CALD: Computer-Aided Logic Design System

Géry Bioul, Daniel Simonelli*, Valeria Otaegui y Susana Heineke
Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Provincia de Buenos Aires
INCA Institute – TE: +54 2293 432466 - Paraje Arroyo Seco, (7000) Tandil, Argentina
e-mail: dsimonel@exa.unicen.edu.ar

Abstract: *This paper presents a new design tool (CALD) for logical circuit design. This system is shown to be powerful because of its relationship with an HDL language and thanks to a user friendly drawing interface which allows both automatic and manual connection drawing processes.*

The system is limited to circuit drawing generation while actual physical simulation is not considered.

This paper carries out a comparative analysis of the CALD along with classic similar tools from the market, such as B^2Logic, EWB, Circuit Maker.

The discussion is illustrated through Windows environment applications.

The CALD is mainly an education-oriented tool; so synthesis oriented design systems such as Viewlogic, Orcad, Altera or Xilinx Foundation are not dealt with.

Introduction

The starting point of CALD has been the analysis of classic tools for generating logical circuits. From this analysis, good features have been pointed out in order to preserve or improve them, while bad features have been identified with the purpose of elimination or, as far as possible, impact alleviation.

Routing and visualization processes have been improved while intermediate code generation is reviewed for its extension to simulation and synthesis applications.

The CALD system is providing effective interface for capturing circuit schemes with both didactical and industrial applications. In this context, the system is addressed to educational institutions as well as technological design groups.

The innovative CALD feature rests upon the following:

- circuit drawings can be generated from a custom CALD-HDL hardware description language mixed with manual user operations,
- from any compatible circuit drawing CALD automatically generates an HDL description.

CALD-HDL is a structural subset of the VHDL language. As such it is fully portable as a front-end of standard design tools [HEI99][BIO2000].

Standard design tools reviewed in this paper are the following:

- Electronics Workbench. Version 4.0d. Interactive Image Technologies Ltd. 1992-1995.
- Electronics Workbench EDA. Demo. Interactive Image Technologies Ltd. 1992-1996.
- CircuitMaker Version 3.03 MicroCode Engineering, 1993-1995.
- B-Squared Logic. Version 3.0.8a. Beige Bag Software. 1993.

Friendly interfaces

Nowadays computer systems are available to an increasing number of people. One of the main tasks for the designer is to cope with abilities and needs of the potential users so that fulfilling needs ends in enhancing abilities.

Friendly interface refers to something more powerful than a simple *help* feature. Actually it consists of an environment in which the user can handle tasks with minimum effort and maximum profit.

Within this context the first step for the tool designer is to identify functional requirements and to define a task set in the most

* - Supported by CIC (Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

straightforward way. In this respect, using environments familiar to the user (Windows, Unix, ...) eases both understanding/learning and operating the tool.

The final goal is to provide an interface with a classic basic structure in its appearance. Outstanding examples are Microsoft tools: each of them are generally complying with a set of menus with standard commands to which are added specific ones such as state bar or tool bar.

Menu systems are attractive for eliminating needs for tedious training and memorizing complex command sequences.

MDI versus SDI applications

A SDI application (Single Document Interface) only allows a single active document, i.e. a single opened window. On the other hand, a MDI application (Multiple Document Interface) allows multiple simultaneous working documents. This implies several simultaneous opened windows. The two reviewed EWB versions together with CircuitMaker are SDI applications. B^{2}Logic is MDI (fig. 1).

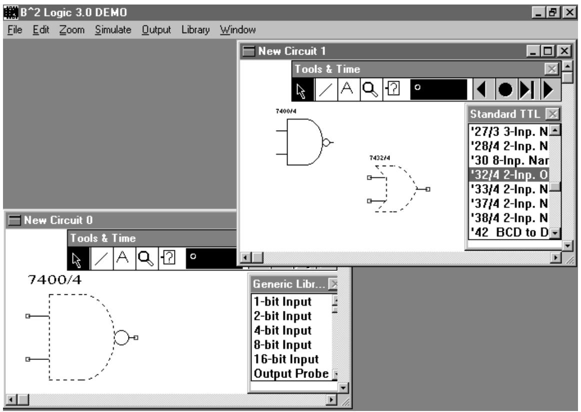
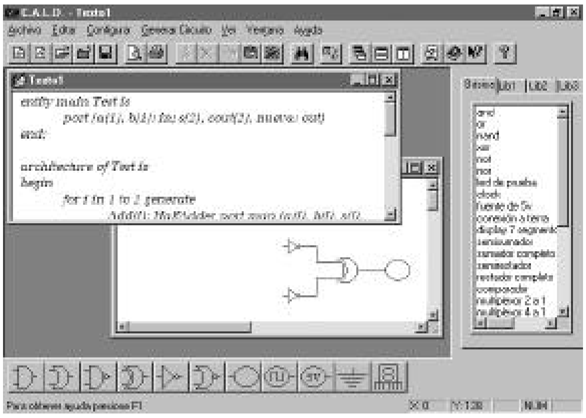


