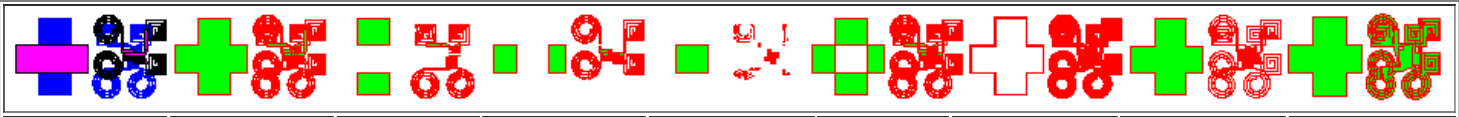
*This page is first released 1-01-98*
*Last Page Update: 3-3-2007*

# Boolean: GDSII viewer/editor + (boolean) operations on sets of 2d polygons.

# Operations (some)



| original | merge | a-b | b-a | intersect | exor | makering | negative_cor | positive_cor |
|---|---|---|---|---|---|---|---|---|

## Index


- Introduction
- screenshots
- Copyright License?
- C ++ source, executables and documentation for download
- Acknowledgments, Feedback, mailing list
- documentation (algorithm)
- documentation commands and graphical interface and GDSII format and KEY format


## Introduction

**The program is a viewer and editor for:**

- GDSII files (tree of structures containing polygons and polylines on layers).
- KEY files (own made extended GDSII in ascii format)

  supports : circles, arcs, text, polygons(including arc segments), polylines(including arc segments) , boxes, structures, references to structures, all with a width parameter,
  user defined properties to primitives and structures.
- **DAVID MANN** files (david man flash format for mask plotting)

Mainly the program is used in the electronic chip design industry. But it it offers much more then that. Many effort has gone in automating the jobs to be done.
Therefore the commands isued by the GUI, go through a command interpreter as ASCII strings, in order to record them and replay it them afterwords.

**Features:**

- It allows to draw primitives on a chosen layer, and to manipulate them. A (limited) undo mechanism is available to support this.
- stack oriented tools allows zooming while drawing new primitives and editing them. This also makes it possible to draw extremely accurate.
- primitives on the layers or/can be transparent
- colors and fill patterns can be set on a layer basis
- drawing order of layers can be changed
- a hiearchy of pictures, named structures, can be handled and manipulated seperately
- saving as a bitmap and other formats.
- measuring distances
- adding user defined properties to primitives

Next to this it contains several algorithms for performing 2D polygon operations (boolean, offset, arc recognition etc.). All the primitives are situated on layers. The operations take two sets of layers (each set containing all primitives on the selected layers). The sets are the operands of the (boolean) operation, a destination layer is chosen to contain the result of the operation.

**Operation:**

- boolean OR AND EXOR A-B B-A
- positive and negative process offset
- circle recognition in polygon and polyline data.
- move copy delete etc.
- transformations (scaling , rotation , moving)
- flatten the hiearchy of the drawing
- drivers for CNC (laser and milling machinery)

The C++ source code, executables and sample files are available.

## Copyright License?

The code is written by Klaas Holwerda, it is licensed as GPL.

For commercial use in proprietary software, contact me.- mail me .

## The C++ Source and executables and documentation

## Download PAGE

**Feedback** In case of problems installing etc. and for remarks, suggestions and bug fixes - mail me .
Also i always appreciate some feedback on where this program is used for, or what you think of it.

## Documentation Technical

### The algorithm description

The basis of the boolean algorithms is scanlines. For the basic principles the book: Computational Geometry an introduction by Franco P Preparanta and Michael Ian Shamos is very good. For the rest many sources gave small hints into the final direction the algorithms were implemented.
The documentation explains the principles of the algorithm, i am working on a new version of this document.
But it evolves more quickly (for sure the ideas), to keep up with it.
See COMPLETE BOOLEAN DISCRIPTION to see a more detailed description.
Roughly the following steps are involved for performing a boolean operation:

1. convert all polygons to graph structures
2. combine all graph segments in to one big graph
3. calculate all intersections between the graphs using a scanbeam and insert them as extra vertices/segments
    a. first at 0 degrees angle, snapping vertices and lines that are within the snapping range in vertical direction.
    b. secondly at 90 degrees angle, snapping vertices and lines that are within the snapping range in horizontal direction.

    (just swapping x,y coordinates)
4. set appropriate segment flags for all boolean operations at once using a scanbeam
5. extract for the given boolean operation the resulting graphs (hole graphs and outside graphs).
6. link holes into outside graphs using a scanbeam.
7. extract resulting polygons from the graph structures again

## Command documentation

- **Command overview**

  The graphical and the command line version of the program, are based on a command interpreter. In order to make the code more platform independent and to make it possible to record the commands to a file for later repetition. Click for Command Help to see an overview of the complete set of commands.

- **Graphical Interface**

  The graphical interface help system is based on HTML here (graphical help system) you will find the context sensitive help.
  The GDSII and KEY format description is also part of it.

---