is called ***functionally complete*** if any other function over M can be composed from the functions in this set

- {AND, OR, NOT} is functionally complete for Boolean functions $f: \{0,1\}^n \rightarrow \{0,1\}$

- {MIN, MAX, literals} is functionally complete (with constants) for *m*-valued functions $f: \{0,1, ..., m\text{-}1\}^n \rightarrow \{0,1, ..., m\text{-}1\}$

Many other functionaly complete sets over M exist

- {m-valued Sheffer-stroke} (Webb 1935)
- {sum mod m, mult mod m}, m-prime (Cohn 1960), m-power of prime (Pradhan 1974)
- {sum mod m, MIN} (Dubrova 1997)

Representation of multiple-valued functions

- Generalized Karnaugh map
- Sum-of-products expression over P
- Multiple-Valued Decision Diagram (MDD)

Karnaugh maps & expressions

$x_2 \backslash x_1$	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

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

$1 + 2 = 2$ "+" = MAX

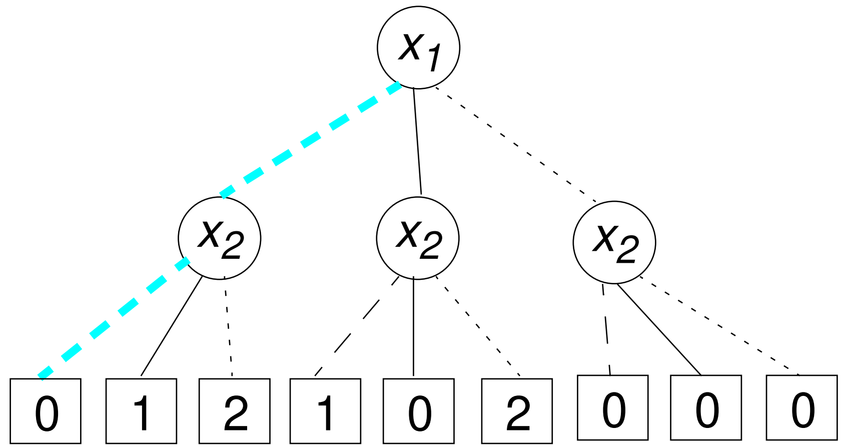
Multiple-Valued Decision Diagrams

truth table

x_1	x_2	f
0	0	0
0	1	1
0	2	2
1	0	1
1	1	0
1	2	2
2	0	0
2	1	0
2	2	0

$\rightarrow$

decision diagram



- + ordering rules + reduction rules to get Reduced Ordered MDD

Applications of MVL

- **First group**: uses multiple-valued logic domain to solve binary problems more efficiently

- **Second group**: targets the design of electronic circuits which employ more than two discrete levels of signals

Applications to binary problems

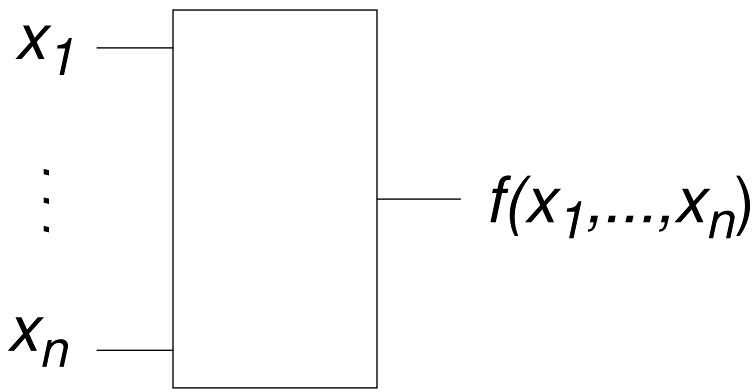
- multiple-output functions - treat the output part as a single multiple-valued variable. Allows better utilization of common subfunctions

- Programmable Logic Arrays (PLAs) with decoders - pair two inputs and treat them as a single 4-valued input. Allows to reduce the area of PLA

- in testing - use the third value as a means for signaling faulty operation

What is an MVL circuit?

- Boolean (or 2-valued) logic circuit:
  - has *n* inputs taking values from {0,1}
  - has 1 (or more) output(s) taking values from {0,1}
  - is built out of gates realizing 2-valued logic operations, like AND, OR, NOT
- *m*-valued logic circuit:
  - has *n* inputs taking values from {0,1, ... , m-1}
  - has outputs taking values from {0,1, ... m-1}
  - is build out of gates realizing *m*-valued logic operations like MIN, MAX, literals



- Boolean logic circuit implements the function:

  $f : \{0,1\}^n \text{ -> } \{0,1\}$

- *m*-valued logic circuit implements the function:

  $f : \{0,1, ..., m-1\}^n \text{ -> } \{0,1, ..., m-1\}$

Theoretical advantages of MVL circuits

- In a typical VLSI chip, about 70% of the chip area is devoted to interconnection, 20% to insulation and 10% to devices
- In a multiple-valued circuit:
  - wires carry more information - saving in the number of wires and in insulation between wires
  - pins carry more information - saving in the number of pins

- Alternative to binary number systems allow fast arithmetic operations
- MVL storage allows to store more bits of information per memory cell

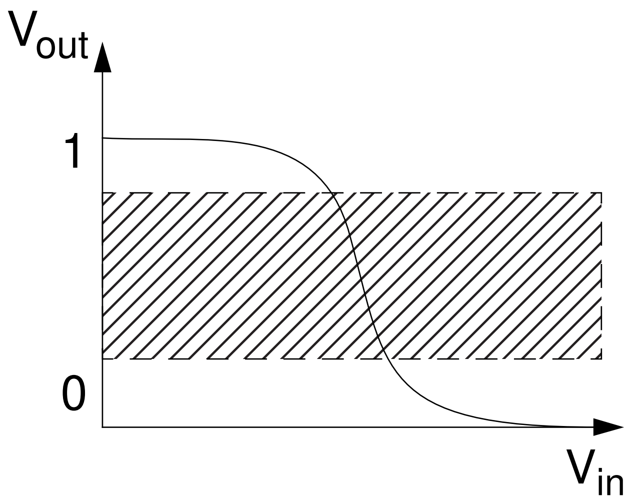
Why aren't MVL circuits widely used?

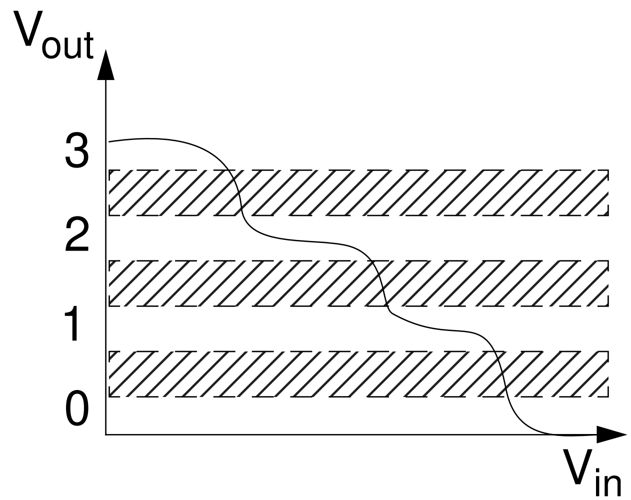
- History: before 1947:
    - three-position polarized relays
    - 3-valued SETUN computer (1960)
  - after: transistors are cheap, reliable and efficient
    - MVL circuit can be built with binary transisitors, but the theoratical advantage is lost, except for some applications (arithmetic logic, memories)

- Cheap, reliable and efficient device with *m* stable states is not discovered yet

Main practical problem:

- noise immunity





Recent achievements in MVL circuit design:

- **Arithmetic circuits**: multipliers, adders - prototype chips are fabricated
- **Memories**: Flash, DRAM - great commercial success

Number representation

- A string of digits $(a_{n-1}, a_{n-2}, ..., a_1, a_0)$ represents a number

$$a_{n-1} \cdot r^{n-1} + a_{n-2} \cdot r^{n-2} + ... + a_1 \cdot r^1 + a_0$$

where r is a **radix** of the number system

- Examples:

$$6 = 1{\cdot}2^2 + 1{\cdot}2^1 + 0{\cdot}2^0 = (110)_, \quad r = 2, \;\; a_i \in \{0,1\}$$

$$6 = 1{\cdot}3^2 - 1{\cdot}3^1 + 0{\cdot}3^0 = (1\text{-}10)_, \quad r = 3, \;\; a_i \in \{-1, 0,1\}$$

- There are two alternative ways to represent digits in an m-valued system:
  - unsigned (unbalanced):

$$a_i \in \{0, 1, ..., m-1\}$$

  - signed (balanced), $m = 2k+1$:

$$a_i \in \{-k, -k+1, ..., -1, 0, 1, ..., k-1, k\}$$

- Advantages of signed representation:
  - the sign of a number is the sign of the leading non-zero digit
  - the negation of a number is obtained by interchanging k and -k, for all $k \neq 0$

Redundant number systems

- A string of n digits (a_{n-1}, ..., a_0) can code r^n different numbers in a radix r number system
- If each of a_i can take m values, and m > r, then two different strings might represent the same number
- For example, if r = 2, and $a_i \in \{0,1,2,3\}$:

$$(11111) = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 31$$

$$(10231) = 1 \cdot 2^4 + 0 \cdot 2^3 + 2 \cdot 2^2 + 3 \cdot 2^1 + 1 \cdot 2^0 = 31$$

