Simulation and Implication using a Transfer Function Model for Switching Logic

SMU | Bobby B. Lyle School of Engineering

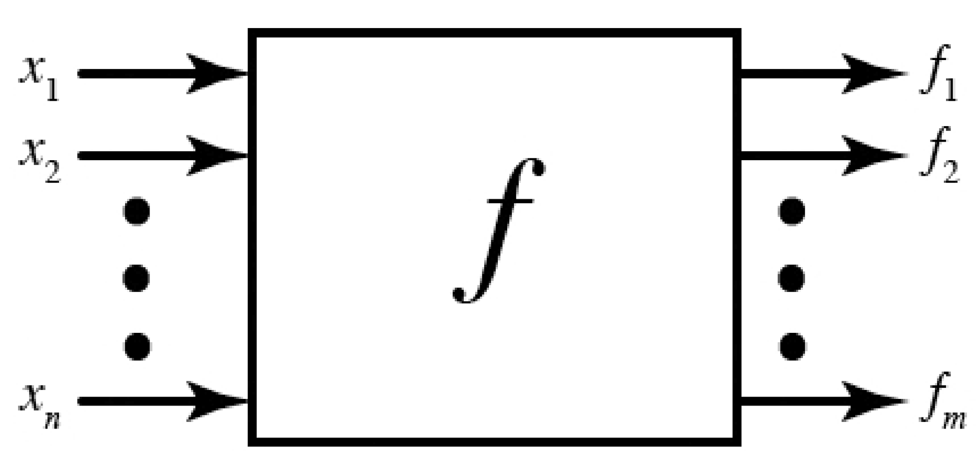
By: David Kebo Houngninou, Advisor: Dr. Mitch Thornton

Computer Science and Engineering Department, Bobby B. Lyle School of Engineering

Southern Methodist University, P.O. Box 750122, Dallas, TX 75275-0122

SMU | HACNet High Assurance Computing And Networking Labs

WHAT IS A TRANSFER FUNCTION ?



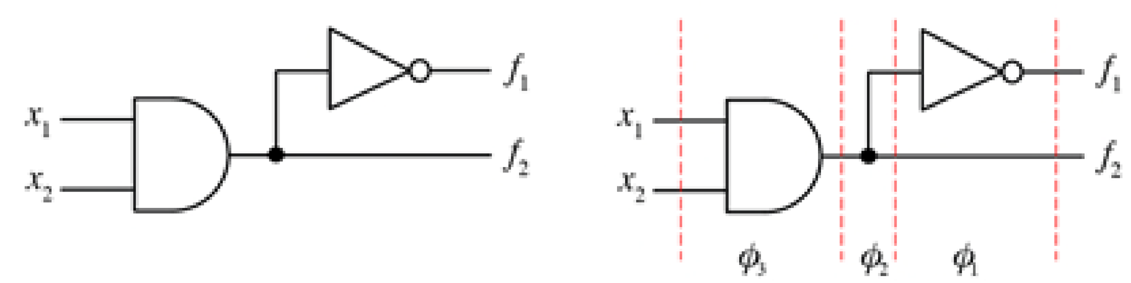
A **transfer function** is a mathematical function relating the output or response of a system to the input or stimulus.

STEPS

The technique for deriving a transfer matrix directly from a logic network involves:

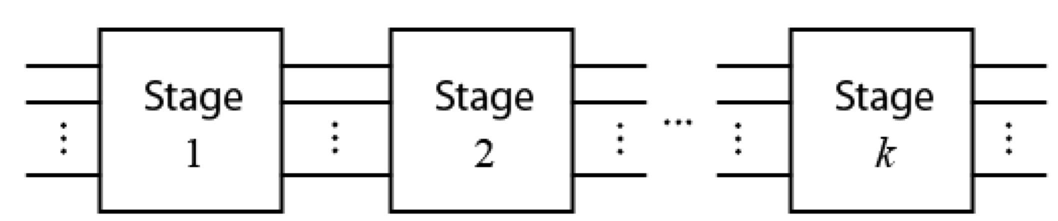
1. Parsing HDL into a structure
2. Performing Levelization
3. Partitioning the logic network
4. Computing the transfer matrix for each partition (Outer Product of parallel elements)
5. Computing the overall transfer matrix (Direct matrix product of each cascade)

PARTITIONING



Partitioning separates the network into series or cascades of sub circuits. The individual cascades are determined by forming partition cuts such that all components within the cascade stage are in a parallel arrangement.

$$T = (\text{AND}) \,.\, (\text{FO}) \,.\, (\text{NOT} \oplus \text{I})$$



The netlist is partitioned into a set of serial cascade stages. All cascades are combined using an outer product.