Fig. 1: B^2 Logic, MDI application.

An important feature of MDI application is the interactive edition between several working spaces (windows). This powerful feature has been included in CALD.

B^{2}Logic (fig. 1), provides independent windows with individual floating tools bar and components libraries.

This feature yields a non-negligible space consumption drawback, for this reason it is not included in CALD: only one shared set of bars is available, nevertheless different component libraries can be simultaneously opened (fig. 2).



```
entity main Test is
        port (a(1), b(1): in; s(2), cout(2), nueva: out)
end;

architecture of Test is
begin
        for i in 1 to 2 generate
                Add(i): HalfAdder port map (a(i), b(i), s(i)
```

Fig. 2: CALD, MDI application.

Available help

Help systems included in the tools offer several alternatives. As matter of fact, B^2 Logic lacks of any kind of *help*. CircuitMaker *help* presents a table of contents, search indexes (alphabetical and thematic) and also a "help on help" item. EWB and CALD *helps* basically hold the same structure.

CALD shares all those *help* features.

Tools under review present reduced *help* with respect to graphic interface components and library items. Circuit Maker does not bring information about components on screen but does provide *help* on library items. Nevertheless, a main drawback comes from the fact that information on any library item is only available when this item is inserted in the layout on screen. EWB does not bring information on items on screen but does it on library items.

CALD does bring *help* on both library mode and screen layout mode items.

EDIT FUNCTIONS.

In what concerns usual edit functions the four tools under review share *cut*, *copy*, *paste*, *delete*, *select* and *move* basic operations.

EWB holds a one-level *undo* function while CircuitMaker as well as B^2 Logic the (one-level) *undo* function only allows undoing basic edit functions such as *cut*, *paste*, *delete*, *move*. Only B^2 Logic allows undoing the last created wire. Functions such as rotate, text insert, ... cannot be undone neither in B^2 Logic nor in CircuitMaker.

There is a variety of ways for eliminating elements according to the tool.

In EWB, a wire is deleted through a *point*, *drag* and *drop* operation performed on any wire end while other circuit components are eliminated through *select* and delete key or menu option.

CircuitMaker uses a standard delete operation: screen button then clicking the element or wire to be deleted.

Drawing wires in EWB is achieved through a similar drag and drop procedure. B^2 Logic and CircuitMaker need a previous mode selection then proceed basically in the same way.

CircuitMaker and B^2 Logic allow inserting comment texts in any place, while EWB restricts text to labels and titles.

For drawing clarity purposes, CALD adopted EWB concept for text. Labels versatility provides suitable descriptions of components and wire ends. Any other comment may be inserted off the circuit layout.

Series of useful characteristics, shared by the four tools under review, have been improved in CALD.

In what follows, those features are reviewed.

- **Components labeling.**

CALD moves elements and labels jointly.

Independent movements, such as CircuitMaker does, make the operation more tedious.

Gate labeling in B^2 Logic needs library modifications.

- **Rotating gates.**

As well as any other tool, CALD allows gate rotating. Moreover, as doing so, CALD recalculates optimized wire layout. B^2 Logic neither recalculates nor move connections related to rotated elements. This implies a new connection drawing procedure. EWB does move wires while rotating gates but does not optimize drawing legibility.

- **Font selection.**

CALD allows full font selection (including color). EWB provide standard font while neither B^2 Logic nor CircuitMaker allow color selection.

- **Color selection for wire distinction.**

CALD and EWB share wire coloring feature. Nevertheless, CALD provide user-friendly procedure with extended color set. Other tools under review do not allow wire coloring.

- **Grid selection.**

As well as B^2 Logic and CircuitMaker, CALD provides grid support for layout building. So is doing EWB, but the system imposes discrete locations on screen, according to the grid size. Other tools respect user's choice.

