

Biddy

1.8.2

Generated by Doxygen 1.8.13

Mon Feb 11 2019 14:10:13

Contents

1	USER MANUAL	2
2	Data Structure Documentation	13
2.1	Biddy_Boolean Class Reference	13
2.2	Biddy_Cache Class Reference	13
2.2.1	Detailed Description	13
2.3	Biddy_Edge Class Reference	13
2.3.1	Detailed Description	14
2.4	Biddy_GCFunction Class Reference	14
2.5	Biddy_LookupFunction Class Reference	14
2.6	Biddy_Manager Class Reference	14
2.6.1	Detailed Description	14
2.7	Biddy_String Class Reference	15
2.8	Biddy_Variable Class Reference	15
3	File Documentation	15
3.1	biddy.h File Reference	15
3.1.1	Detailed Description	29
3.1.2	Macro Definition Documentation	30
3.1.3	Function Documentation	68
3.2	biddyInOut.c File Reference	137
3.2.1	Detailed Description	138
3.2.2	Function Documentation	139
3.2.3	Variable Documentation	144
3.3	biddyInt.h File Reference	145
3.3.1	Detailed Description	147
3.4	biddyMain.c File Reference	148
3.4.1	Detailed Description	152
3.4.2	Function Documentation	153
3.5	biddyOp.c File Reference	182
3.5.1	Detailed Description	184
3.5.2	Function Documentation	185
3.6	biddyStat.c File Reference	202
3.6.1	Detailed Description	204
3.6.2	Function Documentation	205

1 USER MANUAL

TL;DR

Biddy is a multi-platform academic Binary Decision Diagrams package. It supports plain ROBDDs, ROBDDs with complemented edges, 0-sup-BDDs, 0-sup-BDDs with complemented edges, and tagged 0-sup-BDDs.

Biddy is capable of all the typical operations regarding Boolean functions, combination sets, and BDDs.

Biddy is a library to be included in your C and C++ projects:

```
#include "/path/to/biddy.h"
```

To **compile** Biddy library use "make static" or "make dynamic" or "make debug". There is no configuration script, you should edit **Makefiles** to adapt the system configuration. Alternatively, use the prepared **Visual Studio project** (VS/Biddy.sln).

```
biddy> make static  
biddy> make clean
```

Dependencies (tested on Ubuntu and Windows 10):

- on GNU/Linux, you need libgmp (<https://gmplib.org/>)
- on MS Windows, you need MPIR library (<http://mpir.org/>)

When using Biddy on GNU/Linux, you may have to tell bash about the library:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/absolute/path/to/biddy/library
```

Tests:

- create a static Biddy library
- compile the Hanoi example (notes for building are given in the source code)
- alternatively, compile any of the other example

```
biddy> make static  
biddy> gcc -DUNIX -DBIDDY -o hanoi biddy-example-hanoi.c -I. -L./bin -static -lbiddy -lgmp  
biddy> ./hanoi 10
```

There are two **additional packages** included into Biddy distribution:

- **bddview** is a pure Tcl/Tk script for visualization of BDDs,
- **BDD Scout** is a demo application demonstrating the capability of Biddy and bddview.

To create BDD Scout (please, check its dependencies):

```
biddy> make static  
biddy> cd bddscout  
bddscout> make
```

Biddy is **free software** maintained by Robert Meolic (robert@meolic.com).

- You can fork this project on <https://github.com/meolic/biddy>.
- Fell free to report requests, bugs, and other issues on GitHub.
- **Contributions** (patches and ideas) can be send also via author's email.

Homepage: <http://biddy.meolic.com/>

1. AN OVERVIEW

Biddy supports ROBDDs as described in "K. S. Brace, R. L. Rudell, R. E. Bryant. Efficient Implementation of a BDD Package. 27th ACM/IEEE DAC, pages 40-45, 1990".

Biddy supports 0-sup-BDDs (also denoted ZBDDs or ZDDs) as described in "S. Minato. Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, 30th ACM/IEEE DAC, pages 272-277, 1993".

Biddy supports tagged 0-sup-BDDs (also denoted TZDDs or TZBDDs) as introduced in "R. Meolic. Implementation aspects of a BDD package supporting general decision diagrams. <https://dk.um.si/lzpisGradiva.php?id=68831>" and described in "T. van Dijk, R. Wille, R. Meolic. Tagged BDDs: Combining Reduction Rules from Different Decision Diagram Types, 17th FMCAD, pages 108-115, 2017".

Biddy includes:

- automatic garbage collection with a system age (a variant of a mark-and-sweep approach),
- node management through formulae protecting,
- variable swapping and sifting algorithm for all supported BDD types.

Biddy is optimized for efficiency, but it is mainly oriented towards readable and comprehensible source code in C.

Biddy is currently used in the following projects:

- BDD Scout, demo project which allows visualization of BDDs and also includes some benchmarks
- Efficient Symbolic Tools (EST), model checking and other algorithms for formal verification of systems (<http://est.meolic.com/>)

2. SOURCE CODE

Biddy is free software released under GPL.

The short name of Biddy package is 'biddy'. This name is placed in front of all filenames and external identifiers. It may appear in all lowercase, or with its first letter capitalized, written as 'biddy' and 'Biddy', respectively.

There are three categories of C functions.

- Exported functions are visible outside the package.
- Internal functions are visible to all files within the package.
- Static functions are visible to the file only.

There are two types of C functions.

- General functions which operates on a particular BDD and considering only graph properties (i.e. changing edge's mark, selecting nodes, counting nodes etc.). These functions are the same for different type of decision diagrams (BDD, ZDD, etc.). Functions, which add or delete nodes or those which needs info about variables (e.g. a name or a value) are not general functions. Exported general functions have prefix Biddy_.

- Managed functions, which operates on a global properties of a BDD system (e.g. node table, variable table, formula table, various caches, etc.) or consider a BDD as a Boolean function (e.g. Boolean operations, counting minterms, etc.). These functions need info stored in a manager. Exported managed functions have prefix `Biddy_Managed_`.

Biddy consists of the following core files:

- [README.md](#) (this file)
- `paper.md` (JOSS paper)
- `CHANGES` (history of changes)
- `COPYING` (license file)
- `VERSION` (project's version)
- `Makefile` (used to produce binary code)
- `Makefile.Linux` (Makefile definitions for GNU/Linux)
- `Makefile.MINGW` (Makefile definitions for MS Windows)
- `Makefile.Darwin` (Makefile definitions for MacOS)
- [biddy.h](#) (header)
- [biddyInt.h](#) (header)
- [biddyMain.c](#) (main functions)
- [biddyOp.c](#) (functions for operations on BDDs)
- [biddyStat.c](#) (functions for statistic)
- [biddyInOut.c](#) (parsers and generators for Boolean functions)
- `biddy-example-*.*` (examples)
- `package-source` (script used to build distribution)
- `package-bin` (script used to build distribution)
- `package-bin.bat` (script used to build distribution)
- `package-tgz` (script used to build distribution)
- `package-deb` (script used to build distribution)
- `package-rpm` (script used to build distribution)
- `VS/*` (project files for MS Visual Studio)
- `debian/*` (files used when creating deb package)
- `rpm/*` (files used when creating rpm package)

There are two C headers, external and internal. The external header file, named [biddy.h](#), defines features visible from outside the package. The internal header file, named [biddyInt.h](#) defines features used in multiple files inside the package, but not outside.

Details about building are given in Section 4.

3. USING BIDDY LIBRARY

Precompiled packages include dynamically linked library (i.e. *.so on GNU/Linux, *.dll on MS Windows, *.dylib on Mac OS X), and the appropriate C header [biddy.h](#). Currently, there are no interfaces for other programming languages.

For linking with Biddy library you have to use (remove -lgmp if you have Biddy as a dynamic library):

```
-lbidddy -lgmp
```

The following code is an example of usage. Please note, that functions for node management are not shown. Moreover, Biddy has a manager but its usage is optional and it is also not shown in the given example.

IMPORTANT: You should define UNIX, MACOSX, or MINGW. You should define USE_BIDDDY if you are using Biddy via dynamic library.

```
/* Linux + gcc: compile with gcc -DUNIX -o program.exe source.c -I. -L. -lbidddy -lgmp */
/* WINDOWS + MINGW: compile with gcc -DMINGW -o program.exe source.c -I. -L. -lbidddy -lgmp */
/* in this example, x1, x2, and x3 are Boolean variables */
#include "bidddy.h"
int main() {
    Biddy_Edge f,g,h,r;
    Biddy_Init(); /* use default, i.e. PLAIN ROBDDs WITHOUT COMPLEMENTED EDGES */
    printf("Biddy is using %s.\n",Biddy_GetManagerName());
    f = Biddy_Eval1((Biddy_String)"(OR (AND (NOT x1) x3) (AND x1 (NOT x2) (NOT x3)))"); /* PREFIX
    */
    g = Biddy_Eval2((Biddy_String)"~x1*(~x3*0+x3*1)+x1*(~x2*(~x3*1+x3*0)+x2*0)"); /*
    INFIX */
    if (f == g) printf("Boolean functions f and g are equal.\n");
    if (f != Biddy_Not(g)) printf("Boolean functions f and ~g are not equal.\n");
    printf("Boolean function f depends on %u variables.\n",
        Biddy_DependentVariableNumber(f,FALSE));
    printf("Boolean function f has %.0f minterms.\n",Biddy_CountMinterms(f,0));
    printf("BDD for function f has %u nodes (including both terminals).\n",
        Biddy_CountNodes(f));
    printf("Here is a truth table for Boolean function f\n");
    Biddy_PrintfTable(f);
    Biddy_WriteDot("bidddyexample1.dot",f,"f",-1,FALSE);
    printf("USE 'dot -y -Tpng -O bidddyexample1' to visualize BDD for Boolean function f.\n");
    Biddy_Exit();
}
```

If you do not want to use the default BDD type you simply change the initialization call. Supported BDD types are BIDDYTYPEOBDD, BIDDYTYPEOBDDC, BIDDYTYPEZBDD, BIDDYTYPEZBDDC, and BIDDYTYPEZBDD.

