



*An umbrella package of software tools that produce solutions to the “Buffer Tree Synthesis Problem”. There are three separate tools that produce solutions in different and sometimes overlapping solution spaces.*

**Technology Reference**  
CW048, CW049, CW050

**Contact**  
Lanny Feder  
Director, Engineering  
Intellectual Property  
(M/C 159)  
851 South Morgan Street  
Chicago, IL 60607-7043

Phone: 312-413-5496  
Fax: 312-413-5846  
E-mail: Lfeder@uic.edu

**Lead Inventor**  
John Lillis

**Field**  
Design of Semiconductor  
Chips

**Key Words**  
Semiconductors  
Semiconductor  
Manufacture  
Floorplanning  
CAD

**Stage of Development**  
Validated by experimental  
results  
Ready for commercial use

**Status**  
Seeking non-exclusive  
Licensees

Each of these tools provides Buffer Tree Synthesis Techniques that are used in laying out the components on semiconductor chips and that differ in the solution method and solution space.

- CW048 is based On A Given Topology, Sink Partitioning and Dynamic Programming (S-Tree)
- CW049 is based On A Sink Permutation Constraint and Dynamic Programming (P-Tree)
- CW050 is based On A Sink Permutation Constraint, Sink Partitions and Dynamic Programming (SP-Tree)

**What was known before discovery**

Commercial and proprietary packages for buffer tree synthesis are available. Other packages either adopt a multi-step process to account for topology construction, topology embedding and buffering or are lacking in the practical relevance of problem formulation. In contrast the tools in this invention, simultaneously optimize across all three dimensions while adopting a general and practically useful formulation.

**New Discovery**

There are two important aspects of the invention. **First** is the scope of the intrinsic problem formulation. The approach takes as input technology parameters, pin locations, timing requirements for sinks, buffer and routing blockages, a buffer library and wiring and buffering costs. From the constraints and freedoms imposed by this input, the output is to be a set of solutions (embedded buffer trees) that provide a tradeoff between the performance of the solution (given by the required arrival time at the input of the tree or equivalently, the slack) and the cost of the solution. **Second** is the solution method to find such solutions.

Three important characteristics of these tools are:

**Generality and practical relevance of the problem formulation**

- Ability to deal with blockages including buffer and routing blockages
- Ability to capture cost/performance tradeoffs
- R-C delay models must be incorporated
- Ability to handle varying sink polarity requirements

**Runtime Efficiency**

- Run time and memory usage are clearly important characteristics
- The ability to trade off these parameters with solution quality
- SP-Tree yields the best solution quality but has the lowest runtime efficiency followed generally by P-Tree and S-Tree.

**Quality of the solution**

Both solution quality and predictability of quality are important traits of the tools. In all three tools, experimental results have demonstrated such quality and robustness. This strength is achieved in each tool by finding optimum solutions in carefully designed exponentially large solution spaces. (Both topology and embedding spaces are simultaneously explored.) The exact nature of the solution spaces is a property of the individual tools (S-Tree, P-Tree and SP-Tree).

Of notable interest is the ability of S-tree and SP-Tree to deal effectively not only with physical locality but also with widely varying timing requirements or “temporal locality”. This capability often results in lower resource usage with equivalent or better timing properties.