- **Remarks editing.**

Excepting B^2 Logic, other tools provide dedicated space for text editing. CALD provides a system generated information space together with a user defined one.

ROUTING FEATURES

As a matter of fact, one of the central points of this kind of tools is legibility. As far as possible, EWB and CircuitMaker avoid crossing elements but allow it whenever no other solution is available.

B^2 Logic only provides manual drawing facilities, neither automatic routing nor optimization processes are at hand (fig. 3c).

In what concerns wires overlapping EWB avoids it, as far as possible. Whenever components are overloading workspace, the system selects straight crosswise route as default option but keeps avoiding overlapping (fig. 3a).

CircuitMaker does not forbid it; actually, CircuitMaker currently generates overlapping wire section (fig. 3b).

CALD adopted EWB concept but it also allows manual editing. This feature let the user the possibility of moving components and rebuilding wiring (fig. 4).

Tools under review do not optimize wire drawing. They do not minimize wire crossings or avoid overlapping. This can jeopardize the legibility. CALD has been bringing special care to this aspect when designing routing algorithms.

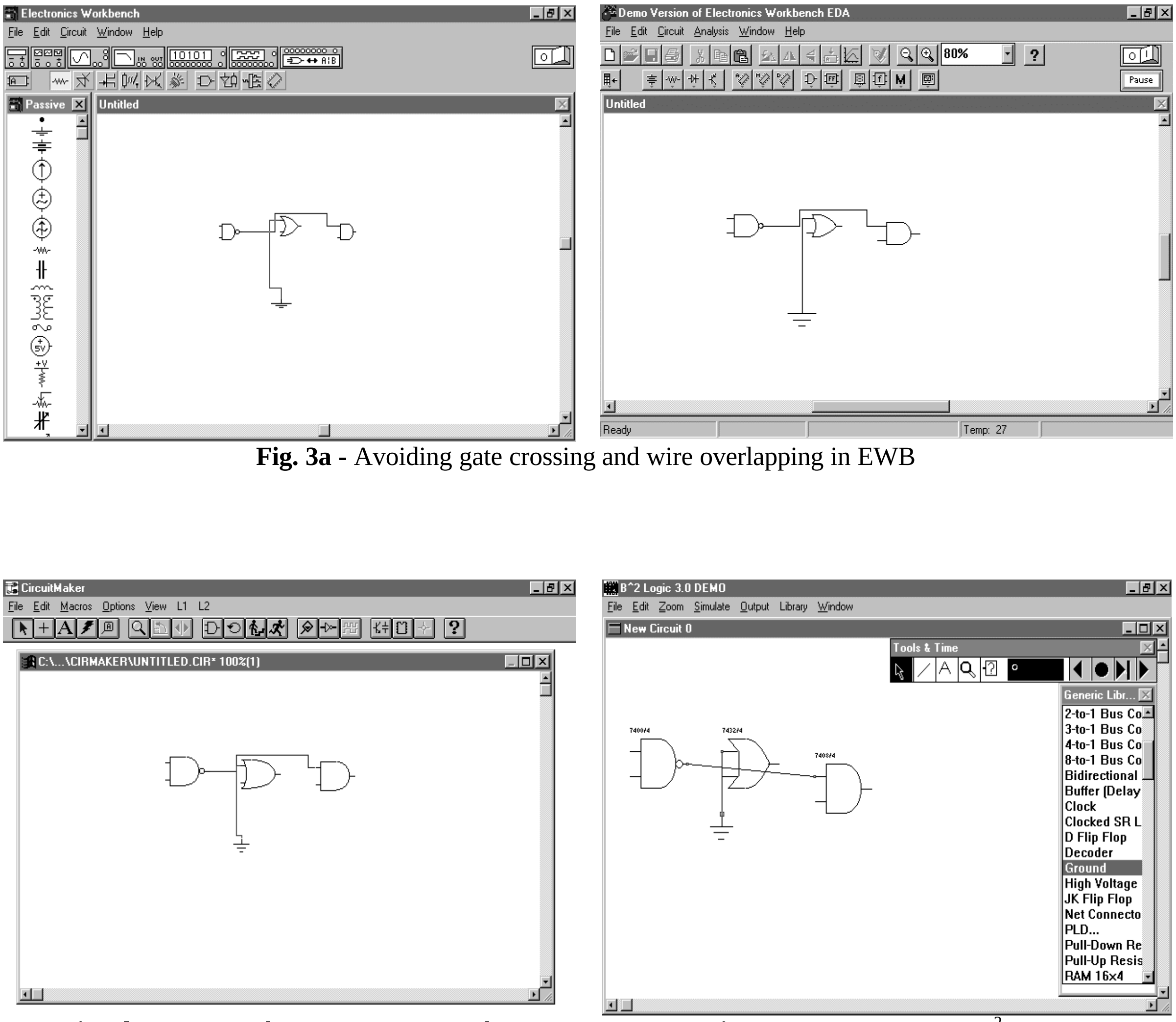
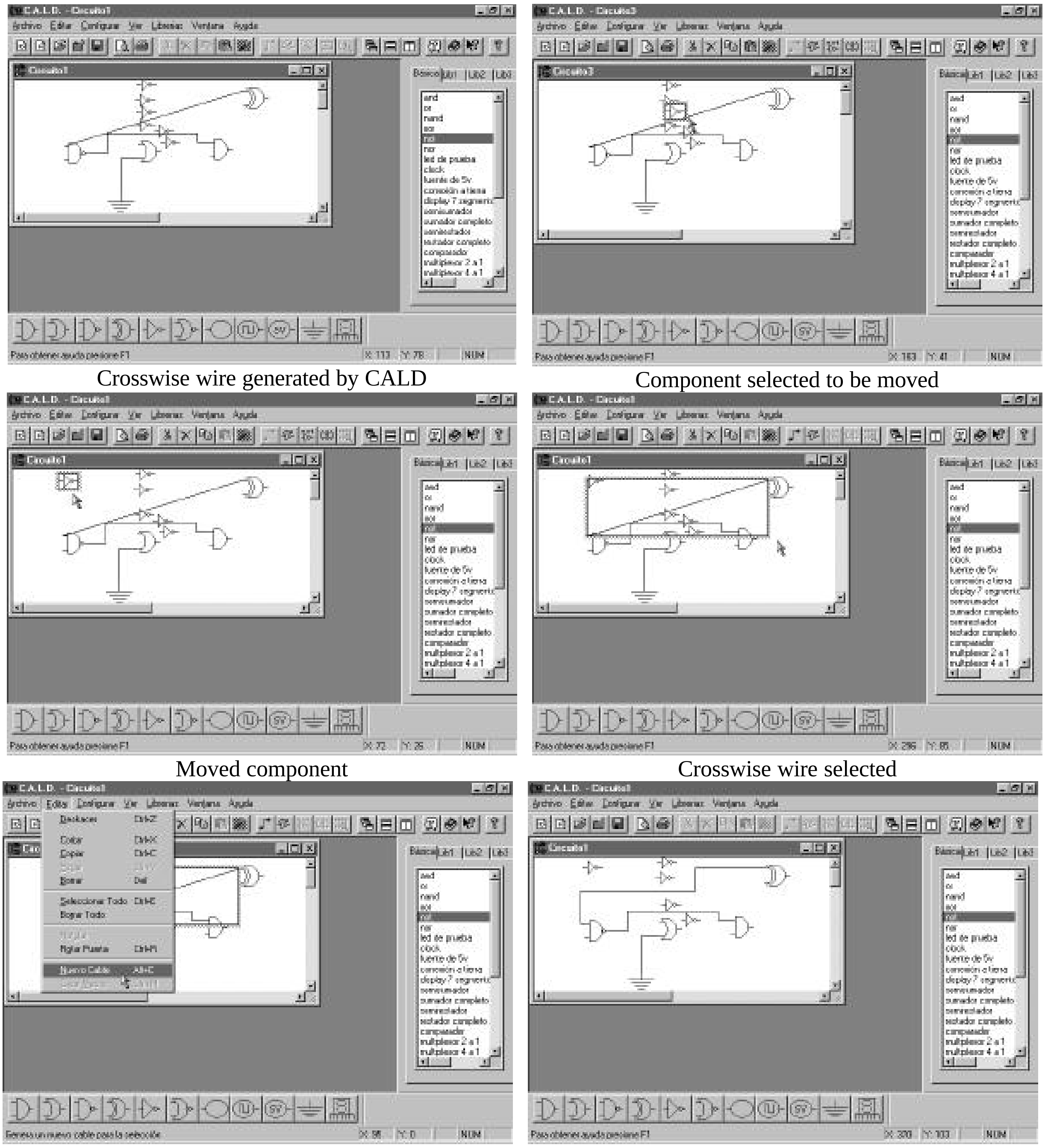


