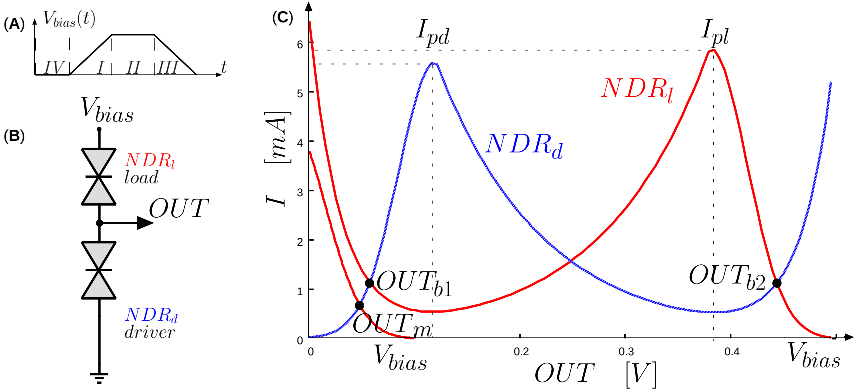
The 15th Asia and South Pacific
Design Automation Conference
January 21, 2010

Generalised Threshold Gate Synthesis based on AND/OR/NOT Representation of Boolean Function

Marek A. Bawiec (speaker), Maciej Nikodem
Wrocław University of Technology

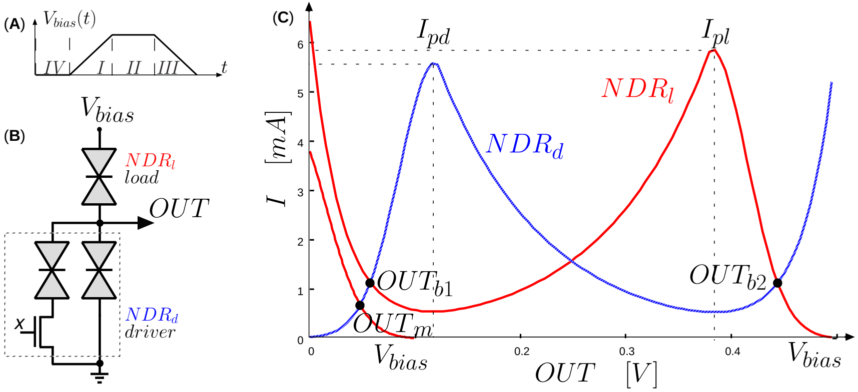
NDRs in MOBILE Logic Circuits



NDR Negative Differential Resistance

MOBILE Monostable-Bistable Transition Logic Element

NDRs in MOBILE Logic Circuits



The simplest function - an inverter

Circuit Structures

- linearly separable Boolean functions

  threshold gate (TG)

- any arbitrary Boolean function
  - multi–threshold threshold gate (MTTG)
  - generalized threshold gate (GTG)

Synthesis Algorithms

- no synthesis algorithm for MTTG
- synthesis algorithm for GTG, but:
  - input function in Reed–Muller form
  - lower number of branches possible
  - lower number of switching elements possible

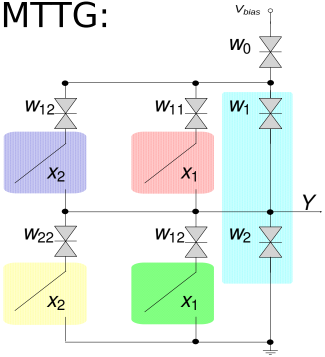
Challenge

New Synthesis Algorithm

- input in SOP form
- improvement of circuit structure
  - simpler functions
  - lower number of branches
- algorithm efficiency

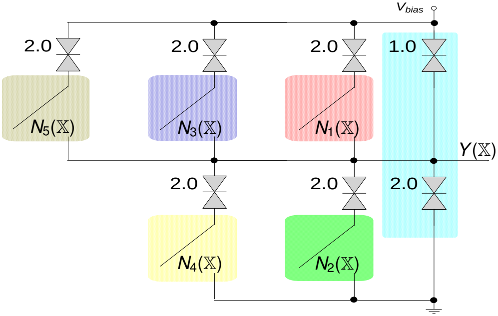
MTTG vs. GTG

MTTG:



- each branch – one NDR with **one transistor**
- needs different NDRs (weights)
- implements multi-threshold threshold function directly
- number of NDRs increases quadratically with number of inputs