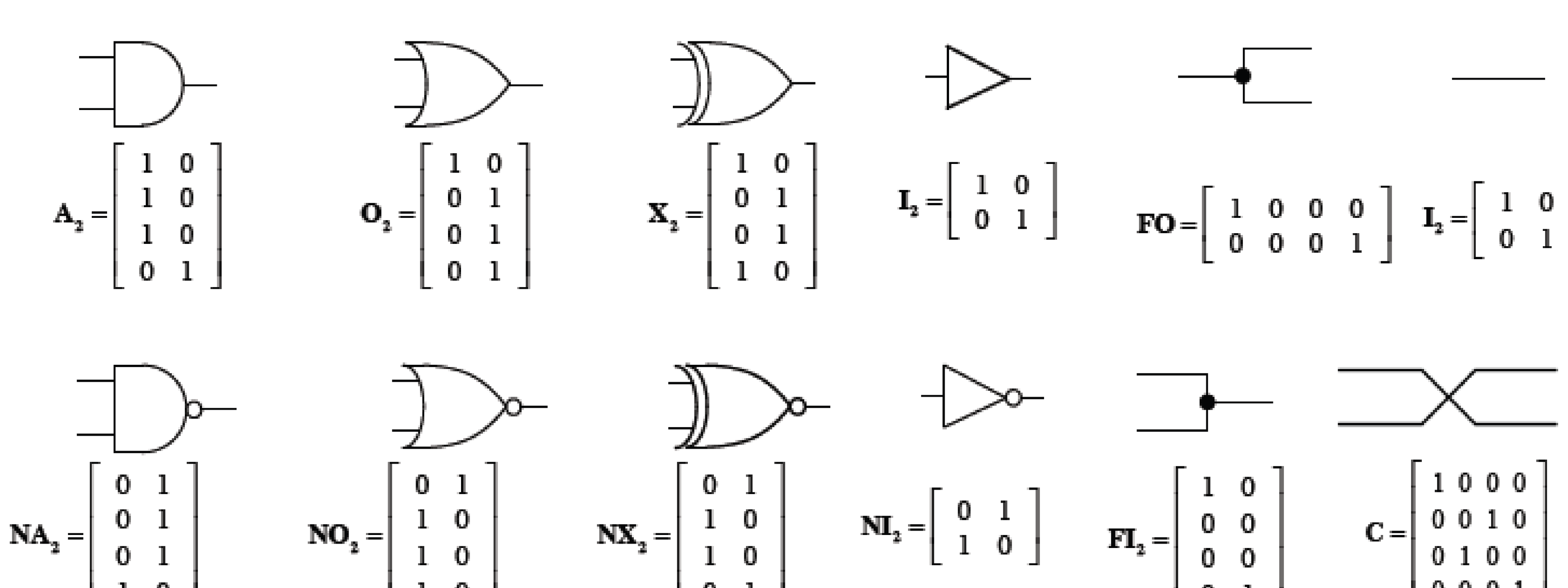
x1	x2	f1	f2
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Corresponding transfer function matrix

LOGIC GATES AND TRANSFER MATRICES



$$A_2 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad O_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad X_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad FO = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$NA_2 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad NO_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \quad NX_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad NI_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad FI_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

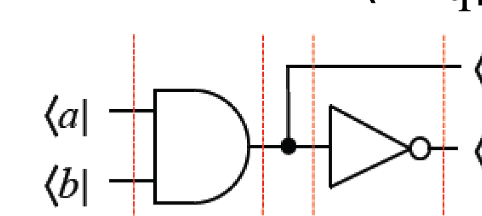
VECTOR SPACE INFORMATION MODEL

	CONVENTIONAL METHOD	OUR METHOD
INFORMATION	bits {0,1}	row vectors $\{\langle 0|, \langle 1|\}$
LOGIC GATE	small truth table or Boolean function	small transfer matrix
LOGIC CIRCUIT	large truth table	larger transfer matrix
SIMULATION	discrete event algorithm	solve $\langle f| = \langle x|\, T$
IMPLICATION	backtracking algorithm	solve $\langle x| = \langle f|\, T^{-1}$

SIMULATION

The output response of a logic network stimulated by input $\langle x_q|$ and modeled by a transfer matrix T is:

$$\langle F_q| = \langle x_q|\; T$$



$$\mathbf{T} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right)$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

IMPLICATION

Implication is the inverse problem of simulation. In this case, an output response and the characterization of a logic network are known and it is possible to compute the corresponding input stimuli.

$$\langle X| = \langle f|\; T^{-1}$$

The inverse transfer function T^{-1} is used to determine a corresponding input stimulus given an output response.

EXPERIMENTAL RESULTS

NAME	IN/OUT	STAGES	PARTITION TIME (ms)	MATRIX TIME (ms)
i3	2/3	3	3.00	5.505
test1	3/3	6	7.28	4.794
xor5	5/1	4	1.73	6.882
majority	5/1	6	11.8	17.71
C17	5/2	7	5.00	22.75
rd53	5/3	6	5.32	10.10
squar5	5/8	9	19.5	922.1
con1	7/2	6	7.09	546.1
rd73	7/3	8	5.01	37.33
radd	8/5	11	12.3	1107
x2	10/7	9	11.4	846.2
cm85a	11/3	11	9.78	1586.2
alu1	12/8	5	8.88	521.9

APPLICATIONS

EDA Tools implication and simulation

Satisfiability 1

Equivalence checking =