Fig. 3a - Avoiding gate crossing and wire overlapping in EWB

Fig. 3b - Wire overlapping in CircuitMaker

Fig. 3c - Gate crossing in B^2 Logic

Fig. 3 - Routing Features



Crosswise wire generated by CALD

Component selected to be moved

Moved component

Crosswise wire selected

Option: New wire

New wire

Fig. 4: Improving circuit in CALD

CircuitMaker does not make difference between pin-to-pin or pin-to-wire connections. B^2 Logic drawing process, for being exclusively manual, allows any kind of layout connections; this does not prevent inconsistencies such as several outputs connected to a same input, input connected to another input only, unconnected wire end (fig. 5). EWB does demarcate pin-to-wire or wire-to-wire connections.

Actually a contact point element (four-pin dot) has to be dragged from the library and inserted to any wire to be further connected (fig.6 a,b).

CALD allows all connection types; moreover it does not need contact points for wires to be connected. Wire-to-wire linkage is a straightforward process.

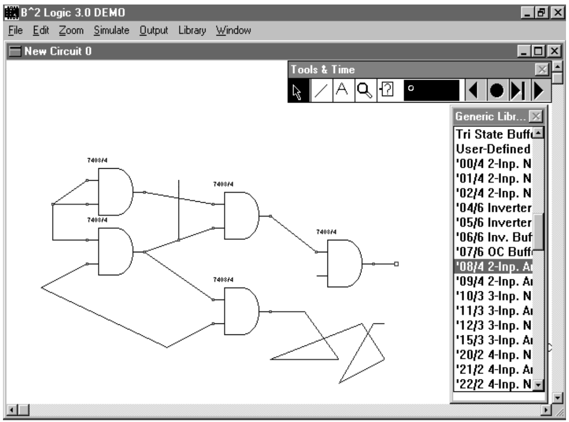


Fig. 5: Wire editing in B^2 Logic

VALIDATION

The designer has to avoid endless wires, redundancies and all physically inconsistent connections: short circuit, critical loop a.s.o..
Deleting wires in B^2 Logic does not eliminate eventual wire links or disconnected elements (fig.6 e,f). EWB automatically cancel wires connected to deleted gates or modules but leaves eventual contact dots in the layout. CircuitMaker strictly eliminates what the designer without validating leftovers, allowing e.g. loose wire fragments (fig. 6 c, d) requests. EWB deals with connection dots as library elements; as so, they can be located anywhere (fig. 6 a,b).

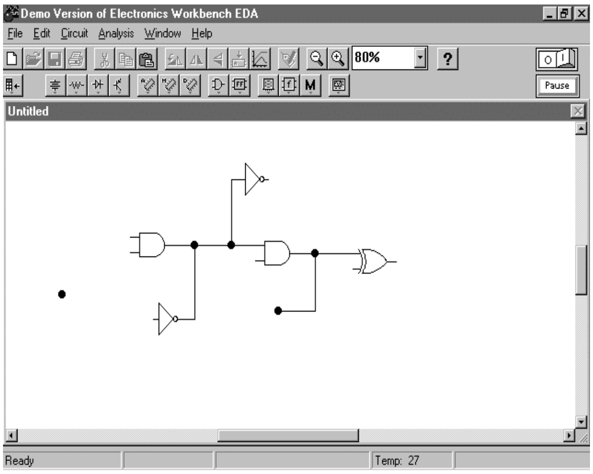

Demo Version of Electronics Workbench EDA
File Edit Circuit Analysis Window Help
80%
?
Pause
Untitled
Ready
Temp: 27