- redundancy allow fast arithmetic operations

Example

- $r = 2$, $a_i \in \{-1,0,1\}$ signed-digit redundant number system, developed by Avizenis in 1961
- Restricts carry propagation in adders to only adjacent cells
- Allows addition of two *n* bit numbers in constant time (worst-case time *n* for ripple-carry adders, *log n* for carry look-ahead adders)

- the addition of two numbers A = (a_{n-1}, ..., a_0) and B = (b_{n-1}, ..., b_0), is performed by three successive steps:

  1. Compute the linear sum $z_i = a_i + b_i$

  2. Determine the intermediate sum w_i and the carry c_i, satisfying the equation $2c_i + w_i = z_i$

  3. Compute the final sum $s_i = w_i + c_{i-1}$

- Example of addition in $r = 2$, $a_i \in \{-1,0,1\}$ number system: Let $A = 2 = (010)$ and $B = 3 = (10\text{-}1)$

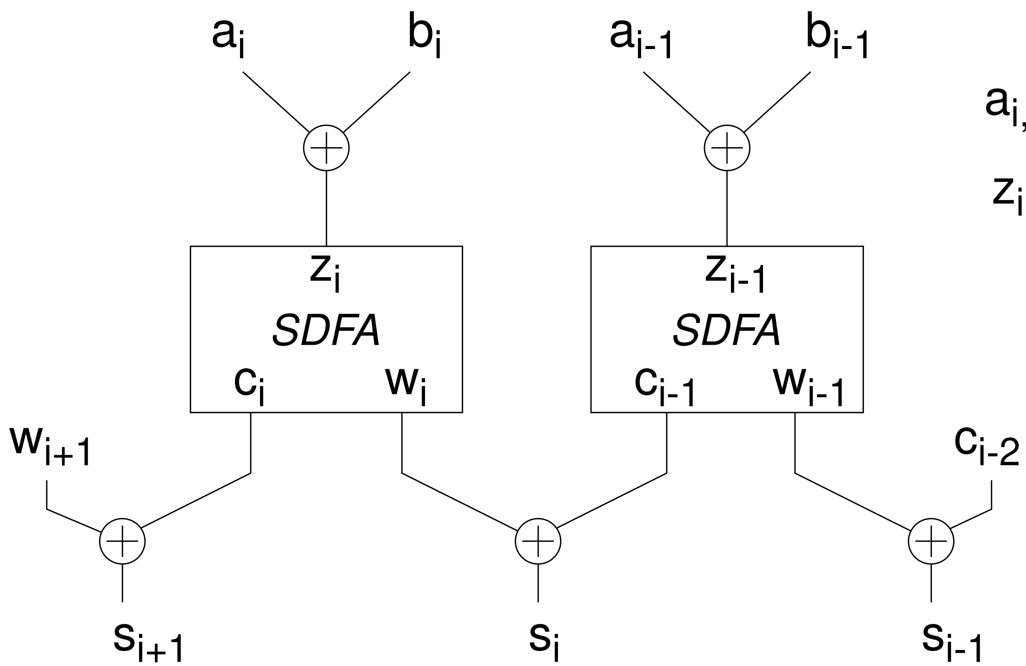
A:		0	1	0
B:		1	0	-1
z_i:		1	1	**-1**
w_i:		1	1	**1**
c_{i-1}:	0	0	**-1**	0
s_i:	0	1	0	1

$$z_i = a_i + b_i$$

$$2c_i + w_i = z_i$$

$$s_i = w_i + c_{i-1}$$

$$(101) = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$$



$a_i, b_i, s_i, c_i \in \{-1,0,1\}$

$z_i, w_i \in \{-2,-1,0,1,2\}$

- SDFA = signed-digit full adder
- carry-propagation chain is limited to one digit on the left
- inherently multiple-valued operations - use of multiple-valued signals allows more compact design

MVL vs. binary implementation of signed-digit full adder

- PSPICE simulation based on 0.8-micron CMOS:

SDFA	Binary CMOS	Multiple-valued CMOS
Transistors count	56	34
Delay	3.9 ns	3.0 ns
Power dissipation	0.6 mW	0.6 mW
Supply voltage	1.5 V	1.5 V

MVL memories

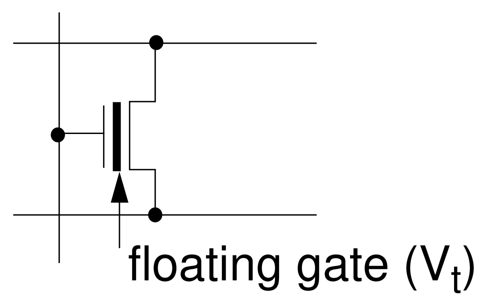
- 4-valued Flash memories
- 4-valued DRAM memories

