Digital Breadboard

v1.1.9

Danny L. Griffin

Abstract

Digital Breadboard is a full GUI sequential-solving digital logic circuit simulator that will allow you to design, build, and test digital logic circuits in software.

1. Preface

1.1. Introduction

Ever since the Amiga's introduction, there has been a steadily increasing number of scientific applications available, but one application that many people have been asking for that is not currently available (that I know of) is a digital logic circuit simulator. So, rather than just talk about it, I decided to write one.

1.2. What it is

Digital Breadboard is a full GUI sequential-solving digital logic circuit simulator that will allow you to design, build, and test digital logic circuits in software. In this release, there are some limitations put on the type of circuits that you can build, the primary one being the size of the circuit. No more than forty elements are allowed in this version.

Still, this release does incorporate many useful tools, allowing Digital Breadboard some small measure of power. Among these are support for:

- multiple input AND, OR, NAND, and NOR gates
- XOR and NOT gates
- D, JK, and SR flip flops
- multiple user-definable clocks
- switched and pulsed inputs
- event counter
- outputs
- independant four-channel trace scope
- preferences printing

Circuits can, of course, be saved, recalled, and edited at any time.

2. Getting Started

2.1. Requirements

Digital Breadboard requires:

- AmigaOS 2.04 (V37 or higher)
- reqtools.library (V37 or higher)

...that's it!

2.2. Installation

If you don't already have `reqtools.library` in your `libs:`, move it there. You may wish to install the hpcalc font in your fonts: directory:

```
copy fonts fonts:  all
```

However, this is not necessary. Since hpcalc is not a full font, some people may not want to install it in fonts:. DBB will first look in `[DBBdir]`[1]`fonts` for the hpcalc font, then in `fonts:`. If Digital Breadboard doesn't find the font it will still run, but the **Timer** window will use Topaz 8 and it will look goofy. This font will also be used for the 7-segment display element in the next rev. This is the order in which Digital Breadboard looks for files:

Fonts - [DBBdir]fonts, fonts:
DBB.prefs - [DBBdir]
DBB.guide - [DBBdir]
Design files - [DBBdir]Circuits, [DBBdir]

All icons are saved with a fully qualified Tool path.

2.3. Start

Digital Breadboard can be started from the CLI by typing `DBB` or by double-clicking on the icon. Digital Breadboard will attempt to open up an overscan interlaced hires screen (682 x 440 or greater) while taking into account the user preferences, and two windows will initially appear: the design window and the tools (Elements) window. However, some people don't use an overscan display, and so for both of you Digital Breadboard will limit the horizontal screen size to match the user preferences rather than have the screen autoscroll left every time the mouse is moved to the right edge. The Elements window will overlap the design window. This, however, will be fixed when a resizeable window is implemented to support the larger virtual work area in the next revision. PAL screens are supported.

Digital Breadboard accepts a single command-line argument—the name of a design file. If given, Digital Breadboard will load the file upon startup. All other arguments will be ignored.

To begin placing objects, simply click on the desired element with the mouse, move your pointer to where you wish to place the element, and click the left mouse button to drop it. Then, click on another element and do the same.

When you're ready to connect the elements, choose the menu option EDIT/CONNECT. The design window's titlebar will change to read:

Connect - Select output element

Using the mouse, click on the output element to start with. When selected, a highlight box will appear around the element. You do not have to click on a specific output. If an element has more than one output, as in a D flip flop, a requester will appear asking you which one you wish to use. Simply select the appropriate response (see Section 4.1). If you

[1] We will use the symbol [DBBdir] throughout this manual to represent the directory in which the executable program `DBB` resides. Digital Breadboard will produce a fully qualified path for this location whether it was started from the Workbench or CLI, even if it is not the current directory (DBB may be 'path'ed, for example), e.g. `[DBBdir]Circuits` might represent `DF0:Circuits` as well as `DH4:Breadboard/Circuits`. The user does not have to do any `assign`'s.

change your mind, press ESC and the requester will go away and you can select a different output element.

After you've selected an output, you need to select an input. The window's titlebar will again prompt you.

Connect - Select input element

Again, you do not need to click on a specific input, just the element. An element with multiple inputs will respond with a requester asking you which input you wish to use. Select the appropriate response, or press ESC to cancel.

When you have completed wiring your design, you may save it to disk with PROJECT/SAVE or begin simulation with CONTROL/RUN. You may at any time return to any of the editing screens (Add, Connect, Cut, Delete, etc.)

To quit Digital Breadboard, select the menu option PROJECT/QUIT or press **RAmiga-Q**. If your work isn't saved, you will be asked if you wish to save your work before exiting.