Fig. 6a: Drawing connection dots in EWB

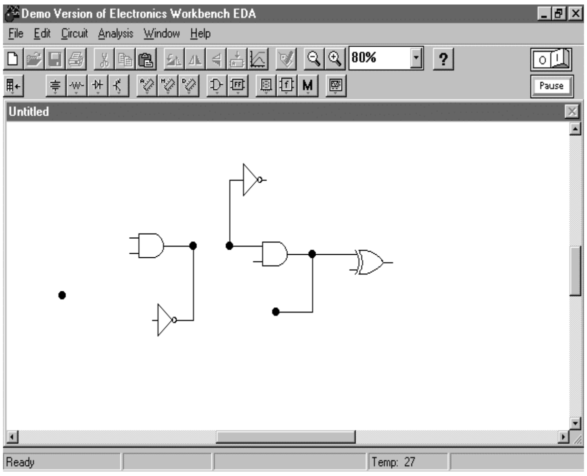

Demo Version of Electronics Workbench EDA
File Edit Circuit Analysis Window Help
80%
?
Pause
Untitled
Ready
Temp: 27

Fig. 6b: Eliminating wire in EWB

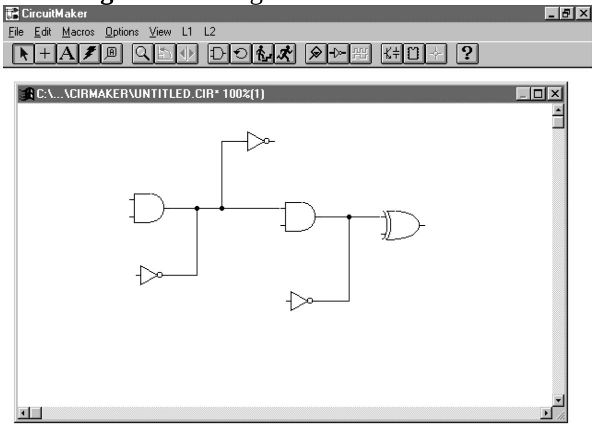

CircuitMaker
File Edit Macros Options View L1 L2
C:\...\CIRMAKER\UNTITLED.CIR* 100%(1)

Fig. 6c: Drawing wire connections in CircuitMaker

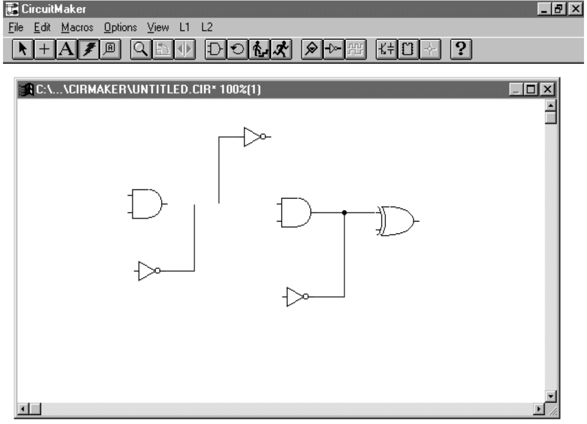

CircuitMaker
File Edit Macros Options View L1 L2
C:\...\CIRMAKER\UNTITLED.CIR* 100%(1)

