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

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.

\[ T = (\text{AND}) \cdot (\text{FO}) \cdot (\text{NOT } \oplus 1) \]

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

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)

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.

\[ \langle X \rangle = \langle f \rangle \cdot T^{-1} \]

The vector space information model uses a transfer matrix to represent the logic network. The information is represented as:
- 0/1 bits
- Row vectors \( \langle 0 \rangle, \langle 1 \rangle \)
- Small truth table or Boolean function
- Small transfer matrix
- Large truth table
- Larger transfer matrix

Simulation involves solving the equation:

\[ \langle f \rangle = \langle x \rangle \cdot T \]

Implication involves solving the equation:

\[ \langle x \rangle = \langle f \rangle \cdot T^{-1} \]

The experimental results show the performance comparison between conventional and our method.

### Applications
- EDA Tools implication and simulation
- Satisfiability
- Equivalence checking