These are the basics of operation, but there are many more features that are described in the following pages. Try loading one of the sample project files.

3. Menus

3.1. Project

New

Start a new design. This will replace the current design, and if you try to overwrite a modified design, a security requester will pop up giving you a chance to save the current file before proceeding.

Open

Open an existing Digital Breadboard design file. A filerequester will appear in which you select the new design file to load. This file will replace the current one, and if you try to overwrite a modified design, a security requester will pop up giving you a chance to save the current file before proceeding.

Save

Save the current circuit under the current name. If there is no current filename, a filerequester will appear.

Save As

Save the current circuit under a new name. A filerequester will appear.

Print

Print the circuit using the preferences printer. Digital Breadboard will attempt to determine the maximum resolution of your printer, as selected in **Preferences**, and print the design circuit currently displayed. Due to the ability of the user to select his own screen colors, and to keep as much control in the

hands of the user as possible, Digital Breadboard provides control of the printer from within the program. This will not modify the user PrinterGfx Preferences settings. It is normally a good idea to print with 'Black & White' shading, a reasonable threshold, and the correct Image settings. For the default WB2.x colors (grey, black, white, and blue), a *Positive* image with a threshold of 7 is fine. *See Section 3.4, the Settings menu.*

About

Display some useful (?) information.

Quit

Quit

3.2. Edit

Delete Device

Remove digital logic elements from the circuit. The window titlebar will prompt you with:

Delete - Select element to remove

Click on the element to remove from the circuit, and it, along with any associated connections, will be removed. Press `ESC` to cancel.

Cut Connection

Cut a connection between two elements. Use the mouse to click on the first (output) element, then the second (input) element. Again, Digital Breadboard responds with positive feedback by highlighting the first element with a box. The connection between the two elements will be removed. Press `ESC` to cancel.

Move Device

Move an element. Select an element to move to a new location by clicking on the element, and then clicking on a new (empty) location. All associated connections will be rerouted as well. Press `ESC` to cancel.

Redraw Screen

Redraw screen. Sometimes, after adding elements to a previously routed design, the wiring as displayed on the screen may need to be redrawn. This is because Digital Breadboard attempts to route the connections as directly as possible, without causing 'conflicts' (wires colinearly overlapping by small amounts). Selecting this will refresh the screen and reroute the design, thereby eliminating any drawing problems. Note that this is purely asthetic—all connections are unaffected by the screen display. Any time that the screen is redrawn, whether due to circuit changes or by user command, all probes are disconnected.

Add Legend

Edit and place a legend in the design window. A requester will pop up giving you a chance to attach a legend window to your design. You may enter a title

(circuit name), your name, the date, and revision of the design. If present, it will also be printed if you elect to print your circuit. If there is an existing legend, the information will appear in the gadgets in the requester. Close the window, then place the window frame that follows your mouse on the screen. Press `ESC` to cancel.

Remove Legend

Remove the legend from the window. All information is still associated with the file. This is usually done if you are editing an existing design or if you don't want the legend to be printed.

3.3. Control

Add

Add digital logic elements to the circuit. Select a logic element with the mouse and place it on the screen, as described above.

Connect

Connect elements together. Select the first (output) element (a highlight box will surround the selected element for a more positive feedback as described above), and then the second (input) element. A connection will be drawn between them, according to the current routing method. *See Section 3.4, the Settings menu.*

Run/Stop

Begin solving logic. Digital Breadboard will begin circuit simulation at the current selected speed. To exit RUN mode, select CONTROL/STOP or press `ESC`. *See Section 4.2, the Tool Window: Up Arrow, Down Arrow, T, t.*

3.4. Settings

Smooth

Elements will follow the mouse pointer smoothly.

Snap

Elements will snap to allowable placement positions (default).

Direct Path

Connects adjacent elements by directly routing inputs (default). This produces a more visually pleasing, and easier to follow, display. In certain element placements, an output of one element may overlap (colinearly) the input of another element by a small amount. If this condition exists, Digital Breadboard will detect it and will automatically route the input via a longer path to avoid this confusion. However, if elements are added after this initial routing is done, it is possible to still have an output overlap an input. All that is needed to do if this should occur is to refresh the screen (EDIT/SCREEN REDRAW) and the problem will be taken care of. It might be a good idea to do this as a matter of course before printing a design if it is particularly complicated and you are not sure if this has occured. It is not necessary to ever refresh the screen before saving a design, and all designs loaded from disk will be routed correctly.

Long Path

Connects adjacent elements by routing below the inputs. Will never cause wire routing conflicts, but is not particularly pleasing to the eye. Why is this even offered as an option? I dunno. See *Direct Path* above.

