KUNGL TEKNISKA HÖGSKOLAN

Inte

Master Pr

in System-on-Chip Design

L10: Multiple-valued logic

Reading material

- Muzio & Wesselkamper "Multiple-valued switching theory", p. 38 - 66
- Hudsson & Sasao, chapter 4

Completeness in MVL case

- Boolean algebras are not functionally complete for functions over the sets other than B={0,1}
- We will generalize Boolean algebras to handle functions over M = {0,1,…,m-1}
- General conditions for completeness in MVL case were formulated by Ivo Rosenberg in 1965

Generalizing Shannon decomposition

- First, we extend Shannon decomposition theorem:

$$f(x_1,x_2, \ldots,x_n) = x'_1 \cdot f|_{x_1=0} + x_1 \cdot f|_{x_1=1}$$

by generalizing the operations ', + and · to multiple-valued case

Generalizing complement

- We replace functions x and x' by m literal functions, defined by

$$J_i x = \begin{cases} 1, \text{ if } x = i \\ 0, \text{ otherwise} \end{cases}$$

So, for m=2 we get:

i.e. $J_0 x = x'$, $J_1 x = x$

x	$J_0 x$	$J_1 x$
0	1	0
1	0	1

Sum and product generalization

- We generalize the sum and product operations by specifying the properties we want them to have:
  - A binary operation "·" over M is a product-type operation iff $\forall a \in M \; a \cdot \mathbf{0} = \mathbf{0} \cdot a = \mathbf{0}$ and $a \cdot \mathbf{1} = \mathbf{1} \cdot a = a$
  - A binary operation "+" over M is a sum-type operation iff $\forall a \in M \; a + \mathbf{0} = \mathbf{0} + a = a$

Example for m = 4

x+y	0	1	2	3
0	0	1	2	3
1	1	-	-	-
2	2	-	-	-
3	3	-	-	-

x·y	0	1	2	3
0	0	0	0	0
1	0	-	-	1
2	0	-	-	2
3	0	1	2	3

$1 = 3, 0 = 0$

Generalized Shannon decomposition theorem

- Every function $f: M^n \rightarrow M$ can be written in the form:

$$f(x_1,x_2,\ldots, x_n) = \sum_{i=0}^{m-1} J_i x_1 \cdot f_i(x_2,\ldots \cdot x_n)$$

where $f|_{x_1=i} := f(i,x_2, \ldots,x_n)$, are subfunctions of f

Completeness theorem

- The set of operations $\{+, \cdot, J_i x\}$, $\forall i \in M$, defined as above is functionally complete for functions $f: M^n \rightarrow M$

Canonical form

- Every function f: $M^n \rightarrow M$ has a canonical form of type:

$$f(x_1,x_2,\ldots, x_n) = \sum_{i=0}^{m^n-1} c_i \cdot J_{i1}x_1 \cdot J_{i2}x_2 \cdot \ldots \cdot J_{in}x_n$$

where

  - $c_i \in M$ is a constant
  - $(i_1,i_2,\ldots, i_n)$ is the m-ary expansion of i

Alternative decomposition

- Every function f: $M^n \rightarrow M$ can be written in the form:

$$f(x_1,x_2,\ldots, x_n) = \sum_{i=0}^{m-1} i \cdot J_i f\ (x_1,x_2,\ldots, x_n)$$

where $J_i f$ is the i_{th} literal of f

Post algebra

- One example of an algebra, satisfying the above restrictions, is the chain-based Post algebra based on the set of operations corresponds to the first multiple-valued logic developed in 1921 by Emil Post

Definition of chain-based Post algebra

- P := < M; +, ·, J; **0, 1**>
  - 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
  - J := {J_0x, J_1x,..., $J_{m-1}x$ } is the set of literal operators (defined as on p.4)
  - **0** = 0, **1** = m-1