```
Bidddy_InitAnonymous(BIDDYTYPEZBDD);
```

Here is another example. Please note the explicit usage of a manager which enables working with different types of BDDs simultaneously.

```
/* Linux + gcc: compile with gcc -DUNIX -o program.exe source.c -I. -L. -lbidddy -lgmp */
/* WINDOWS + MINGW: compile with gcc -DMINGW -o program.exe source.c -I. -L. -lbidddy -lgmp */
/* in this example, letters D, E, H, L, O, R, and W are Boolean variables */
#include "bidddy.h"
#define Str2Var(mng,x) (Biddy_Managed_GetVariable(mng,(Biddy_String)x))
int main() {
    Biddy_Manager mngobdd, mngobddc;
    Biddy_Edge f,g,r1,r2;

    Biddy_InitMNG(&mngobdd,BIDDYTYPEOBDD); /* PLAIN ROBDDs WITHOUT COMPLEMENTED EDGES */
    Biddy_InitMNG(&mngobddc,BIDDYTYPEOBDDC); /* ROBDDs WITH COMPLEMENTED EDGES */
    f = Biddy_Managed_Eval1(mngobdd,(Biddy_String)"(EXOR H E L L O (AND W O R L D))"); /* PREFIX
    */
    g = Biddy_Managed_Eval2(mngobdd,(Biddy_String)"~(H * E + R * O)"); /*
    INFIX */
    /* BASIC OPERATION */
    r1 = Biddy_Managed_Nor(mngobdd,f,g);
    /* SOP OUTPUT */
}
```

```

printf("Here is Boolean function r1\n");
Bidly_Managed_PrintfSOP(mngobdd,r1);
/* REPLACE SOME VARIABLES: H-->L, R-->L */
Bidly_Managed_ResetVariablesValue(mngobdd);
Bidly_Managed_SetVariableValue(mngobdd,Str2Var(mngobdd,"H"),
    Bidly_Managed_GetVariableEdge(mngobdd,Str2Var(mngobdd,"L")));
Bidly_Managed_SetVariableValue(mngobdd,Str2Var(mngobdd,"R"),
    Bidly_Managed_GetVariableEdge(mngobdd,Str2Var(mngobdd,"L")));
r1 = Bidly_Managed_Replace(mngobdd,r1);
/* SOP OUTPUT AFTER REPLACING VARIABLES */
printf("Here is Boolean function r1 after replacing H-->L and R-->L\n");
Bidly_Managed_PrintfSOP(mngobdd,r1);
/* SIMPLE RESTRICTION */
r1 = Bidly_Managed_Restrict(mngobdd,r1,Str2Var(mngobdd,"E"),FALSE);
/* SOP OUTPUT AFTER REPLACING VARIABLES */
printf("Here is Boolean function r1 after restricting E\n");
Bidly_Managed_PrintfSOP(mngobdd,r1);
/* CONVERT OBDD TO OBDDC */
r2 = Bidly_Managed_Copy(mngobdd,mngobddc,r1);
/* SOME STATS */
printf("OBDD for Boolean function r1 has %u nodes.\n",Bidly_Managed_CountNodes(
    mngobdd,r1));
printf("OBDDC for the same Boolean function has %u nodes.\n",
    Bidly_Managed_CountNodes(mngobddc,r2));
/* GRAPHVIZ/DOT OUTPUT */
Bidly_Managed_WriteDot(mngobdd,"bidlyexample2obdd.dot",r1,"r1",-1,FALSE);
printf("USE 'dot -y -Tpng -O bidlyexample2obdd.dot' to visualize OBDD for function r1.\n");
Bidly_Managed_WriteDot(mngobddc,"bidlyexample2obddc.dot",r2,"r2",-1,FALSE);
printf("USE 'dot -y -Tpng -O bidlyexample2obddc.dot' to visualize OBDDC for function r2.\n");
Bidly_ExitMNG(&mngobdd);
Bidly_ExitMNG(&mngobddc);
}

```

3.1 NODE MANAGEMENT THROUGH FORMULAE PROTECTING

Bidly includes powerful node management based on formulae tagging. There are six user functions to maintain nodes.

[Bidly_AddFormula\(name,bdd,c\)](#)

Given BDD becomes a formula. Its nodes will be preserved for the given number of cleanings. If (name != NULL) then formula is accessible by its name. If formula with a given name already exists it is overwritten. If (c == -1) then formula is refreshed but not preserved. If (c == 0) then formula is persistently preserved. There are five macros defined to simplify adding formulae: [Bidly_AddTmpFormula\(name,bdd\)](#), [Bidly_AddPersistentFormula\(name,bdd\)](#), [Bidly_KeepFormula\(bdd\)](#), [Bidly_KeepFormulaUntilPurge\(bdd\)](#), and [Bidly_KeepFormulaProlonged\(bdd,c\)](#).

[Bidly_DeleteFormula\(name\)](#)

Nodes of the given formula are marked as deleted. Formula is not accessible by its name anymore. Nodes of deleted formula which are preserved or persistently preserved will not be removed by regular GC whilst [Bidly_Purge](#) will immediately remove all of them.

[Bidly_Clean\(\)](#)

Discard all nodes which were not preserved or which are not preserved anymore. Obsolete nodes are not immediately removed, they will be removed during the first garbage collection. Use [Bidly_Purge](#) or [Bidly_PurgeAndReorder](#) to immediately remove all obsolete nodes.

[Bidly_Purge\(\)](#)

Immediately removes all nodes which were not preserved or which are not preserved anymore. Moreover, all formulae without a name are deleted! All nodes from all deleted formulae are removed if they are not needed by other formulae. Call to [Bidly_Purge](#) does not count as cleaning and thus all preserved formulae remains preserved for the same number of cleanings.

Bidly_PurgeAndReorder(bdd)

The same as Bidly_Purge but also trigger reordering on function (if BDD is given) or global reordering (if NULL is given).

Bidly_Refresh(bdd)

All obsolete nodes in the given bdd become fresh nodes. Formula is not created and no information about the existing formulae is changed. This function is needed to implement user's operation caches, only.

3.2. EXAMPLES OF NODE MANAGEMENT WITH BIDDY

The first example is a straightforward calculation.

```
f1 = op(...);
f2 = op(...);
g1 = op(f1,f2,...);
Bidly_KeepFormula(g1); /* g1 is preserved for next cleaning */
f1 = op(...);
f2 = op(...);
g2 = op(f1,f2,...);
Bidly_KeepFormula(g2); /* g2 is preserved for next cleaning */
Bidly_Clean(); /* g1 and g2 are still usable, f1 and f2 are obsolete */
result = op(g1,g2,...);
Bidly_KeepFormulaUntilPurge(result); /* result is preserved until Bidly_Purge() */
Bidly_Clean(); /* only nodes from result remain preserved */
```

If additional garbage collection is needed also after the calculation of g1, you can use the following code after the calculation of g1:

```
Bidly_KeepFormulaProlonged(g1,2); /* g1 is preserved for next two cleanings */
Bidly_Clean(); /* g1 remains preserved for next cleaning */
```

In this approach, f1 and f2 become obsolete after Bidly_Clean, but their nodes are not immediately removed (automatic garbage collection is only started when there are no free nodes in the system).

Alternatively, you can use the following code which is simpler but somehow less efficient because Bidly_Purge() always starts garbage collection:

```
f1 = op(...);
f2 = op(...);
g1 = op(f1,f2,...);
Bidly_AddTmpFormula("g1",g1); /* g1 is named */
Bidly_Purge(); /* keep only nodes from non-obsolete named formulae */
f1 = op(...);
f2 = op(...);
g2 = op(f1,f2,...);
Bidly_AddTmpFormula("g2",g2); /* g2 is named */
Bidly_Purge(); /* keep only nodes from non-obsolete named formulae */
result = op(g1,g2,...);
Bidly_AddPersistentFormula("result",result); /* result is permanently preserved */
Bidly_Clean(); /* all tmp formulae become obsolete */
Bidly_Purge(); /* keep only nodes from non-obsolete named formulae */
```

The second example is an iterative calculation:

```
result = op(...);
while (!finish) {
  Bidly_KeepFormula(result); /* result is preserved for next cleaning */
  Bidly_Clean(); /* result remains preserved */
  f = op(...);
  g = op(f,...);
  result = op(result,g,...);
}
Bidly_KeepFormulaUntilPurge(result); /* result is preserved until Bidly_Purge() */
Bidly_Clean(); /* tmp results are not needed, anymore */
```


If garbage collection is needed also after the calculation of g , you must use the following code:

```
result = op(...);
while (!finish) {
  Biddy_KeepFormulaProlonged(result,2); /* result is preserved for next two cleanings */
  Biddy_Clean(); /* result remains preserved */
  f = op(...);
  g = op(f,...);
  Biddy_KeepFormula(g); /* g is preserved for next cleaning */
  Biddy_Clean(); /* result and g are still usable, f is obsolete */
  result = op(result,g,...);
}
Biddy_KeepFormulaUntilPurge(result); /* result is preserved until Biddy_Purge() */
Biddy_Clean(); /* tmp results are not needed, anymore */
```

The third example is an outline of an implementation of bisimulation:

```
init = AND(init_p,init_q)
Biddy_KeepFormulaUntilPurge(init) /* init is preserved until Biddy_Purge() */
eq = InitialEq(init_p,tr_p,init_q,tr_q,...);
do {
  Biddy_KeepFormula(eq); /* eq is preserved for next cleaning */
  Biddy_Clean(); /* eq remains preserved */
  last = eq;
  eq1 = NextEqPart1(eq,tr_p,tr_q,...);
  eq2 = NextEqPart2(eq,tr_p,tr_q,...);
  eq = AND(eq1,eq2);
} while (AND(init,eq)!=0 && eq!=last)
if (AND(init,eq)!=0) result=false; else result=true;
Biddy_Purge(); /* immediately remove all nodes created during the calculation */
```

The fourth example is an outline of an implementation of model checking:

```
sup = Prepare(...);
Biddy_KeepFormulaUntilPurge(sup) /* preserve for the iteration */
Z = Biddy_GetConstantZero();
last = Biddy_GetConstantOne();
while (Z != last) {
  last = Z;
  Z = Image(Z,sup,...);
  Biddy_KeepFormula(last); /* preserve for the next cleaning */
  Biddy_KeepFormula(Z); /* preserve for the next cleaning */
  Biddy_Clean(); /* remove unnecessary intermediate results */
}
Biddy_AddPersistentFormula("result",Z);
Biddy_Purge(); /* remove all nodes not belonging to result */
```

The fifth example is again a model checking algorithm, but here we are trying to benefit from regularly reordering (in contrast to `Biddy_Purge()`, `Biddy_PurgeAndReorder()` will delete all unnamed formulae, all obsolete formulae, and also all named non-obsolete tmp formulae):

```
sup = Prepare(...);
Biddy_AddPersistentFormula("sup",sup) /* sup is permanently preserved */
Z = 0;
last = 1;
while (Z!=last) {
  Biddy_AddPersistentFormula("Z",Z); /* old Z is marked as deleted */
  Biddy_PurgeAndReorder(Z); /* perform reordering to optimize Z */
  last = Z;
  Z = NextSet(Z,sup,...);
}
result = Z;
Biddy_AddPersistentFormula("result",result); /* final result is permanently preserved */
Biddy_DeleteFormula("sup"); /* sup is marked as deleted */
Biddy_DeleteFormula("Z"); /* Z is marked as deleted */
Biddy_Purge(); /* remove nodes not belonging to result */
```

The sixth example is an outline of an implementation of parallel composition where we are trying to benefit from intensive GC:

```

sacc = snew = AND(init_1,init_2,...,init_N);
for (i=1;i<=N;i++) di[i] = 0;
for (i=1;i<=N;i++) for (j=1;j<=N;j++) dij[i,j] = 0;
do {
  Biddy_KeepFormulaProlonged(snew,N*(N+1)); /* snew is preserved just long enough */
  Biddy_KeepFormulaProlonged(sacc,N*(N+1)); /* sacc is preserved just long enough */
  new1 = 0;
  for (i=1;i<=N;i++) {
    sup = OneStep(snew,tr_i,...);
    di[i] = OR(d[i],sup);
    new1 = OR(new1,NextState(sup,...));
    Biddy_KeepFormulaProlonged(d[i],N*(N+1)); /* di[i] is preserved just long enough */
    Biddy_KeepFormulaProlonged(new1,1); /* new1 is preserved for next cleaning, only */
    Biddy_Clean(); /* new1 remains usable */
  }
  Biddy_KeepFormulaProlonged(new1,N*N); /* new1 is preserved just long enough */
  new2 = 0;
  for (i=1;i<=N;i++) for (j=1;j<=N;j++) {
    sup = OneStep(snew,tr_i,tr_j,...);
    dij[i,j] = OR(d[i,j],sup);
    new2 = OR(new2,NextState(sup,...));
    Biddy_KeepFormulaProlonged(dij[i,j],N*(N+1)); /* dij[i,j] is preserved just long enough */
    Biddy_KeepFormulaProlonged(new2,1); /* new2 is preserved for next cleaning, only */
    Biddy_Clean(); /* new2 remains usable */
  }
  snew = AND(OR(new1,new2),NOT(sacc));
  sacc = OR(sacc,snew);
} while (snew!=0)
tr1 = 0;
for (i=1;i<=N;i++) {
  sup = AddStab(di[i],...);
  tr1 = OR(tr1,sup);
  Biddy_KeepFormulaProlonged(tr1,1); /* tr1 is preserved for next cleaning, only */
  Biddy_Clean(); /* tr1 remains usable */
}
Biddy_KeepFormulaProlonged(tr1,N*N); /* tr1 is preserved just long enough */
tr2 = 0;
for (i=1;i<=N;i++) for (j=1;j<=N;j++) {
  sup = AddStab(dij[i,j],...);
  tr2 = OR(tr2,sup);
  Biddy_KeepFormulaProlonged(tr2,1); /* tr2 is preserved for next cleaning, only */
  Biddy_Clean(); /* tr2 remains usable */
}
result = OR(tr1,tr2);
Biddy_KeepFormulaUntilPurge(result); /* final result is permanently preserved */
Biddy_Clean(); /* tmp results are not needed, anymore */

```

3.3 MORE DETAILS ON MEMORY MANAGEMENT (SYSTEM AGE AND NODE CHAINING)

Garbage collection is automatically triggered if nodes from all reserved blocks of nodes are used. Garbage collection will remove as many obsolete nodes as possible.

Biddy does not use reference counter but a different approach. We call the implemented algorithm "GC with a system age". It is a variant of a mark-and-sweep approach.

Using system age instead of reference counter has some advantages. It allows GC to be started in any time without breaking the ongoing calculation. Thus, there is no need to taking care of repeating broken calculations. Moreover, the usage of system age instead of reference counter removes all the hassle of referencing and dereferencing nodes and thus it is favorable in an academic package oriented towards simple and readable source code.

There are four classes of nodes. Every node belongs to one of these classes:

- **fortified** node (expiry value == 0);
- **fresh** node (expiry value == biddySystemAge);
- **prolonged** node (expiry value > biddySystemAge);
- **obsolete** node (0 < expiry value < biddySystemAge).

Before each GC, nodes must be refreshed in such a way that no successor of a non-obsolete node is obsolete. This is achieved with a relatively simple loop which check every nodes at most once.

Biddy relies on a single hash table for all variables. However, it supports chaining of nodes to form different lists (using an extra pointer in each node). This facility is used to improve the efficiency of garbage collection and sifting.

Please note, that node chaining is not determined or limited by using formulae protecting schema or a system age approach, it is an independent mechanism.

4. BUILDING PACKAGES

Compiling Bidy library

On GNU/Linux, we are using gcc.

```
biddy> make dynamic
biddy> make clean
```

Alternatively, you can use:

```
biddy> make static
biddy> make debug
biddy> make profile
```

You can use specify the target folder:

```
biddy> make dynamic "BINDIR = ./bin"
biddy> make clean "BINDIR = ./bin"
```

On MS Windows, we are using MSYS2. We use pacman to prepare the environment:

```
MSYS shell> pacman -Syuu
MSYS shell> pacman -S mingw-w64-i686-gcc
MSYS shell> pacman -S mingw-w64-x86_64-gcc
MSYS shell> pacman -S make
MSYS shell> pacman -S bison
MSYS shell> pacman -S gdb
MSYS shell> pacman -S nano
MSYS shell> pacman -S tar
MSYS shell> pacman -S subversion
```

Alternatively, you can use Visual Studio for building. There is a prepared solution consisting of many projects (VS/Biddy.sln). You need to adapt include and lib folders.

To produce nice setup files, we use **Advanced Installer** (<http://www.advancedinstaller.com/>). We have been granted a free licence. MANY THANKS!

Dependencies

On GNU/Linux, we are using libgmp (<https://gmplib.org/>).

On MS Windows, we are using MPIR library (<http://mpir.org/>).

Creating Bidy library as a zip package

```
biddy> ./package-bin
```

You need a zip program.

On MS Windows, you need 7-Zip (<http://7-zip.org/>) - and it has a strange use of -x! You also need file 7zsd_All_x64.sfx that you should download as part of "7z SFX Tools" from <http://7zsfx.info/en/> and put in the directory containing 7z.exe.

You install the resulting package by extracting libraries to the appropriate directory (may be local, e.g. user's home directory).

When using this package on GNU/Linux, you have to tell bash about the library:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/absolute/path/to/biddy/library
```

Creating zip file with source code of a complete Bidy project

```
biddy> ./package-source
```

If available, source code of bddview and BDD Scout will be included, too.