Fig. 6d: Eliminating wire in CircuitMaker

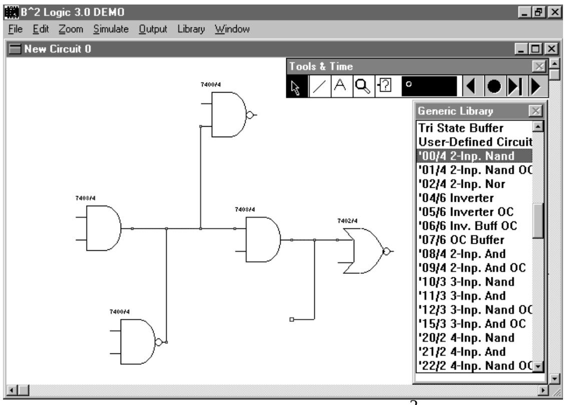


Fig. 6e: Connecting wires in B^2 Logic

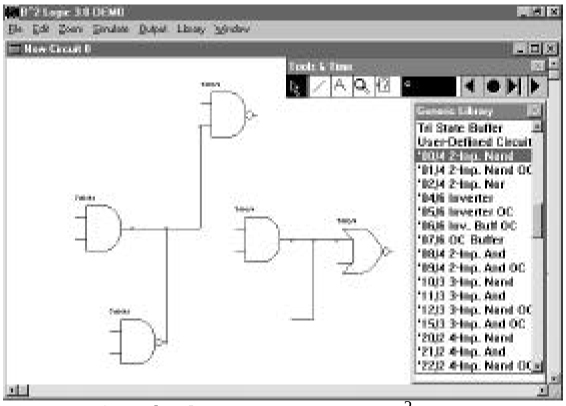
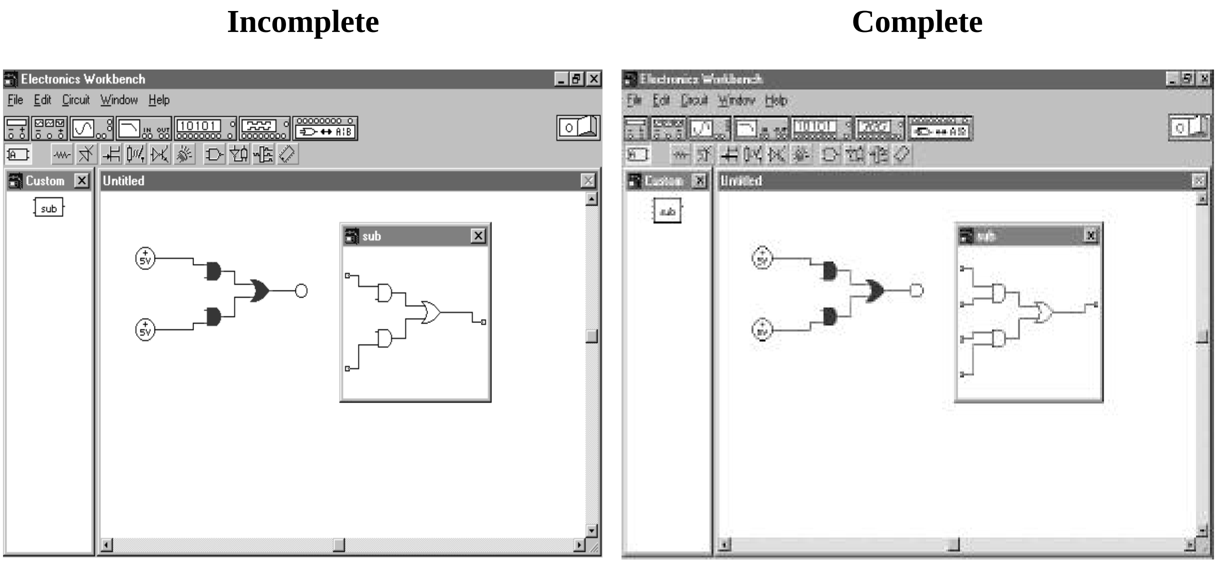


Fig. 6f: Eliminating wire in B^2 Logic

MACROS

EWB as well as B^2 Logic offer handy procedures for creating sub-circuits or Macros, while CircuitMaker is poorer in this respect. Based on EWB concept, CALD environment let the designer locate the external pins of the macro. Moreover CALD demands a full pin set description. Inconsistencies could result from a non-complete pin assignment (fig.7) each free gate output or input has to be related to an external pin.


Incomplete

Complete

Figure 7: Building a macro in EWB

CONCLUSIONS

This paper has reviewed some of the most popular design tools for logical layout design. CALD system, designed by the authors, has been compared to these tools, within an extended set of quality and performance criteria. Details on CALD are available in [HEI99] and [BIO00]. In what follows, synthetic comparison tables are presented as a synthesis of the comparative analysis.◆

SUMMARY INFORMATION

Table I - Comparative features

Tool Feature	EWB	EWB EDA	Circuit Maker	B^2 Logic	CALD
INTERFACE					
Classic look and feel structure	Partial (No tool bar, No state bar)	Yes	No	Partial (No tool bar, No state bar)	Yes
Menu organization	Good	Good	Regular	Good	Good
Items info (“tips”)	No	Partial	Partial	No	Yes
Advanced user shortcuts	Good	Good	Regular	Poor	Good
LIBRARY					
Add new library component	Yes	Yes	No	No	Yes
Easy access	Yes	Yes	Regular	Yes	Yes
Easy localization	Yes	Yes	No	Yes	Yes
Quantity of user libraries	1	1	0	∞	3
Quantity of predefined libraries	10	13	2	3	1
Document-library linkage	Yes	Yes	No	No	No
MDI vs. SDI					
Multiple windows	No	No	No	Yes	Yes
HELP					
Table of contents	Yes	Yes	Yes	No	Yes
Search indexes	Yes	Yes	Yes	No	Yes
Contextual help	Partial (Only document and libraries items)	Partial (Only document and libraries items)	Partial (Only document items)	No	Yes
EDITION					
Copy, paste, cut, delete, move and select	Partial (No available for menu and bar items)	Partial (Only document items)	No	Yes	Yes
Undo	Yes (1 operation)	Yes (1 operation)	Partial (Just a few operations)	Partial (Just a few operations)	Yes (5 operations)