GTG:

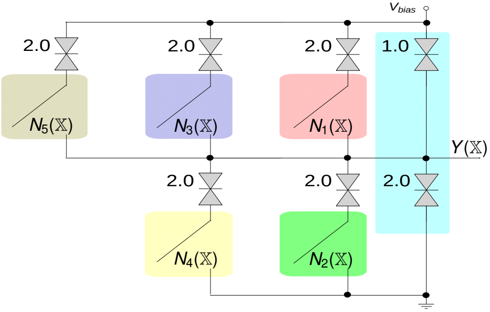


- each branch – one NDR and **serial-parallel transistors network**
- uniform NDRs elements but one
- implements any *n*–variable Boolean function
- number of NDRs increases proportionally with number of inputs

GTG – Formal Model

IF:

- $N_i(\mathbb{X})$–unate function
- "upper" functions: $N_i(\mathbb{X})$
  where $i \bmod 2 = 1$
  "lower" functions: $N_i(\mathbb{X})$
  where $i \bmod 2 = 0$



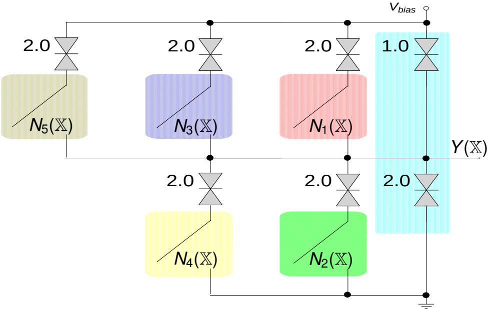
THEN:

$$Y_l(\mathbb{X}) = \begin{cases} 0 & l = 0 \\ Y_{l-1}(\mathbb{X}) + N_l(\mathbb{X}) & l = 2k-1 \\ Y_{l-1}(\mathbb{X})\overline{N_l(\mathbb{X})} & l = 2k \end{cases}$$

GTG – Formal Model

IF:

- $N_i(\mathbb{X})$–unate function
- "upper" functions: $N_i(\mathbb{X})$
  where $i \bmod 2 = 1$
  "lower" functions: $N_i(\mathbb{X})$
  where $i \bmod 2 = 0$
- $N_i(\mathbb{X})N_{i+j}(\mathbb{X}) = N_{i+j}(\mathbb{X})$



THEN:

$$\boldsymbol{Y(\mathbb{X}) = \bigoplus_{i=1}^{k} N_i(\mathbb{X})}$$

The Smallest Unate Function

When $Y_{i-1}(\mathbb{X})$ is represented as sum of minterms:

$$Y_{i-1}(\mathbb{X}) = \bigcup_{i} \mathbb{M}^{(i)}_{i-1}(\mathbb{X}),$$

then

$$N_i(\mathbb{X}) = \bigcup_{i} \mathbb{M}^{(i)}_{i-1}(\mathbb{X}) + \bigcup_{j} \mathbb{C}^{(j)}_{j-1}(\mathbb{X}),$$

where $\mathbb{C}^{(j)}_{j-1}(\mathbb{X})$ is a cofactor of $\mathbb{M}^{(i)}_{i-1}(\mathbb{X})$.
When simplified to SOP form, we get $N_i(\mathbb{X}) = \bigcup_{k} I^{(k)}_{i-1}$, where $I^{(k)}_{i-1}$ are positive unate function, therefore $N_i(\mathbb{X})$ is also a positive unate function.

Corollary To obtain $N_i(\mathbb{X})$ implied by $Y_{(i-1)}(\mathbb{X})$ it is enough to remove complemented variables from $Y_{(i-1)}(\mathbb{X})$ represented in SOP form.

Key Observation

$Y_0(\mathbb{X})$ in sum of products (SOP) form

$$\Downarrow$$

repeat until $Y_i(\mathbb{X}) = 0$

- obtain smallest unate $N_i(\mathbb{X})$ implied by $Y_{i-1}(\mathbb{X})$
- $Y_i(\mathbb{X}) = Y_{i-1}(\mathbb{X}) \oplus N_i(\mathbb{X})$

$$\Downarrow$$

$$Y_0(\mathbb{X}) = N_1(\mathbb{X}) \oplus N_2(\mathbb{X}) \oplus \ldots \oplus N_n(\mathbb{X})$$