Creating packages for GNU/Linux

```
biddy> ./package-tgz  
biddy> ./package-deb  
biddy> ./package-rpm
```

These scripts are intended to be used on Ubuntu. These scripts need release number as an argument. Script `package-tgz` must be invoked before running `package-deb`. Debian packages must be created before RPM packages.

`./package-tgz` should create `orig.tar.gz` file and prepare directories for creating debian and RPM packages. You can run `./package-tgz` only if version changes.

`./package-deb` should create debian packages (`libbiddy` and `libbiddy-dev`). They are tested on Ubuntu system.

`./package-rpm` should create RPM packages (`libbiddy` and `libbiddy-devel`). They are tested on openSUSE system.

Creating demo application bddscout

You need complete sources for `biddy`, `bddview`, and `bddscout`. Scripts are located in `biddy/bddscout`.

```
bddscout> ./package-bin  
bddscout> ./package-tgz  
bddscout> ./package-deb  
bddscout> ./package-rpm
```

`package-bin` should create BDD Scout (statically linked with Bidy library). The script will produce a zip file. You install BDD Scout by simply unzip to the target directory.

`./package-tgz` should create `orig.tar.gz` file and prepare directories for creating debian and RPM packages. You can run `./package-tgz` only if version changes.

`./package-deb` should create debian packages (`bddscout`, `bddscout-bra`, `bddscout-ifip`, `bddscout-bddtrace`, `bddscout-ifip-data`, and `bddscout-bddtrace-data`). They are tested on Ubuntu system.

`./package-rpm` should create RPM packages (`bddscout`, `bddscout-bra`, `bddscout-ifip`, `bddscout-bddtrace`, `bddscout-ifip-data`, and `bddscout-bddtrace-data`). They are tested on openSUSE system.

5. HISTORY

Biddy is based on a BDD package written in Pascal in 1992 as a student project. At that time, it was a great work and the paper about it won a second place at IEEE Region Student Paper Contest (Paris-Evry, 1993). The paper was published by IEEE as "A. Casar, R. Meolic. Representation of Boolean functions with ROBDDs. IEEE Student paper contest : regional contest winners 1990-1997 : prize-winning papers demonstrating student excellence worldwide, Piscataway, 2000" and can be obtained from <http://www.meolic.com/research/papers/robdd-casar-meolic.pdf>.

In 1995, this BDD package was rewritten in C. Later, this BDD package become an integral part of EST package, a prototype tool for formal verification of concurrent systems. The homepage for EST project is <http://est.meolic.com/>.

In 2003, BDD package from EST was included in the report presented at 16th Symposium on Integrated Circuits and Systems Design (SBCCI'03). The report is available as a paper "G. Janssen. A Consumer Report on BDD Packages. IBM T.J. Watson Research Center. 2003". Get it from <http://doi.ieeecomputersociety.org/10.1109/SBCCI.2003.1232832>.

In 2006, BDD package in EST got the name Biddy.

In 2007, a main part of Biddy package was extracted from EST forming a separate package called Biddy. The code has been reorganized in such a way, that EST is not using internal structures (e.g. node table) directly but using the provided API only.

In 2007, we created local svn repository for maintaining the source code (not live, anymore).

On May 15, 2008, Biddy v1.0 was released containing also bddview v0.95 (Tcl/Tk BDD viewer) and Bdd Scout v0.90 (demo application).

In 2009, 2010, and 2012 updated versions of Biddy v1.0 were released which added support for debian packaging, support for RPM packaging, fix errors, and improve documentation, packaging, and Tcl/Tk GUI.

In 2012, a research paper about Biddy library appears in Journal of Software (doi:10.4304/jsw.7.6.1358-1366) <http://www.jsoftware.us/vol7/jsw0706-23.pdf>.

In 2013, Biddy v1.1 was released. [Biddy_Edge](#) becomes a pointer instead of a structure and all other structures were optimized.

In 2014, Biddy v1.2 was released. Variable swapping and sifting algorithm were the most significant additions.

In 2014, svn repositories for biddy, bddview and bddscout are moved to Savannah. <http://svn.savannah.nongnu.org/viewvc/?root=biddy>.

In 2015, Biddy v1.3, v1.4 and v1.5 was released. Various input/output methods have been added. Support for 64-bit architectures and support for Visual Studio projects were improved. Biddy got a manager. Many CUDD-like functions have been added. Comment's style changed to support doxygen. HTML and PDF documentation were produced.

Also in 2015, Personal Package Archive ppa:meolic/biddy has been created <https://launchpad.net/~meolic/+archive/ubuntu/biddy>.

Also in 2015, sources became available on GitHub <https://github.com/meolic/biddy>.

In 2016, Biddy v1.6 was released. Formulae are not recursively refreshed all the time, anymore. The size of Node table became resizable.

In 2017, Biddy v1.7 was released (there were 4 minor releases). Terminology has changed a lot, e.g. "formulae counter" became "system age". Added support for ROBDDs without complemented edges, 0-sup-BDDs and Tagged 0-sup-BDDs. Implemented creation and manipulation of non-anonymous managers. Added manipulation of combination sets. Improved many functionalities, e.g sifting. Many new CUDD-like functions have been added. Moreover, bddview and BDD Scout have been significantly improved.

In 2018, Biddy v1.8. was released (there were 2 minor releases). Added support for 0-sup-BDDs without complemented edges. Added several new operations on 0-sup-BDDs that enable us to write a very interesting paper "Flexible job shop scheduling using zero-suppressed binary decision diagrams". Get it from http://apem-journal.org/Archives/2018/APEM13-4_373-388.pdf. Again, bddview and BDD Scout have been significantly improved. Started BDD Encyclopedia, check it out at http://svn.savannah.nongnu.org/viewvc/*checkout*/biddy/bddscout/ENCYCLOPEDIA/bddencyclopedia.html.

6. PUBLICATIONS

If you find our work useful, please, cite us.

- Robert Meolic. **Biddy - a multi-platform academic BDD package**. Journal of Software, 7(6), pp. 1358-1366, 2012. <http://ojs.academypublisher.com/index.php/jsw/article/view/jsw070613581366>
- Robert Meolic. **Implementation aspects of a BDD package supporting general decision diagrams**. Technical report, University of Maribor, 2016. <https://dk.um.si/IzpisGradiva.php?id=68831>
- Robert Meolic. **The Biddy BDD package**. University of Maribor, 2019. We are preparing a paper for JOSS.

2 Data Structure Documentation

2.1 Biddy_Boolean Class Reference

[Biddy_Boolean](#) is used for boolean values.

```
#include <biddy.h>
```

The documentation for this class was generated from the following file:

- [biddy.h](#)

2.2 Biddy_Cache Class Reference

[Biddy_Cache](#) is used to specify user's cache table.

```
#include <biddy.h>
```

2.2.1 Detailed Description

Caches for different operations are different and the user is responsible for the correct internal structure.

The documentation for this class was generated from the following file:

- [biddy.h](#)

2.3 Biddy_Edge Class Reference

[Biddy_Edge](#) is a marked edge (i.e. a marked pointer to BiddyNode).

```
#include <biddy.h>
```

2.3.1 Detailed Description

Mark is encoded as the value of the last significant bit. For TZBDDs and TZFDDs, edges are tagged. Tag is a 16 bit number (unsigned short int) which is stored in the highest part of the pointer (this is safe because only 48 bits are used). TZBDDs and TZFDDs are supported only on 64-bits architectures. Internal structure of BiddyNode is not visible to the user.

The documentation for this class was generated from the following file:

- [biddy.h](#)

2.4 Biddy_GCFunction Class Reference

[Biddy_GCFunction](#) is used in `Biddy_AddCache` to specify user's function which will performs garbage collection.

```
#include <biddy.h>
```

The documentation for this class was generated from the following file:

- [biddy.h](#)

2.5 Biddy_LookupFunction Class Reference

[Biddy_LookupFunction](#) is used in `Biddy_Eval1x` to specify user's function which will lookups in a user's formula table.

```
#include <biddy.h>
```

The documentation for this class was generated from the following file:

- [biddy.h](#)

2.6 Biddy_Manager Class Reference

[Biddy_Manager](#) is used to specify manager.

```
#include <biddy.h>
```

2.6.1 Detailed Description

Manager is a pointer to BiddyManager. A manager includes Node Table, Variable Table, Formulae Table, Ordering Table, three basic caches (ITE Cache, EA Cache and RC Cache), list of user's caches, system age and some other structures needed for memory management. Internal structure of BiddyManager is not exported but must be imitated to create user's managers

The documentation for this class was generated from the following file:

- [biddy.h](#)

2.7 Biddy_String Class Reference

[Biddy_String](#) is used for strings.

```
#include <biddy.h>
```

The documentation for this class was generated from the following file:

- [biddy.h](#)

2.8 Biddy_Variable Class Reference

[Biddy_Variable](#) is used for indices in variable table.

```
#include <biddy.h>
```

The documentation for this class was generated from the following file:

- [biddy.h](#)