Tool Feature		EWB	EWB EDA	Circuit Maker	B^2 Logic	CALD
Text editing	**Labels**	Yes	Yes	No	No	Yes
	Docu-ment	Yes	Yes	Yes	Yes	Yes
CONNECTION MODES						
Obstacle avoiding	**Gates**	Yes	Yes	Yes	No	Yes
	Wires	Yes (as far as possible)	Yes (as far as possible)	No	No	Yes (as far as possible)
EDITION VALIDATION						
Endless wires allowed		No	No	Yes	Yes	No
Unnecessary connections allowed		Yes	Yes	Yes	No connections	No
MACRO CREATION SUPPORT						
New Macro		Yes	Yes	Yes	No	Yes
Clear Process		Partial (Pin layout)	Partial (Pin layout)	No	-	Yes

BIBLIOGRAFY

[ASH93] Peter J. Ashenden, Henry Detmold and Wayne S. Mc.Keen. “Parallel Execution of VHDL Models”. Department of computer Science, University of Adelaide. Technical Report N°93/01. January 1993.

[BER92] Jean-Michel Bergé, Alain Fonkoua, Serge Maginot and Jacques Rouillard. “VHDL Designer’s Reference”. Kluwer Academic Publishers. Netherlands 1992.

[BIO76] Géry Bioul y J.J. Quisquater. “Application of shortest path problems to design automation of circuit layouts”. Philips Research Reports 31. 423-465. 1976.

[BIO00] Géry Bioul, S. Heineke, V. Otaegui y D. Simonelli, "CALD: una herramienta para la captura de circuitos con ruteo eficiente", VI Workshop Iberchip, Sao Paulo, Brazil, March 16th - 18th, 2000.

[CHE96] Chung-Kuan Cheng, Ting-Ting Y. Lin, Ching-Yen Ho and John Lillis. “New Performance Driven Routing Techniques with Explicit Area/Delay Tradeoff and Simulltaneous Wire Sizing”. DAC 96.

[DAV83] M. Davio, J.-P. Deschamps and A. Thayse. “Digital Systems with Algorithm Implementation”. John Wiley & Sons. Brussells 1983.

[DRE96] Rolf Drechsler, Nicole Göckel, Gregor Pudelko and Bernd Becker. “A Hybrid Genetic Algorithm for the Channel Routing Problem”. Internat. Symposium on Circuits and Systems (ISCAS). Atlanta, May 1996.

[ESA95] Peter Sinander. “The Usage of VHDL in the European Space Agency”. European Space Research and Technology Centre. European Space Agency 1995.

[HEI99] Susana Heineke y Valeria Otaegui. "Computer Aided Logic Design", Tesis de Grado. UNCPBA, 1999.

[LEE61] C. Y. Lee. “An Algorithm for Path Connections and Its Applications”. IRE Transactions on Electronic Computers. September 1961.

[LIE96] Jens Lienig. “A Parallel Genetic Algorithm for two detailed Routing Problems”. Integration, the VLSI Journal. March 1996.

[OH95] Jaewon Oh, Iksoo Pyo and Massoud Pedram. “Constructing Minimal Spanning/Steiner Trees with Bounded Path Length”. ISCAS 95.

[RUB74] Frank Rubin. “The Lee Path Connection Algorithm”. IEEE Transactions on Computers, Vol. c-23, N°9. September 1974.

[SHE96] Charles Shelor. “VHDL Designer”. Revista de Internet VHDL Times N°1

[SIN94] P. Sinander. “VHDL Modeling Guidelines”. ESA. 1994.

[VHD93] “IEEE Standard VHDL Languaje Reference Manual”, IEEE Std 1076-93, New York, USA, 1994.

[WOL96] Hans G. Wolf and Dieter A. Mlynski. “A New Genetic Single-Layer Routing Algorithm for Analog Transistor Arrays”. Institut fuer Theoretische Elektrotechnik und Messtechnik Universitaet Karlsruhe. Germany. 1996.