Save With Icon

If selected, all files will be saved with a Project icon, with a fully qualified Tool path. Select SAVE SETTINGS to save.

Palette

Allows user to define Digital Breadboard's color palette. Select SAVE SETTINGS to save.

Printer Prefs

Allows user direct control over Digital Breadboard's printer output without the user having to modify his user preferences. Control of *Image, Aspect, Shade, Threshold, Smoothing,* and *Centering* behave just as in the PrinterGfx Preferences settings. Select SAVE SETTINGS to save.

Save Settings

Saves current color Palette and Printer Prefs to `[DBBdir]dbb.prefs`. If dbb.prefs is found upon startup, its settings will be used, else DBB will use default settings.

4. Tool Window

Fig. 1: Elements Toolbox

4.1. Elements

AND, OR, NAND, NOR, NOT, XOR

Two- or three-input gates (except XOR and NOT). Keyboard equivalents: A, O, N, R, ! or 1, X.

D

Positive edge-triggered flip flop. The upper left input is D, the lower left input is the CLK, the upper right output is Q, and the lower right output is NOT Q. Keyboard equivalent: D.

JK

Negative edge-triggered JK flip flop. The upper left input is J, the middle left input is the CLK, the lower left input is K, the upper right output is Q, and the lower right output is NOT Q. Keyboard equivalent: J.

SR

Negative edge-triggered SR flip flop. The upper left input is S, the middle left input is the CLK, the lower left input is R, the upper right output is Q, and the lower right output is NOT Q. Keyboard equivalent: S.

CLK

Independant, user-adjustable clock. The initial **Delay**, **Width**, and **Period** are adjustable. The default values are 20, 10, and 100 respectively. The attributes may be changed after placement by selecting the Info button. Keyboard equivalent: K. *See Info button.*

IN

Switched (default) or pulsed input. The input is activated in RUN mode by clicking on it with the left mouse button. The pulsed input has a user-adjustable pulse width. Keyboard equivalent: I. *See Info button.*

OUT

Output status indicator. The output element 'lights up' when *TRUE*, 1, or high. The color is user-adjustable. Keyboard equivalent: U. *See Info button.*

+5

Vcc, +5 volts, high. You can use this element to specifically tie inputs high. Active only in CONNECT mode. *Note: all unconnected inputs are considered* ***LOW***. This is considered the *input* element when cutting connections. Keyboard equivalent: 5.

GRD

Ground, 0 volts, low. You can use this element to specifically tie inputs low. Active only in CONNECT mode. *Note: all unconnected inputs are considered* ***LOW***. This is considered the *input* element when cutting connections. Keyboard equivalents: G or 0.

4.2. Buttons

Info

This is a multi-purpose button who's actions are modified depending on what element is next selected in the design window. To use, click on the Info button, then click on an element that you have placed in your circuit. Keyboard equivalent: F. Valid elements are:

IN

A requester will pop up allowing you to select a switched (default) or pulsed input. Switched inputs are toggled on/off by clicking the mouse on them while in RUN mode. A pulsed input will stay high for the duration of the pulse width. The default is 20 clock cycles. An input's parameters may be changed at any time, even while Digital Breadboard is in RUN mode solving logic.

OUT

At this time the only user-selectable output element options are the color. Click on the desired color button. An output's parameters may be changed at any time, even while Digital Breadboard is in RUN mode solving logic.

CLK

A requester will pop up allowing you to specify the clock's pulse **Width** (on-time), **Period**, and initial **Delay**. A sample waveform accompanies and responds to the settings. Click the

requester's close gadget when done. A clock's parameters may be changed at any time, even while Digital Breadboard is in RUN mode solving logic.

2, 3, 4, 8

These buttons determine the number of inputs for AND, OR, NAND, and NOR gates. The default is 2. To change this, simply select another button, then select the element to place. *Note: Digital Breadboard v1.1.x only supports 2- and 3-input gates.* Keyboard equivalents: 2, 3, 4, 8.

Count

This button brings up an event counter that you may connect to any output point (or the input of an OUT element) in the circuit using alpha characters. You may monitor to four points simultaneously. The counter value increments on *FALSE* to *TRUE* transitions. If used in a circuit, the counts continue to accumulate whether or not the counter is displayed. To attach a counter, click on one of the gadgets, then click on the element to monitor. You may move the counter at any time you wish; the count will be reset to zero. This is of very limited utility and will probably be eliminated in the next version for something more useful. (Real counters will appear as elements in the next version of DBB). Keyboard equivalent: C.