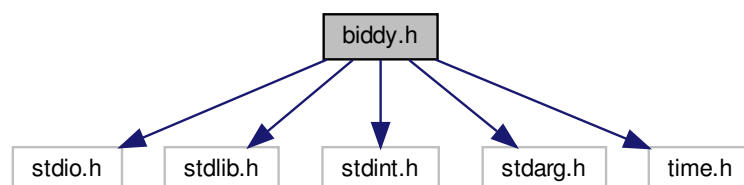
3 File Documentation

3.1 biddy.h File Reference

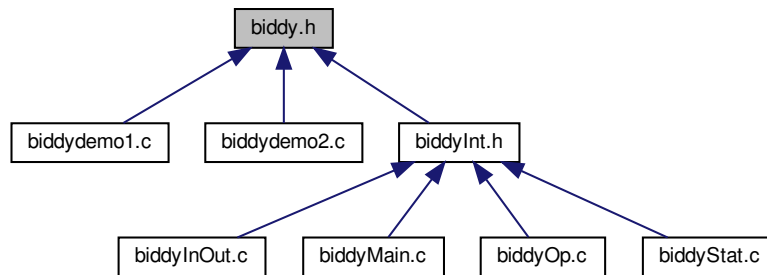
File [biddy.h](#) contains declaration of all external data structures.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdarg.h>
#include <time.h>
```

Include dependency graph for biddy.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **TRUE** (0 == 0)
- #define **FALSE** !TRUE
- #define **BIDDYTYPEOBDD** 1
- #define **BIDDYTYPENAMEOBDD** "ROBDD"
- #define **BIDDYTYPEOBDDC** 2
- #define **BIDDYTYPENAMEOBDDC** "ROBDD WITH COMPLEMENTED EDGES"
- #define **BIDDYTYPEZBDD** 3
- #define **BIDDYTYPENAMEZBDD** "ZBDD"
- #define **BIDDYTYPEZBDDC** 4
- #define **BIDDYTYPENAMEZBDDC** "ZBDD WITH COMPLEMENTED EDGES"
- #define **BIDDYTYPEZBDD** 5
- #define **BIDDYTYPENAMEZBDD** "TAGGED ZBDD"
- #define **BIDDYTYPEZBDDC** 6
- #define **BIDDYTYPENAMEZBDDC** "TAGGED ZBDD WITH COMPLEMENTED EDGES"
- #define **BIDDYTYPEOFDD** 7
- #define **BIDDYTYPENAMEOFDD** "ROFDD"
- #define **BIDDYTYPEOFDDC** 8
- #define **BIDDYTYPENAMEOFDDC** "ROFDD WITH COMPLEMENTED EDGES"
- #define **BIDDYTYPEZFDD** 9
- #define **BIDDYTYPENAMEZFDD** "ZFDD"
- #define **BIDDYTYPEZFDDC** 10
- #define **BIDDYTYPENAMEZFDDC** "ZFDD WITH COMPLEMENTED EDGES"
- #define **BIDDYTYPEZFDD** 11
- #define **BIDDYTYPENAMEZFDD** "TZFDD"
- #define **BIDDYTYPEZFDDC** 12
- #define **BIDDYTYPENAMEZFDDC** "TZFDD WITH COMPLEMENTED EDGES"
- #define **Biddy_IsNull**(f) (f == NULL)
- #define **Biddy_IsTerminal**(f) (((void*)((uintptr_t) f & ~((uintptr_t) 1)))&[2] == NULL) && (((void*)((uintptr_t) f & ~((uintptr_t) 1)))&[3] == NULL)
- #define **Biddy_IsEqvPointer**(f, g) (((uintptr_t) f & ~((uintptr_t) 1)) == ((uintptr_t) g & ~((uintptr_t) 1)))
- #define **Biddy_GetMark**(f) (((uintptr_t) f & (uintptr_t) 1) != 0)
- #define **Biddy_SetMark**(f) (f = (**Biddy_Edge**) ((uintptr_t) f | (uintptr_t) 1))
- #define **Biddy_ClearMark**(f) (f = (**Biddy_Edge**) ((uintptr_t) f & ~((uintptr_t) 1)))
- #define **Biddy_InvertMark**(f) (f = (**Biddy_Edge**) ((uintptr_t) f ^ (uintptr_t) 1))
- #define **Biddy_Inv**(f) ((**Biddy_Edge**) ((uintptr_t) f ^ (uintptr_t) 1))

- #define [Bidly_InvCond](#)(f, c) (c ? (([Bidly_Edge](#)) ((uintptr_t) f ^ (uintptr_t) 1)) : f)
- #define [Bidly_Regular](#)(f) (([Bidly_Edge](#)) ((uintptr_t) f & ~((uintptr_t) 1)))
- #define [Bidly_Complement](#)(f) (([Bidly_Edge](#)) ((uintptr_t) f | (uintptr_t) 1))
- #define [Bidly_GetTag](#)(f) 0
- #define [Bidly_SetTag](#)(f, t) 0
- #define [Bidly_ClearTag](#)(f) 0
- #define [Bidly_Untagged](#)(f) 0
- #define [Bidly_Init](#)() [Bidly_InitMNG](#)(NULL, BIDDYTYPEOBDD)
- #define [Bidly_InitAnonymous](#)(bddtype) [Bidly_InitMNG](#)(NULL, bddtype)
- #define [Bidly_Exit](#)() [Bidly_ExitMNG](#)(NULL)
- #define [Bidly_GetManagerType](#)() [Bidly_Managed_GetManagerType](#)(NULL)
- #define [Bidly_GetManagerName](#)() [Bidly_Managed_GetManagerName](#)(NULL)
- #define [Bidly_SetManagerParameters](#)(gcr, gcrF, gcrX, rr, rrF, rrX, st, cst) [Bidly_Managed_SetManagerParameters](#)(NULL, gcr, gcrF, gcrX, rr, rrF, rrX, st, cst)
- #define [Bidly_Managed_GetThen](#)(MNG, f) [Bidly_GetThen](#)(f)
- #define [Bidly_Managed_GetElse](#)(MNG, f) [Bidly_GetElse](#)(f)
- #define [Bidly_Managed_GetTopVariable](#)(MNG, f) [Bidly_GetTopVariable](#)(f)
- #define [Bidly_IsEqv](#)(f1, MNG2, f2) [Bidly_Managed_IsEqv](#)(NULL, f1, MNG2, f2)
- #define [Bidly_SelectNode](#)(f) [Bidly_Managed_SelectNode](#)(NULL, f)
- #define [Bidly_DeselectNode](#)(f) [Bidly_Managed_DeselectNode](#)(NULL, f)
- #define [Bidly_IsSelected](#)(f) [Bidly_Managed_IsSelected](#)(NULL, f)
- #define [Bidly_SelectFunction](#)(f) [Bidly_Managed_SelectFunction](#)(NULL, f)
- #define [Bidly_DeselectAll](#)() [Bidly_Managed_DeselectAll](#)(NULL)
- #define [Bidly_GetTerminal](#)() [Bidly_Managed_GetTerminal](#)(NULL)
- #define [Bidly_GetConstantZero](#)() [Bidly_Managed_GetConstantZero](#)(NULL)
- #define [Bidly_Managed_GetEmptySet](#)(MNG) [Bidly_Managed_GetConstantZero](#)(MNG)
- #define [Bidly_GetEmptySet](#)() [Bidly_Managed_GetConstantZero](#)(NULL)
- #define [Bidly_GetConstantOne](#)() [Bidly_Managed_GetConstantOne](#)(NULL)
- #define [Bidly_Managed_GetUniversalSet](#)(MNG) [Bidly_Managed_GetConstantOne](#)(MNG)
- #define [Bidly_GetUniversalSet](#)() [Bidly_Managed_GetConstantOne](#)(NULL)
- #define [Bidly_GetBaseSet](#)() [Bidly_Managed_GetBaseSet](#)(NULL)
- #define [Bidly_GetVariable](#)(x) [Bidly_Managed_GetVariable](#)(NULL, x)
- #define [Bidly_GetLowestVariable](#)() [Bidly_Managed_GetLowestVariable](#)(NULL)
- #define [Bidly_GetIthVariable](#)(i) [Bidly_Managed_GetIthVariable](#)(NULL, i)
- #define [Bidly_GetPrevVariable](#)(v) [Bidly_Managed_GetPrevVariable](#)(NULL, v)
- #define [Bidly_GetNextVariable](#)(v) [Bidly_Managed_GetNextVariable](#)(NULL, v)
- #define [Bidly_GetVariableEdge](#)(v) [Bidly_Managed_GetVariableEdge](#)(NULL, v)
- #define [Bidly_GetElementEdge](#)(v) [Bidly_Managed_GetElementEdge](#)(NULL, v)
- #define [Bidly_GetVariableName](#)(v) [Bidly_Managed_GetVariableName](#)(NULL, v)
- #define [Bidly_GetTopVariableEdge](#)(f) [Bidly_Managed_GetTopVariableEdge](#)(NULL, f)
- #define [Bidly_GetTopVariableName](#)(f) [Bidly_Managed_GetTopVariableName](#)(NULL, f)
- #define [Bidly_GetTopVariableChar](#)(f) [Bidly_Managed_GetTopVariableChar](#)(NULL, f)
- #define [Bidly_ResetVariablesValue](#)() [Bidly_Managed_ResetVariablesValue](#)(NULL)
- #define [Bidly_SetVariableValue](#)(v, f) [Bidly_Managed_SetVariableValue](#)(NULL, v, f)
- #define [Bidly_GetVariableValue](#)(v) [Bidly_Managed_GetVariableValue](#)(NULL, v)
- #define [Bidly_ClearVariablesData](#)() [Bidly_Managed_ClearVariablesData](#)(NULL)
- #define [Bidly_SetVariableData](#)(v, x) [Bidly_Managed_SetVariableData](#)(NULL, v, x)
- #define [Bidly_GetVariableData](#)(v) [Bidly_Managed_GetVariableData](#)(NULL, v)
- #define [Bidly_Eval](#)(f) [Bidly_Managed_Eval](#)(NULL, f)
- #define [Bidly_EvalProbability](#)(f) [Bidly_Managed_EvalProbability](#)(NULL, f)
- #define [Bidly_IsSmaller](#)(fv, gv) [Bidly_Managed_IsSmaller](#)(NULL, fv, gv)
- #define [Bidly_IsLowest](#)(v) [Bidly_Managed_IsLowest](#)(NULL, v)
- #define [Bidly_IsHighest](#)(v) [Bidly_Managed_IsHighest](#)(NULL, v)
- #define [Bidly_FoaVariable](#)(x, varelem) [Bidly_Managed_FoaVariable](#)(NULL, x, varelem)