where: $N_i(\mathbb{X})N_{i+j}(\mathbb{X}) = N_{i+j}(\mathbb{X})$

Proposed Synthesis Algorithm

Require: n–variable Boolean function $Y(\mathbb{X})$

Ensure: NDR_l vs. NDR_d relation, and $N_i(\mathbb{X})$ functions

if $Y(0^n) = 0$ **then**
  $\mathrm{NDR}_l > \mathrm{NDR}_d$,
else
  $\mathrm{NDR}_l < \mathrm{NDR}_d$,
  $Y(\mathbb{X}) = 1 \oplus Y(\mathbb{X})$,
end if
set $i = 1$,
find the smallest unate function $N_i(\mathbb{X})$ implied by $Y(\mathbb{X})$,
if $Y(\mathbb{X}) = N_i(\mathbb{X})$ **then** exit algorithm
calculate $Y_i(\mathbb{X})$ such that $Y(\mathbb{X}) = N_i(\mathbb{X}) \oplus Y_i(\mathbb{X})$,
while $Y_i(\mathbb{X}) \neq 0$ **do**
  find the smallest unate function $N_{i+1}(\mathbb{X})$ implied by $Y_i(\mathbb{X})$,
  calculate $Y_{i+1}(\mathbb{X})$ such that $Y_i(\mathbb{X}) = N_{i+1}(\mathbb{X}) \oplus Y_{i+1}(\mathbb{X})$,
  set $i = i + 1$,
end while

Proposed Synthesis Algorithm

Require: n–variable Boolean function $Y(\mathbb{X})$
Ensure: NDR_l vs. NDR_d relation, and $N_i(\mathbb{X})$ functions

if $Y(0^n) = 0$ **then**
    $\mathrm{NDR}_l > \mathrm{NDR}_d$,
else
    $\mathrm{NDR}_l < \mathrm{NDR}_d$,
    $Y(\mathbb{X}) = 1 \oplus Y(\mathbb{X})$,
end if
set $i = 1$,
find the smallest unate function $N_i(\mathbb{X})$ implied by $Y(\mathbb{X})$,
if $Y(\mathbb{X}) = N_i(\mathbb{X})$ **then** exit algorithm
calculate $Y_i(\mathbb{X})$ such that $Y(\mathbb{X}) = N_i(\mathbb{X}) \oplus Y_i(\mathbb{X})$,
while $Y_i(\mathbb{X}) \neq 0$ **do**
    find the smallest unate function $N_{i+1}(\mathbb{X})$ implied by $Y_i(\mathbb{X})$,
    calculate $Y_{i+1}(\mathbb{X})$ such that $Y_i(\mathbb{X}) = N_{i+1}(\mathbb{X}) \oplus Y_{i+1}(\mathbb{X})$,
    set $i = i + 1$,
end while

Proposed Synthesis Algorithm

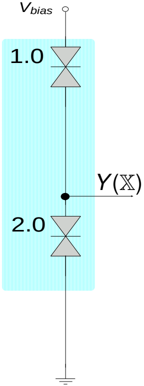
Require: n–variable Boolean function $Y(\mathbb{X})$
Ensure: NDR_l vs. NDR_d relation, and $N_i(\mathbb{X})$ functions

```
if Y(0^n) = 0 then
    NDR_l > NDR_d,
else
    NDR_l < NDR_d,
    Y(X) = 1 ⊕ Y(X),
end if
set i = 1,
find the smallest unate function N_i(X) implied by Y(X),
if Y(X) = N_i(X) then exit algorithm
calculate Y_i(X) such that Y(X) = N_i(X) ⊕ Y_i(X),
while Y_i(X) ≠ 0 do
    find the smallest unate function N_{i+1}(X) implied by Y_i(X),
    calculate Y_{i+1}(X) such that Y_i(X) = N_{i+1}(X) ⊕ Y_{i+1}(X),
    set i = i + 1,
end while
```

How It Works for: $Y(\mathbb{X}^3) = x_1\overline{x_2} + x_2x_3$

step 1: for $Y(0^3)$ setup NDR_d and NDR_l relation

- $Y(0^3) = 0 \Rightarrow NDR_d > NDR_l$
- $N_1(\mathbb{X}^3) = x_1 + x_2x_3$
  $Y_1(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus Y(\mathbb{X}^3) = x_1x_2\overline{x_3}$
- $N_2(\mathbb{X}^3) = x_1x_2$
  $Y_2(\mathbb{X}^3) = N_2(\mathbb{X}^3) \oplus Y_1(\mathbb{X}^3) = x_1x_2x_3$
- $N_3(\mathbb{X}^3) = x_1x_2x_3$
  $Y_3(\mathbb{X}^3) = N_3(\mathbb{X}^3) \oplus Y_2(\mathbb{X}^3) = 0$