Scope

A four-channel trace scope. Connect each channel independantly to any output point of the circuit using numeric labels. Each channel may be turned on or off independantly. The scope window may be dragged to any convenient spot on the display. The scope 'probes' may only be connected in the ADD mode. Click the button of the channel to connect to the circuit, then click on the element you wish to monitor the output of. A number corresponding to the scope's channel will appear above the output it is connected to. You may move these 'probes' at any time by repeating this process. There is also a button on the scope panel marked **Crsr**. This toggles a (scope) screen cursor on and off. Click the **Scope** button again to close the scope window. Keyboard equivalent: P.

Down arrow

Slow down the solving of logic. Every time you click on the down arrow, the solving of the logic will get slower. This is useful for small circuits or fast Amigas. Keyboard equivalent: down arrow.

Up arrow

Speed up the solving of logic. Every time you click on the up arrow, the speed at which logic is solved will increase. The initial speed is maximum. Keyboard equivalent: up arrow.

T

Toggle the system's clock (**Time**) display. When this button is selected, a window will pop up and you will see time furiously passing away. To speed

up or slow down *time* (the speed at which the logic is solved), use the up or down arrow buttons. Keyboard equivalent: T.

t

Toggle the element status indicators (**throb**). Each element has an indicator(s) which will appear adjacent to an output when that function or output is *TRUE*, e.g. all inputs must be *TRUE* for an **AND** gate and at least one input must be *FALSE* for a **NAND** gate's output to be *TRUE*. A status indicator will appear next to a flip flop's *TRUE* output. The **CLK** element's status indicator will throb once for each clock pulse. Keyboard equivalent: t.

5. Comments on Good Design

It is generally not a good idea to design circuits with static or dynamic hazards or critical races in them. And in fact, Digital Breadboard will not check for these, nor will it solve logic in a time-dependant fashion. Let's take a look at the following example:

Fig. 2: Static hazard

As you can see, output Y will momentarily go to a 0 when input A goes from a 1 to a 0 if both input B and output Y are initially 1. This output "glitch" is called a static hazard and can be avoided with the addition of a consensus term.

Fig. 3 shows an improved version. This addition of an **AND** gate will remove the hazard and will produce a stable result.

Fig. 3: Hazard-free

The use of Karnaugh maps (K-maps) greatly aid in the design of digital logic. The first figure below is a K-map of the first circuit. The second shows the addition of the consensus

term added to avoid the critical race. For more information about hazards, races, or other design-related topics, consult a text on digital logic design.[2]

AB \ Y	0	1
0 0	0	1
0 1	0	1
1 1	1	1
1 0	0	0

$Y = AB + \bar{A}y$

AB \ Y	0	1
0 0	0	1
0 1	0	1
1 1	1	1
1 0	0	0

$Y = AB + By + \bar{A}y$

6. Future Improvements

Planned enhancements (Spring 1994) include:

- larger workspace
- improved user-interface
- resizable window
- more elements
- user-definable elements
- text support for labels, comments, etc.
- design by truth table
- basic structured drawing tools (with selectable line types)
- propagation delays
- 8 channel scope, trace store and print ability
- sticky probes
- Workbench option
- save as IFF
- selectable screen modes and 2024 support
- European, IEEE logic symbol support
- AREXX port

Future:

- better printer output including CompuGraphic font support

[2] Digital Design Fundamentals, Breeding, Prentice Hall, is one example detailing these and other design considerations.

7. Bugs

All known bugs, enforcer hits, and mungwall errors have been eliminated. If you encounter an error, please report it, along with a small sample circuit in which the error is reproduceable if possible.

8. Comments, suggestions, and other feedback

I have heard of the availablility of circuit simulators for other computers, and apparently the Amiga once upon a time had a commercial simulator available for it, but I've yet to see any of these. Consequently, there are probably many things I could have done better, or at least differently. I'm certainly open to suggestions.

Please send all bug reports, comments, and other feedback to Dan Griffin at:

griffin@egr.msu.edu

or via snail mail to:

Dan Griffin
2049 Tamarack Dr.
Okemos, MI 48864

9. Distribution

There have been reports of people uploading only the executable to various sites, particularly in Europe. DON'T DO THIS! Please keep all of the files in this distribution together. See the README file for a complete list.

The latest version of Digital Breadboard is available:

all aminet ftp sites (ftp.wustl.edu)
/pub/aminet/util/misc
dbbxxx.lha (where xxx is the version number)

Fireline BBS
+1-517-374-8900
File Area: Misc Files

Directly from the author
Please send $3 to cover postage, $5 if you also want a laser-printed TEX manual (please specify plain or 3-hole paper)

10. Complaints

All complaints will be forwarded to BLAZEMONGER INCORPORATED'S "Customer Service" Department.