- #define [Bidly_ChangeVariableName](#)(v, x) [Bidly_Managed_ChangeVariableName](#)(NULL, v, x)

- #define `Biddy_AddVariableByName(x) Biddy_Managed_AddVariableByName(NULL,x)`
- #define `Biddy_Managed_AddVariable(MNG) Biddy_Managed_AddVariableByName(MNG,NULL)`
- #define `Biddy_AddVariable() Biddy_Managed_AddVariableByName(NULL,NULL)`
- #define `Biddy_Managed_AddVariableEdge(MNG) Biddy_Managed_GetVariableEdge(MNG,Biddy_Managed_AddVariableByName(MNG,NULL))`
- #define `Biddy_AddVariableEdge() Biddy_Managed_GetVariableEdge(NULL,Biddy_Managed_AddVariableByName(NULL,NULL))`
- #define `Biddy_AddElementByName(x) Biddy_Managed_AddElementByName(NULL,x)`
- #define `Biddy_Managed_AddElement(MNG) Biddy_Managed_AddElementByName(MNG,NULL)`
- #define `Biddy_AddElement() Biddy_Managed_AddElementByName(NULL,NULL)`
- #define `Biddy_AddVariableBelow(v) Biddy_Managed_AddVariableBelow(NULL,v)`
- #define `Biddy_AddVariableAbove(v) Biddy_Managed_AddVariableAbove(NULL,v)`
- #define `Biddy_TransferMark(f, mark, leftright) Biddy_Managed_TransferMark(NULL,f,mark,leftright)`
- #define `Biddy_IncTag(f) Biddy_Managed_IncTag(NULL,f)`
- #define `Biddy_TaggedFoaNode(v, pf, pt, ptag, garbageAllowed) Biddy_Managed_TaggedFoaNode(NULL,v,pf,pt,ptag,garbageAllowed)`
- #define `Biddy_Managed_FoaNode(MNG, v, pf, pt, garbageAllowed) Biddy_Managed_TaggedFoaNode(MNG,v,pf,pt,v,garbageAllowed)`
- #define `Biddy_FoaNode(v, pf, pt, garbageAllowed) Biddy_Managed_TaggedFoaNode(NULL,v,pf,pt,v,garbageAllowed)`
- #define `Biddy_IsOK(f) Biddy_Managed_IsOK(NULL,f)`
- #define `Biddy_GC(targetLT, targetGEQ, purge, total) Biddy_Managed_GC(NULL,targetLT,targetGEQ,purge,total)`
- #define `Biddy_Managed_AutoGC(MNG) Biddy_Managed_GC(MNG,0,0,FALSE,FALSE)`
- #define `Biddy_AutoGC() Biddy_Managed_GC(NULL,0,0,FALSE,FALSE)`
- #define `Biddy_Managed_ForceGC(MNG) Biddy_Managed_GC(MNG,0,0,FALSE,TRUE)`
- #define `Biddy_ForceGC() Biddy_Managed_GC(NULL,0,0,FALSE,TRUE)`
- #define `Biddy_Clean() Biddy_Managed_Clean(NULL)`
- #define `Biddy_Purge() Biddy_Managed_Purge(NULL)`
- #define `Biddy_PurgeAndReorder(f, c) Biddy_Managed_PurgeAndReorder(NULL,f,c)`
- #define `Biddy_Refresh(f) Biddy_Managed_Refresh(NULL,f)`
- #define `Biddy_AddCache(gc) Biddy_Managed_AddCache(NULL,gc)`
- #define `Biddy_AddFormula(x, f, c) Biddy_Managed_AddFormula(NULL,x,f,c)`
- #define `Biddy_Managed_AddTmpFormula(mng, x, f) Biddy_Managed_AddFormula(mng,x,f,-1)`
- #define `Biddy_Managed_AddPersistentFormula(mng, x, f) Biddy_Managed_AddFormula(mng,x,f,0)`
- #define `Biddy_Managed_KeepFormula(mng, f) Biddy_Managed_AddFormula(mng,NULL,f,1)`
- #define `Biddy_Managed_KeepFormulaProlonged(mng, f, c) Biddy_Managed_AddFormula(mng,NULL,f,c)`
- #define `Biddy_Managed_KeepFormulaUntilPurge(mng, f) Biddy_Managed_AddFormula(mng,NULL,f,0)`
- #define `Biddy_AddTmpFormula(x, f) Biddy_Managed_AddFormula(NULL,x,f,-1)`
- #define `Biddy_AddPersistentFormula(x, f) Biddy_Managed_AddFormula(NULL,x,f,0)`
- #define `Biddy_KeepFormula(f) Biddy_Managed_AddFormula(NULL,NULL,f,1)`
- #define `Biddy_KeepFormulaProlonged(f, c) Biddy_Managed_AddFormula(NULL,NULL,f,c)`
- #define `Biddy_KeepFormulaUntilPurge(f) Biddy_Managed_AddFormula(NULL,NULL,f,0)`
- #define `Biddy_FindFormula(x, idx, f) Biddy_Managed_FindFormula(NULL,x,idx,f)`
- #define `Biddy_DeleteFormula(x) Biddy_Managed_DeleteFormula(NULL,x)`
- #define `Biddy_DeletelthFormula(x) Biddy_Managed_DeletelthFormula(NULL,x)`
- #define `Biddy_GetlthFormula(i) Biddy_Managed_GetlthFormula(NULL,i)`
- #define `Biddy_GetlthFormulaName(i) Biddy_Managed_GetlthFormulaName(NULL,i)`
- #define `Biddy_GetOrdering() Biddy_Managed_GetOrdering(NULL)`
- #define `Biddy_SetOrdering(ordering) Biddy_Managed_SetOrdering(NULL,ordering)`
- #define `Biddy_SetAlphabeticOrdering() Biddy_Managed_SetAlphabeticOrdering(NULL)`
- #define `Biddy_SwapWithHigher(v) Biddy_Managed_SwapWithHigher(NULL,v)`
- #define `Biddy_SwapWithLower(v) Biddy_Managed_SwapWithLower(NULL,v)`
- #define `Biddy_Sifting(f, c) Biddy_Managed_Sifting(NULL,f,c)`
- #define `Biddy_MinimizeBDD(f) Biddy_Managed_MinimizeBDD(NULL,f)`

- #define [Biddy_MaximizeBDD](#)(f) [Biddy_Managed_MaximizeBDD](#)(NULL,f)
- #define [Biddy_Copy](#)(MNG2, f) [Biddy_Managed_Copy](#)(NULL,MNG2,f)
- #define [Biddy_CopyFormulaFrom](#)(MNG1, x) [Biddy_Managed_CopyFormula](#)(MNG1,NULL,x)
- #define [Biddy_CopyFormulaTo](#)(MNG2, x) [Biddy_Managed_CopyFormula](#)(NULL,MNG2,x)
- #define [Biddy_ConstructBDD](#)(numV, varlist, numN, nodelist) [Biddy_Managed_ConstructBDD](#)(NULL,numV,varlist,numV,nodelist)
- #define [Biddy_Not](#)(f) [Biddy_Managed_Not](#)(NULL,f)
- #define [Biddy_ITE](#)(f, g, h) [Biddy_Managed_ITE](#)(NULL,f,g,h)
- #define [Biddy_And](#)(f, g) [Biddy_Managed_And](#)(NULL,f,g)
- #define [Biddy_Managed_Intersect](#)(MNG, f, g) [Biddy_Managed_And](#)(MNG,f,g)
- #define [Biddy_Intersect](#)(f, g) [Biddy_Managed_And](#)(NULL,f,g)
- #define [Biddy_Or](#)(f, g) [Biddy_Managed_Or](#)(NULL,f,g)
- #define [Biddy_Managed_Union](#)(MNG, f, g) [Biddy_Managed_Or](#)(MNG,f,g)
- #define [Biddy_Union](#)(f, g) [Biddy_Managed_Or](#)(NULL,f,g)
- #define [Biddy_Nand](#)(f, g) [Biddy_Managed_Nand](#)(NULL,f,g)
- #define [Biddy_Nor](#)(f, g) [Biddy_Managed_Nor](#)(NULL,f,g)
- #define [Biddy_Xor](#)(f, g) [Biddy_Managed_Xor](#)(NULL,f,g)
- #define [Biddy_Xnor](#)(f, g) [Biddy_Managed_Xnor](#)(NULL,f,g)
- #define [Biddy_Leq](#)(f, g) [Biddy_Managed_Leq](#)(NULL,f,g)
- #define [Biddy_Gt](#)(f, g) [Biddy_Managed_Gt](#)(NULL,f,g)
- #define [Biddy_Managed_Diff](#)(MNG, f, g) [Biddy_Managed_Gt](#)(MNG,f,g)
- #define [Biddy_Diff](#)(f, g) [Biddy_Managed_Gt](#)(NULL,f,g)
- #define [Biddy_IsLeq](#)(f, g) [Biddy_Managed_IsLeq](#)(NULL,f,g)
- #define [Biddy_Restrict](#)(f, v, value) [Biddy_Managed_Restrict](#)(NULL,f,v,value)
- #define [Biddy_Compose](#)(f, g, v) [Biddy_Managed_Compose](#)(NULL,f,g,v)
- #define [Biddy_E](#)(f, v) [Biddy_Managed_E](#)(NULL,f,v)
- #define [Biddy_A](#)(f, v) [Biddy_Managed_A](#)(NULL,f,v)
- #define [Biddy_IsVariableDependent](#)(f, v) [Biddy_Managed_IsVariableDependent](#)(NULL,f,v)
- #define [Biddy_ExistAbstract](#)(f, cube) [Biddy_Managed_ExistAbstract](#)(NULL,f,cube)
- #define [Biddy_UnivAbstract](#)(f, cube) [Biddy_Managed_UnivAbstract](#)(NULL,f,cube)
- #define [Biddy_AndAbstract](#)(f, g, cube) [Biddy_Managed_AndAbstract](#)(NULL,f,g,cube)
- #define [Biddy_Constrain](#)(f, c) [Biddy_Managed_Constrain](#)(NULL,f,c)