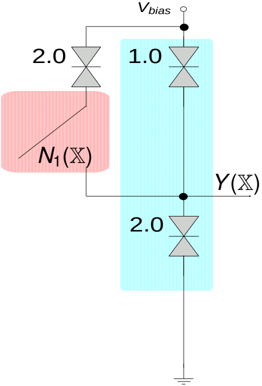


$$Y(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus N_2(\mathbb{X}^3) \oplus N_3(\mathbb{X}^3)$$

How It Works for: $Y(\mathbb{X}^3) = x_1\overline{x_2} + x_2x_3$

step 2: calculate smallest unate function $N_1(\mathbb{X}^3)$ implied by $Y(\mathbb{X}^3)$

- $Y(0^3) = 0 \Rightarrow NDR_d > NDR_l$
- $N_1(\mathbb{X}^3) = x_1 + x_2x_3$
  $Y_1(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus Y(\mathbb{X}^3) = x_1x_2\overline{x_3}$
- $N_2(\mathbb{X}^3) = x_1x_2$
  $Y_2(\mathbb{X}^3) = N_2(\mathbb{X}^3) \oplus Y_1(\mathbb{X}^3) = x_1x_2x_3$
- $N_3(\mathbb{X}^3) = x_1x_2x_3$
  $Y_3(\mathbb{X}^3) = N_3(\mathbb{X}^3) \oplus Y_2(\mathbb{X}^3) = 0$

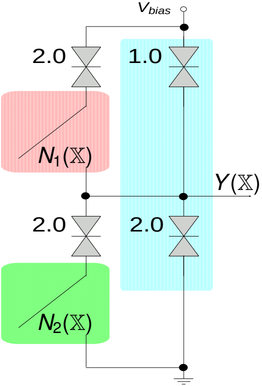


$Y(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus N_2(\mathbb{X}^3) \oplus N_3(\mathbb{X}^3)$

How It Works for: $Y(\mathbb{X}^3) = x_1\overline{x_2} + x_2x_3$

repeat step 2: calculate smallest unate function $N_2(\mathbb{X}^3)$ implied by $Y_1(\mathbb{X}^3)$

- $Y(0^3) = 0 \Rightarrow NDR_d > NDR_l$
- $N_1(\mathbb{X}^3) = x_1 + x_2x_3$
  $Y_1(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus Y(\mathbb{X}^3) = x_1x_2\overline{x_3}$
- $N_2(\mathbb{X}^3) = x_1x_2$
  $Y_2(\mathbb{X}^3) = N_2(\mathbb{X}^3) \oplus Y_1(\mathbb{X}^3) = x_1x_2x_3$
- $N_3(\mathbb{X}^3) = x_1x_2x_3$
  $Y_3(\mathbb{X}^3) = N_3(\mathbb{X}^3) \oplus Y_2(\mathbb{X}^3) = 0$

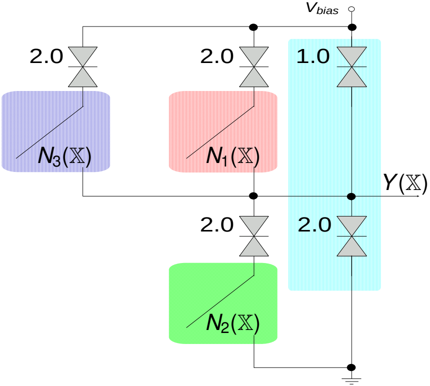


$$Y(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus N_2(\mathbb{X}^3) \oplus N_3(\mathbb{X}^3)$$

How It Works for: $Y(\mathbb{X}^3) = x_1\overline{x_2} + x_2x_3$

repeat step 2: calculate smallest unate function $N_3(\mathbb{X}^3)$ implied by $Y_2(\mathbb{X}^3)$