Flash memory

- Flash - non-volatile multiple-write memory
- Found in over 90% of PCs, over 90% of cellular phones and over 50% of modems
- Key component of the emerging digital imaging and audio markets where it serves as the digital "film" or digital "tape"

4-valued Flash memory

- As in conventional Flash, each memory cell consist of a single transistor with a floating gate



- Instead of 2 threshold voltages V_t, floating gates can be charged to 4 different threshold voltages V_t

- Density is doubled without increasing the die size

Technical parameters

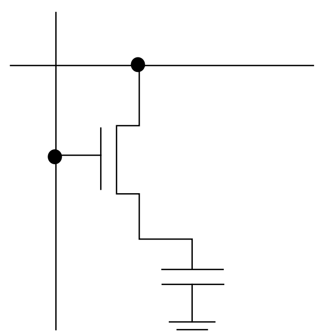
	Intel	NEC	Samsung Electron	Hitachi and Mitsubishi
Memory size	32-Mbit	64-Mbit	128-Mbit	256-Mbit
Process	0.6-µm	0.4-µm	0.4-µm	0.26-µm
Die size	152 mm^2	98 mm^2	117 mm^2	139 mm^2
Power supply	5 V	3.3 V	3.3 V	3 V
Access time	120 ns	80 ns	25 ns	50 ns

(+) doubled density (2 bits instead of 1 per cell)

(-) slower read speed (100ns in Intel's standard, 120ns in 4-valued)

Dynamic RAM (DRAM)

- DRAM - volatile general purpose memory
- applications: main processing units, computer 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 logical value of the signal



Traditional techniques for DRAM density increase

- Process scaling
  - 70% reduction in the minimum design rule for each generation
- Process scaling + better cell structure
  - 50% larger chips size each generation
- 1995 - first 1-Gbit DRAM (standard)
- 1997 - first 4-Gbit DRAM (4-valued)

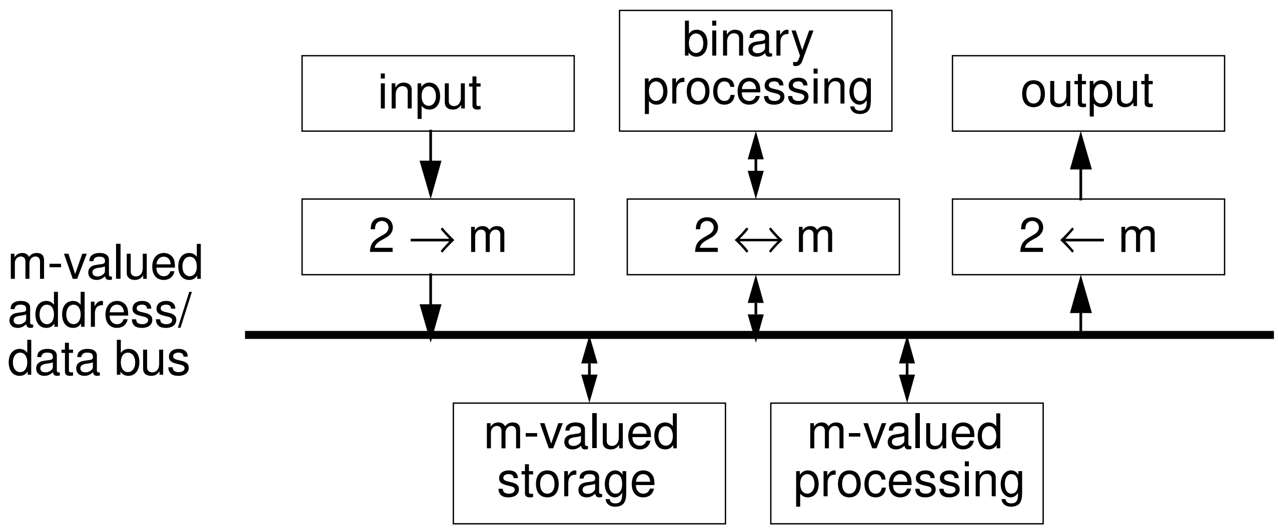
NEC's 4-Gbit 4-valued DRAM

Process	0.15-μm CMOS
Die size	986 mm^2
Power supply	2.2 V
Data transfer rate	1Gbit/sec at 125 MHz

- large storing capacity (doubled compared to standard)
  - capable of storing 47 minutes of full-motion video
- high-speed access to data (same as standard)
  - Jurassic Park in real time

Conclusion

- MVL circuits might be one possible solution to pin limitation and interconnection problems
- The technology is reaching maturity allowing to build MVL circuits for specific applications



Open problem

- Tools for representation and manipulation of multiple-valued functions are not as well developed as ones for Boolean functions