- #define [Biddy_Simplify](#)(f, c) [Biddy_Managed_Simplify](#)(NULL,f,c)
- #define [Biddy_Support](#)(f) [Biddy_Managed_Support](#)(NULL,f)
- #define [Biddy_ReplaceByKeyword](#)(f, keyword) [Biddy_Managed_ReplaceByKeyword](#)(NULL,f,keyword)
- #define [Biddy_Managed_Replace](#)(MNG, f) [Biddy_Managed_ReplaceByKeyword](#)(MNG,f,NULL)
- #define [Biddy_Replace](#)(f) [Biddy_Managed_ReplaceByKeyword](#)(NULL,f,NULL)
- #define [Biddy_Change](#)(f, v) [Biddy_Managed_Change](#)(NULL,f,v)
- #define [Biddy_VarSubset](#)(f, v, value) [Biddy_Managed_VarSubset](#)(NULL,f,v,value)
- #define [Biddy_Managed_Subset0](#)(MNG, f, v) [Biddy_Managed_VarSubset](#)(MNG,f,v,FALSE)
- #define [Biddy_Subset0](#)(f, v) [Biddy_Managed_VarSubset](#)(NULL,f,v,FALSE)
- #define [Biddy_Managed_Subset1](#)(MNG, f, v) [Biddy_Managed_VarSubset](#)(MNG,f,v,TRUE)
- #define [Biddy_Subset1](#)(f, v) [Biddy_Managed_VarSubset](#)(NULL,f,v,TRUE)
- #define [Biddy_Managed_Quotient](#)(MNG, f, v) [Biddy_Managed_Change](#)(MNG,Biddy_Managed_VarSubset(MNG,f,v,TRUE),v)
- #define [Biddy_Quotient](#)(f, v) [Biddy_Change](#)(Biddy_VarSubset(f,v,TRUE),v)
- #define [Biddy_Managed_Remainder](#)(MNG, f, v) [Biddy_Managed_VarSubset](#)(MNG,f,v,FALSE)
- #define [Biddy_Remainder](#)(f, v) [Biddy_Managed_VarSubset](#)(NULL,f,v,FALSE)
- #define [Biddy_ElementAbstract](#)(f, v) [Biddy_Managed_ElementAbstract](#)(NULL,f,v)
- #define [Biddy_Product](#)(f, g) [Biddy_Managed_Product](#)(NULL,f,g)
- #define [Biddy_SelectiveProduct](#)(f, g, cube) [Biddy_Managed_SelectiveProduct](#)(NULL,f,g,cube)
- #define [Biddy_Supset](#)(f, g) [Biddy_Managed_Supset](#)(NULL,f,g)
- #define [Biddy_Subset](#)(f, g) [Biddy_Managed_Subset](#)(NULL,f,g)
- #define [Biddy_Permitsym](#)(f, n) [Biddy_Managed_Permitsym](#)(NULL,f,n)
- #define [Biddy_Stretch](#)(f) [Biddy_Managed_Stretch](#)(NULL,f)

- #define Biddy_CreateMinterm(support, x) Biddy_Managed_CreateMinterm(NULL,support,x)
- #define Biddy_CreateFunction(support, x) Biddy_Managed_CreateFunction(NULL,support,x)
- #define Biddy_RandomFunction(support, r) Biddy_Managed_RandomFunction(NULL,support,r)
- #define Biddy_RandomSet(unit, r) Biddy_Managed_RandomSet(NULL,unit,r)
- #define Biddy_ExtractMinterm(support, f) Biddy_Managed_ExtractMinterm(NULL,support,f)
- #define Biddy_CountNodes(f) Biddy_Managed_CountNodes(NULL,f)
- #define Biddy_Managed_MaxLevel(MNG, f) Biddy_MaxLevel(f)
- #define Biddy_Managed_AvgLevel(MNG, f) Biddy_AvgLevel(f)
- #define Biddy_VariableTableNum() Biddy_Managed_VariableTableNum(NULL)
- #define Biddy_NodeTableSize() Biddy_Managed_NodeTableSize(NULL)
- #define Biddy_NodeTableBlockNumber() Biddy_Managed_NodeTableBlockNumber(NULL)
- #define Biddy_NodeTableGenerated() Biddy_Managed_NodeTableGenerated(NULL)
- #define Biddy_NodeTableMax() Biddy_Managed_NodeTableMax(NULL)
- #define Biddy_NodeTableNum() Biddy_Managed_NodeTableNum(NULL)
- #define Biddy_NodeTableNumVar(v) Biddy_Managed_NodeTableNumVar(NULL,v)
- #define Biddy_NodeTableResizeNumber() Biddy_Managed_NodeTableResizeNumber(NULL)
- #define Biddy_NodeTableFoaNumber() Biddy_Managed_NodeTableFoaNumber(NULL)
- #define Biddy_NodeTableFindNumber() Biddy_Managed_NodeTableFindNumber(NULL)
- #define Biddy_NodeTableCompareNumber() Biddy_Managed_NodeTableCompareNumber(NULL)
- #define Biddy_NodeTableAddNumber() Biddy_Managed_NodeTableAddNumber(NULL)
- #define Biddy_NodeTableGCNumber() Biddy_Managed_NodeTableGCNumber(NULL)
- #define Biddy_NodeTableGCTime() Biddy_Managed_NodeTableGCTime(NULL)
- #define Biddy_NodeTableGCObsoleteNumber() Biddy_Managed_NodeTableGCObsoleteNumber(NULL)
- #define Biddy_NodeTableSwapNumber() Biddy_Managed_NodeTableSwapNumber(NULL)
- #define Biddy_NodeTableSiftingNumber() Biddy_Managed_NodeTableSiftingNumber(NULL)
- #define Biddy_NodeTableDRTime() Biddy_Managed_NodeTableDRTime(NULL)
- #define Biddy_NodeTableITENumber() Biddy_Managed_NodeTableITENumber(NULL)
- #define Biddy_NodeTableITERRecursiveNumber() Biddy_Managed_NodeTableITERRecursiveNumber(NULL)
- #define Biddy_NodeTableANDORNumber() Biddy_Managed_NodeTableANDORNumber(NULL)
- #define Biddy_NodeTableANDORRecursiveNumber() Biddy_Managed_NodeTableANDORRecursiveNumber(NULL)
- #define Biddy_NodeTableXORNumber() Biddy_Managed_NodeTableXORNumber(NULL)
- #define Biddy_NodeTableXORRecursiveNumber() Biddy_Managed_NodeTableXORRecursiveNumber(NULL)
- #define Biddy_FormulaTableNum() Biddy_Managed_FormulaTableNum(NULL)
- #define Biddy_ListUsed() Biddy_Managed_ListUsed(NULL)
- #define Biddy_ListMaxLength() Biddy_Managed_ListMaxLength(NULL)
- #define Biddy_ListAvgLength() Biddy_Managed_ListAvgLength(NULL)
- #define Biddy_OPCacheSearch() Biddy_Managed_OPCacheSearch(NULL)
- #define Biddy_OPCacheFind() Biddy_Managed_OPCacheFind(NULL)
- #define Biddy_OPCacheInsert() Biddy_Managed_OPCacheInsert(NULL)
- #define Biddy_OPCacheOverwrite() Biddy_Managed_OPCacheOverwrite(NULL)
- #define Biddy_CountNodesPlain(f) Biddy_Managed_CountNodesPlain(NULL,f)
- #define Biddy_DependentVariableNumber(f, select) Biddy_Managed_DependentVariableNumber(NULL,f,select)
- #define Biddy_CountComplementedEdges(f) Biddy_Managed_CountComplementedEdges(NULL,f)
- #define Biddy_CountPaths(f) Biddy_Managed_CountPaths(NULL,f)
- #define Biddy_CountMinterms(f, nvars) Biddy_Managed_CountMinterms(NULL,f,nvars)
- #define Biddy_Managed_CountCombinations(MNG, f, nvars) Biddy_Managed_CountMinterms(MNG,f,nvars)
- #define Biddy_CountCombinations(f) Biddy_Managed_CountMinterms(NULL,f,nvars)
- #define Biddy_DensityOfFunction(f, nvars) Biddy_Managed_DensityOfFunction(NULL,f,nvars)
- #define Biddy_DensityOfBDD(f, nvars) Biddy_Managed_DensityOfBDD(NULL,f,nvars)
- #define Biddy_MinNodes(f) Biddy_Managed_MinNodes(NULL,f)
- #define Biddy_MaxNodes(f) Biddy_Managed_MaxNodes(NULL,f)

- #define [Bidly_ReadMemoryInUse\(\)](#) [Bidly_Managed_ReadMemoryInUse](#)(NULL)
- #define [Bidly_PrintInfo](#)(f) [Bidly_Managed_PrintInfo](#)(NULL,f)
- #define [Bidly_Eval0](#)(s) [Bidly_Managed_Eval0](#)(NULL,s)
- #define [Bidly_Eval1x](#)(s, lf) [Bidly_Managed_Eval1x](#)(NULL,s,lf)
- #define [Bidly_Managed_Eval1](#)(MNG, s) [Bidly_Managed_Eval1x](#)(MNG,s,NULL)
- #define [Bidly_Eval1](#)(s) [Bidly_Managed_Eval1x](#)(NULL,s,NULL)
- #define [Bidly_Eval2](#)(boolFunc) [Bidly_Managed_Eval2](#)(NULL,boolFunc)
- #define [Bidly_ReadBddview](#)(filename, name) [Bidly_Managed_ReadBddview](#)(NULL,filename,name)