MIN, MAX, literals for m=3

MAX	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

MIN	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

x	J_0x	J_1x	J_2x
0	2	0	0
1	0	2	0
2	0	0	2

Many other functionaly complete sets over M exist

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

Representation of multiple-valued functions

- Generalized Karnaugh maps
- Sum-of-products expressions over P
- Multiple-Valued Decision Diagrams (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 since “+” = MAX

Minimizing MVL expressions

- Properties of the operations of Post algebra can be used to simplify the expressions
  - for example, we can use the property:

$$x^0 + x^1 + \ldots + x^{m-1} = m-1 \qquad x' + x = 1$$

$$x^0 \cdot Y + x^1 \cdot Y + \ldots + x^{m-1} \cdot Y = Y \qquad x' \cdot Y + x \cdot Y = Y$$

  - another powerful property is: for any $a,b \in M$

$$a + b = a \text{ if } a > b \qquad 1 + 0 = 1$$

Previous example

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

$$2\cdot x_1^{0}\cdot x_2^{2} + 2\cdot x_1^{1}\cdot x_2^{2} + 2\cdot x_1^{2}\cdot x_2^{2} = 2\cdot x_2^{2}$$

$$1\cdot x_1^{1}\cdot x_2^{0} + 1\cdot x_1^{1}\cdot x_2^{1} + 2\cdot x_1^{1}\cdot x_2^{2} = 1\cdot x_1^{1}$$

Extension - set literal

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

$$f(x_1,x_2) = 1x_1^{1}x_2^{0} + 1x_1^{1}x_2^{1} + 2x_1^{0}x_2^{2} + 2x_1^{2}x_2^{2}$$

$$= 1 \cdot x_1^{1} \cdot x_2^{\{0,1\}} + 2 \cdot x_1^{\{0,2\}} \cdot x_1^{2}$$

set literal $$x^{S} = \begin{cases} m-1, & x \in S \\ 0, & otherwise \end{cases}$$

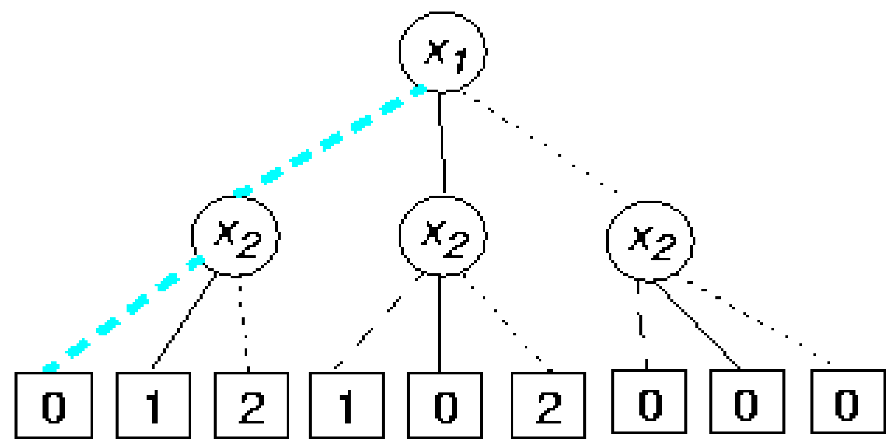
Multiple-Valued Decision Diagrams

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

→

decision diagram



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

Minimization of MVL functions

- As in Boolean case, we can use the properties of the operations of Post algebra to simplify the expressions
  - For literals, the following rules hold:

P1: $\forall i,j \in M,\ J_i x \cdot J_j x = 0$

P2: $\sum_{i=0}^{m-1} J_i x = m-1$

P3: $\forall i \in M,\ (J_i x)' = \sum_{j=0, j \neq i}^{m-1} J_j x$

Minimization of MVL functions

- MVL functions can be minimized directly, but the algorithms are very time-consuming
- A more efficient way is to first split the function f: $M^n \rightarrow M$ into m-1 literals f^i, $i \in \{1,2,\ldots,m-1\}$