- $Y(0^3) = 0 \Rightarrow NDR_d > NDR_l$
- $N_1(\mathbb{X}^3) = x_1 + x_2x_3$
  $Y_1(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus Y(\mathbb{X}^3) = x_1x_2\overline{x_3}$
- $N_2(\mathbb{X}^3) = x_1x_2$
  $Y_2(\mathbb{X}^3) = N_2(\mathbb{X}^3) \oplus Y_1(\mathbb{X}^3) = x_1x_2x_3$
- $N_3(\mathbb{X}^3) = x_1x_2x_3$
  $Y_3(\mathbb{X}^3) = N_3(\mathbb{X}^3) \oplus Y_2(\mathbb{X}^3) = 0$

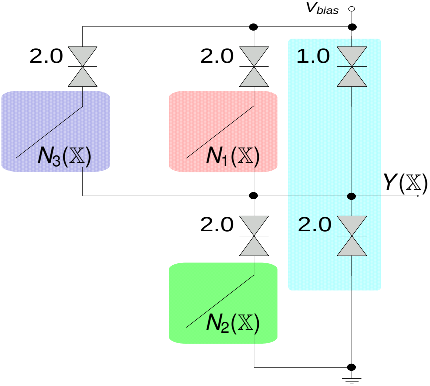


$Y(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus N_2(\mathbb{X}^3) \oplus N_3(\mathbb{X}^3)$

How It Works for: $Y(\mathbb{X}^3) = x_1\overline{x_2} + x_2x_3$

max n iteration needed

- $Y(0^3) = 0 \Rightarrow NDR_d > NDR_l$
- $N_1(\mathbb{X}^3) = x_1 + x_2x_3$
  $Y_1(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus Y(\mathbb{X}^3) = x_1x_2\overline{x_3}$
- $N_2(\mathbb{X}^3) = x_1x_2$
  $Y_2(\mathbb{X}^3) = N_2(\mathbb{X}^3) \oplus Y_1(\mathbb{X}^3) = x_1x_2x_3$
- $N_3(\mathbb{X}^3) = x_1x_2x_3$
  $Y_3(\mathbb{X}^3) = N_3(\mathbb{X}^3) \oplus Y_2(\mathbb{X}^3) = 0$



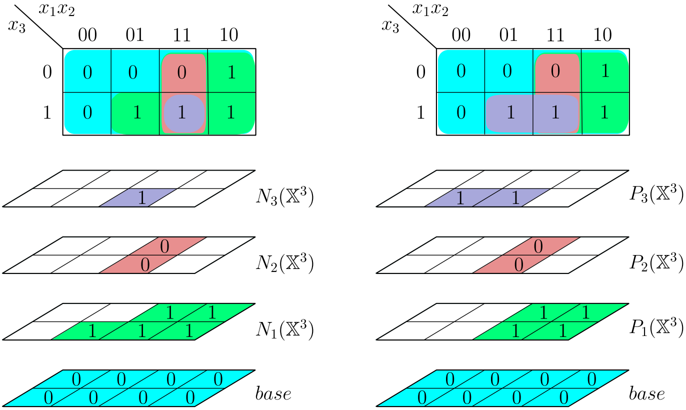
$$Y(\mathbb{X}^3) = N_1(\mathbb{X}^3) \oplus N_2(\mathbb{X}^3) \oplus N_3(\mathbb{X}^3)$$

Algorithm Properties

- at most n iteration needed
- at most $n+2$ branches needed
- no need to use complementary transistors pair
- input – function in SOP form
- output – set of SOP switching functions

	[2]	[4]	Our
Theoretical no. of branches in GTG circuit	$\leq 2^n$	$\leq n+2$	$\leq n+2$
Synthesis algorithm	no	yes	yes
Input function form	N/A	Reed-Muller	SOP
No. of branches in synthesized circuit	N/A	$\leq 2^n$ ($\leq n+2$ on average)	$\leq n+2$
No. of iterations in algorithm's main loop	N/A	$O(2^n)$	$O(n)$

- assumption $N_i(\mathbb{X})N_{i+j}(\mathbb{X}) = N_{i+j}(\mathbb{X})$ restricts number of possible solutions
- when released other solutions exist



different number of switching elements – 8 vs. 5

Conclusion

- GTG switching functions can be synthesized directly from SOP form
- there are at most $n + 2$ branches in the circuit
- algorithm gives the best known solution in terms of number of branches
- further improvements are possible

The 15th Asia and South Pacific Design Automation Conference
January 21, 2010

Generalised Threshold Gate Synthesis based on AND/OR/NOT Representation of Boolean Function

Marek A. Bawiec (speaker), Maciej Nikodem
Wrocław University of Technology