11. Disclaimer and Copyright

Digital Breadboard is freeware. The question of this software's suitability for any particular purpose is left as an exercise for the reader. Source code is not available for version 1.1.x. So there.

A. mdpic

mdpic is a combinational logic design utility that will produce a minimal covering equation in SOP (sum of products) form. It is particularly handy in circuits with 6 or more inputs where solving problems by hand becomes tedious. It is not a GUI program and must be run in a shell. To run mdpic, simply type `mdpic` in a shell. You will be asked to enter the number of variables, the minterms [3] for the equation, and then the don't-cares. Following this, a minimal covering equation, i.e. the essential prime implicants, will be supplied in Sum of Products (SOP) form.

For example, suppose you wanted to know the equation for the minterms 1, 2, 4, 7, 9, 10, 15, and 27 of a five variable design. Following is a sample session to solve this problem.

```
Combinational Logic Design

How many variables?  5

Enter the minterms (whitespace delimiters, period to end):  1 2 4 7 9 10
15 27 .

Enter the don't-cares (whitespace delimiters, period to end):  .

Answer:  a' c' d' e + a' c' d e' + a' b' c d' e' + a' c d e + a b c' d e
```

The function is realized by the above equation, where a', b', etc. indicate negated inputs.

A.1. Options

–v

The verbose option will cause all of the prime implicants to be printed. This information can be used when designing static hazard-free circuits. The extra terms produced in this step can be used as consensus terms to eliminate static hazards. Although it will often (usually) be the case that this is a minimal design, there is no *guarantee* that the equation will not contain redundant terms. For example:

$$\sum \text{m } (2\ 3\ 4\ 6\ 7\ 11\ 12\ 13\ 14\ 15)$$

produces a prime implicant *bc* (6 7 14 15) that becomes redundant. It just so happens in this case that the essential prime implicants also produce a static hazard-free design, even though there are other prime implicants.

mdpic's capabilities will be expanded in the future to produce guaranteed minimal static hazard-free equations.

[3] Minterms are the input conditions for which you wish the function to be true. For example, if you have a six input circuit for which you wish an output to be true when: a) all the inputs are false, b) inputs d, e, and f are true, c) inputs a, b, c, and e are true, the corresponding minterms would be 0, 7, and 58. Just add up the binary weight of the inputs.

B. rtk

rtk solves 4 variable Karnaugh maps (K-maps) interactively. When started by typing rtk in a shell or by double-clicking its icon, the rtk window pops up. The user then clicks on any of the 16 buttons corresponding to a minterm in a 4-variable K-map. Minterms are indicated by a 1, and don't-cares by a −. rtk will update the equation realized by the terms in the upper part of the window. rtk produces a minimal covering SOP equation. rtk requires the supplied XEN 9 point font.

C. Truth tables

AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NAND

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

XOR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

D

D	Clk	Q
0	↑	0
1	↑	1

JK

J	K	Clk	Q
0	0	↓	q
0	1	↓	0
1	0	↓	1
1	1	↓	$\bar{q}$

SR

S	R	Clk	Q
0	0	↓	q
0	1	↓	0
1	0	↓	1
1	1	↓	-

D. Keyboard equivalents

D.1. Element window

A,a AND gate

C,c Counter (event counter)

D,d D flip flop

F,f Info

I,i INput element

J,j JK flip flop

K,k CLK

N,n NAND gate

O,o OR gate

P,p Scope

R,r NOR gate

S,s SR flip flop

T Timer window

t throb toggle

U,u OUTput element

X,x XOR element

ESC quit RUN, CONNECT, DELETE, CUT, MOVE, and probe connect modes

!,1 NOT gate

2 2 input gates

3 3 input gates

4 4 input gates

8 8 input gates

↑ speed up logic solving

↓ slow down logic solving

0,G GROUND, tied low, 0 volts

5 Vcc, tied hi, +5 volts

D.2. Amiga keys (menus)

RAmiga-N New

RAmiga-O Open

RAmiga-S Save

RAmiga-W Save As

RAmiga-P Print

RAmiga-? About

RAmiga-Q Quit

RAmiga-D Delete Device

RAmiga-X Cut Connection

RAmiga-M Move Device

RAmiga-L Redraw Screen

RAmiga-A Add

RAmiga-C Connect

RAmiga-R Run

RAmiga-T Stop

RAmiga-Y Smooth

RAmiga-Z Snap

D.3. Other keys

Help activate AmigaGuide online database