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

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

f

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

f^0

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

f^1

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

f^2

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

f^0 = off-set

f^1 = on-set

Minimization of MVL functions

- Literals f^i are of type $\{0,1,..,m-1\}^n \rightarrow \{0,m-1\}$
  - {AND, OR, literals} basis can be used
  - Boolean minimization techniques extend directly
    - on-set = minterms mapped to m-1
    - off-set = minterms mapped to 0
- Minimize each of f^i using the rule:
  - on-set of f^a becomes don't care set for $f^b, \forall a > b,\ a,b \in M$

Example

f

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

f^1

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

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

f^2

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

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

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

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

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 products
- PLAs with decoders - pair two inputs and treat them as a single 4-valued input. Allows to reduce the area of PLA
- at higher levels of abstractions - allows a more compact and natural description of the problem

Multiple-output functions

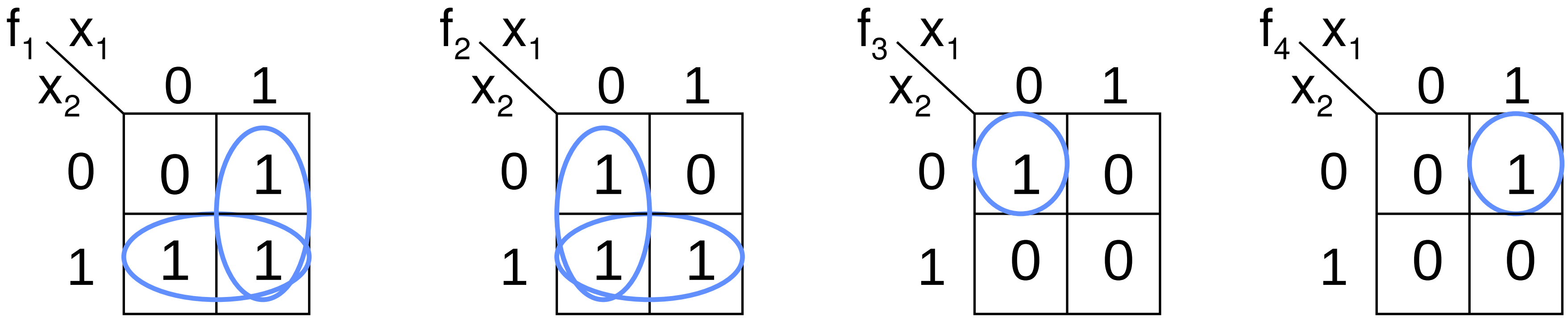
- Convert an n-variable k-output Boolean function f: $B^n \rightarrow (B \cup \{-\})^k$ into an (n+1)-variable 1-output function with one variable being multiple-valued:

$$f: B^n \times \{0,1,\ldots,k-1\} \rightarrow B \cup \{-\}$$

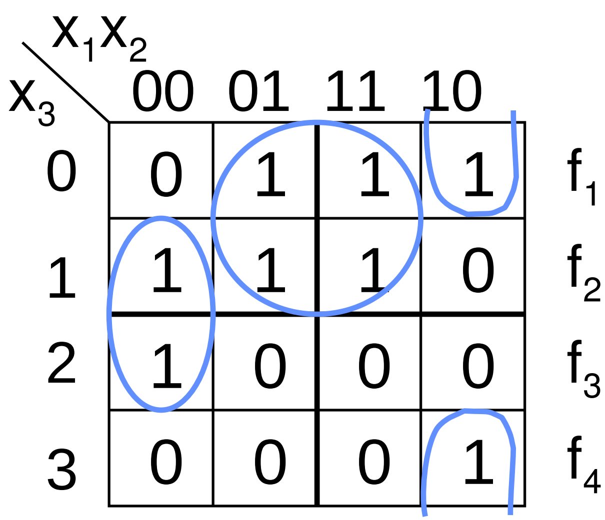
- Minimize using set-literals:

$$J_i x + J_j x = J_{\{i,j\}} x$$

Example

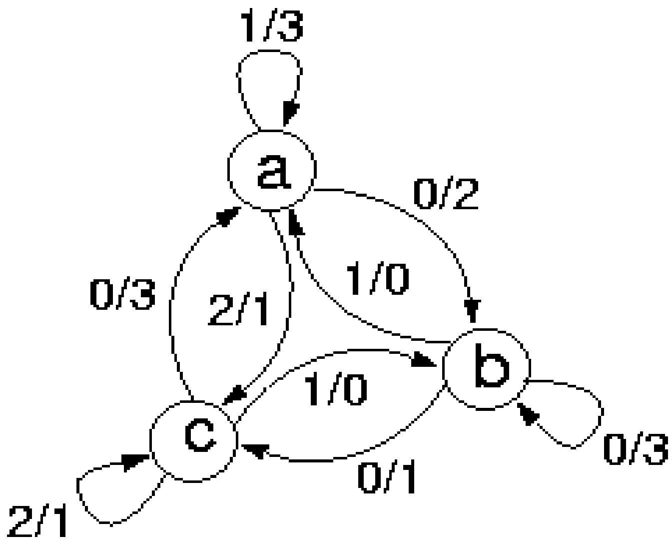


5 cubes



3 cubes

Example of describing a FSM



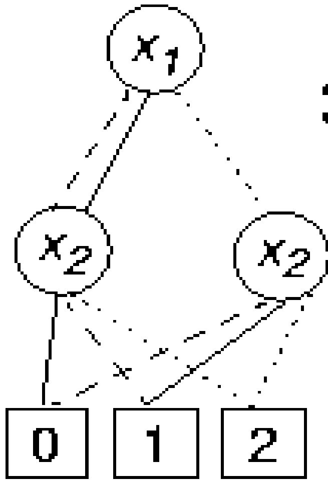
states ∈ {a,b,c} = {0,1,2}

inputs ∈ {0,1,2}

outputs ∈ {0,1,2,3}

pr.s.	*in.*	*next s.*	*out.*
0	0	1	2
0	1	0	3
0	2	2	1
1	0	1	3
1	1	0	0
1	2	2	1
2	0	0	3
2	1	1	0
2	2	2	1

Resulting MDD for FSM



3 non-terminal nodes

- BDDs for this function would have 8 non-terminal nodes in common

MVL logic circuit

- Boolean 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

Theoretical advantages of MVL circuits

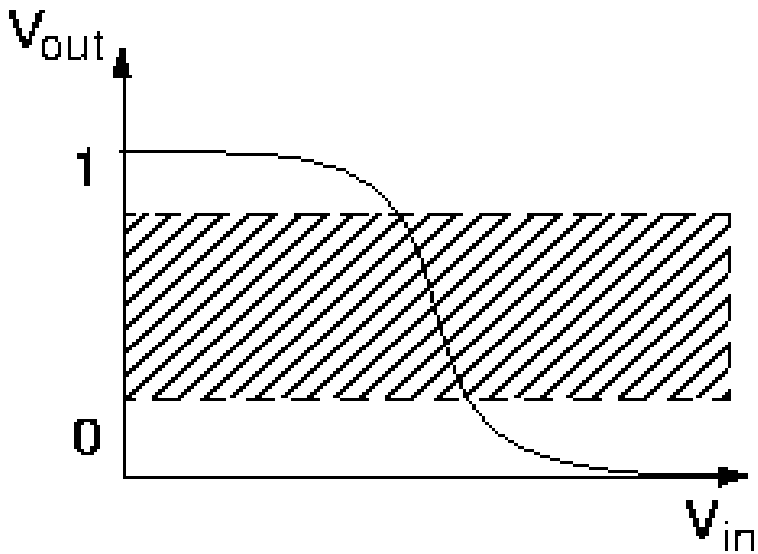
- In a typical VLSI chip, about 70% of the chip area is devoted to interconnection, 20% to insulation and 10% to devices
- In a multiple-valued circuit:
  - wires carry more information - saving in the number of wires and in insulation between wires
  - pins carry more information - saving in pins
- Alternative to binary number systems allow fast arithmetic operations
- MVL storage allows to store more bits of information per memory cell

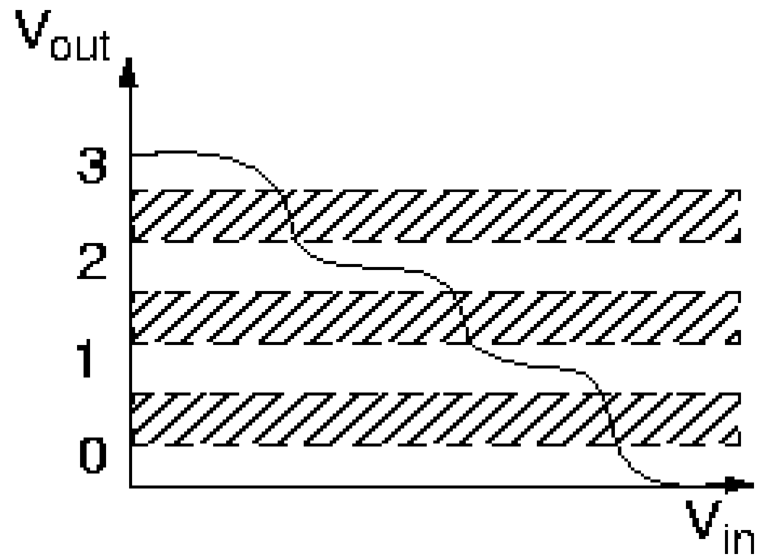
Why aren't MVL circuits widely used?

- 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 theoretical 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

MVL memories

- 4-valued Flash memories
  - digital
  - analog
- 4-valued DRAM memories

Traditional techniques for memory density increase

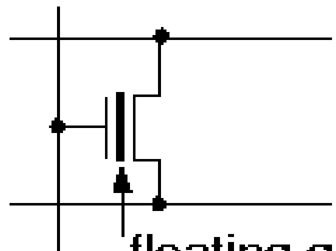
- Process scaling
  - 70% reduction in the minimum design rule for each generation
- Process scaling + better cell structure
  - 50% larger chips size each generation
- By 1995, both techniques reached their limits
  - using 4-valued logic allowed in 1997 to double the chip density without increasing the die size

Flash memory

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

4-valued Flash

- 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

Technical parameters

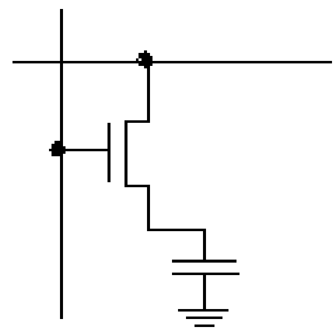
	Intel	NEC	Samsung Electron	Hitachi and Mitsubishi
Memory size	32-Mbit	64-Mbit	128-Mbit	256-Mbit
Process	0.6-μm	0.4-μm	0.4-μm	0.26-μm
Die size	152 mm^2	98 mm^2	117 mm^2	139 mm^2
Power supply	5 V	3.3 V	3.3 V	3 V
Access time	120 ns	80 ns	25 ns	50 ns

Dymanic RAM (DRAM)

- DRAM - volatile general purpose memory
- applications: main processing units, computer operating systems, video and audio data processing

DRAM

- 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

NEC's 4Gbit 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)
  - capable of storing 47 minutes of full-motion video
- high-speed access to data (standard)
  - Jurassic Park in real time

Prospects of MVL circuits

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

Future systems