- #define [Bidly_ReadVerilogFile](#)(filename, prefix) [Bidly_Managed_ReadVerilogFile](#)(NULL,filename,prefix)
- #define [Bidly_PrintfBDD](#)(f) [Bidly_Managed_PrintfBDD](#)(NULL,f)
- #define [Bidly_WriteBDD](#)(filename, f, label) [Bidly_Managed_WriteBDD](#)(NULL,filename,f,label)
- #define [Bidly_PrintfTable](#)(f) [Bidly_Managed_PrintfTable](#)(NULL,f)
- #define [Bidly_WriteTable](#)(filename, f) [Bidly_Managed_WriteTable](#)(NULL,filename,f)
- #define [Bidly_PrintfSOP](#)(f) [Bidly_Managed_PrintfSOP](#)(NULL,f)
- #define [Bidly_WriteSOP](#)(filename, f) [Bidly_Managed_WriteSOP](#)(NULL,filename,f)
- #define [Bidly_PrintfMinterms](#)(f, negative) [Bidly_Managed_PrintfMinterms](#)(NULL,f,negative)
- #define [Bidly_WriteDot](#)(filename, f, label, id, cudd) [Bidly_Managed_WriteDot](#)(NULL,filename,f,label,id,cudd)
- #define [Bidly_WriteBddview](#)(filename, f, label, table) [Bidly_Managed_WriteBddview](#)(NULL,filename,f,label,table)

Typedefs

- typedef char **Bidly_Boolean**
- typedef char * **Bidly_String**
- typedef void ** **Bidly_Manager**
- typedef void * **Bidly_Cache**
- typedef unsigned short int **Bidly_Variable**
- typedef void * **Bidly_Edge**
- typedef void(* **Bidly_GCFunction**) ([Bidly_Manager](#))
- typedef [Bidly_Boolean](#)(* **Bidly_LookupFunction**) ([Bidly_String](#), [Bidly_Edge](#) *)

Functions

- EXTERN void [Bidly_InitMNG](#) ([Bidly_Manager](#) *mng, int bddtype)
Function Bidly_InitMNG initialize a manager.
- EXTERN void [Bidly_ExitMNG](#) ([Bidly_Manager](#) *mng)
Function Bidly_ExitMNG deletes a manager.
- EXTERN [Bidly_String](#) [Bidly_About](#) ()
Function Bidly_About reports version of Bidly package.
- EXTERN int [Bidly_Managed_GetManagerType](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetManagerType reports BDD type used in the manager.
- EXTERN [Bidly_String](#) [Bidly_Managed_GetManagerName](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetManagerName reports the name of the BDD type used in the manager.
- EXTERN void [Bidly_Managed_SetManagerParameters](#) ([Bidly_Manager](#) MNG, float gcr, float gcrF, float gcrX, float rr, float rrF, float rrX, float st, float cst)
Function Bidly_Managed_SetManagerParameters set modifiable parameters.
- EXTERN [Bidly_Edge](#) [Bidly_GetThen](#) ([Bidly_Edge](#) f)
Function Bidly_GetThen returns THEN successor.
- EXTERN [Bidly_Edge](#) [Bidly_GetElse](#) ([Bidly_Edge](#) f)
Function Bidly_GetElse returns ELSE successor.
- EXTERN [Bidly_Variable](#) [Bidly_GetTopVariable](#) ([Bidly_Edge](#) f)
Function Bidly_GetTopVariable returns the top variable.

- EXTERN [Bidly_Boolean Bidly_Managed_IsEqv](#) ([Bidly_Manager](#) MNG1, [Bidly_Edge](#) f1, [Bidly_Manager](#) MNG2, [Bidly_Edge](#) f2)
 - Function Bidly_Managed_IsEqv returns TRUE iff two BDDs are equal.*
- EXTERN void [Bidly_Managed_SelectNode](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_SelectNode selects the top node of the given function.*
- EXTERN void [Bidly_Managed_DeselectNode](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_DeselectNode deselects the top node of the given function.*
- EXTERN [Bidly_Boolean Bidly_Managed_IsSelected](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_IsSelected returns TRUE iff the top node of the given function is selected.*
- EXTERN void [Bidly_Managed_SelectFunction](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_SelectFunction recursively selects all nodes of a given function.*
- EXTERN void [Bidly_Managed_DeselectAll](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_DeselectAll deselects all nodes.*
- EXTERN [Bidly_Edge Bidly_Managed_GetTerminal](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_GetTerminal returns unmarked and untagged edge pointing to terminal node 1.*
- EXTERN [Bidly_Edge Bidly_Managed_GetConstantZero](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_GetConstantZero returns constant 0.*
- EXTERN [Bidly_Edge Bidly_Managed_GetConstantOne](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_GetConstantOne returns constant 1.*
- EXTERN [Bidly_Edge Bidly_Managed_GetBaseSet](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_GetBaseSet returns set containing only a null combination, i.e. it returns {{}}.*
- EXTERN [Bidly_Variable Bidly_Managed_GetVariable](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x)
 - Function Bidly_Managed_GetVariable returns variable with the given name.*
- EXTERN [Bidly_Variable Bidly_Managed_GetLowestVariable](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_GetLowestVariable returns the lowest variable in the current ordering.*
- EXTERN [Bidly_Variable Bidly_Managed_GetIthVariable](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) i)
 - Function Bidly_Managed_GetIthVariable returns ith variable in the current global ordering.*
- EXTERN [Bidly_Variable Bidly_Managed_GetPrevVariable](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetPrevVariable returns previous variable in the global ordering (lower, topmore).*
- EXTERN [Bidly_Variable Bidly_Managed_GetNextVariable](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetNextVariable returns next variable in the global ordering (higher, bottommore).*
- EXTERN [Bidly_Edge Bidly_Managed_GetVariableEdge](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetVariableEdge returns variable's edge.*
- EXTERN [Bidly_Edge Bidly_Managed_GetElementEdge](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetElementEdge returns element's edge.*
- EXTERN [Bidly_String Bidly_Managed_GetVariableName](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetVariableName returns the name of a variable.*
- EXTERN [Bidly_Edge Bidly_Managed_GetTopVariableEdge](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_GetTopVariableEdge returns variable's edge of top variable.*
- EXTERN [Bidly_String Bidly_Managed_GetTopVariableName](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_GetTopVariableName returns the name of top variable.*
- EXTERN char [Bidly_Managed_GetTopVariableChar](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_GetTopVariableChar returns the first character in the name of top variable.*
- EXTERN void [Bidly_Managed_ResetVariablesValue](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_ResetVariablesValue sets all variable's value to biddyZero.*
- EXTERN void [Bidly_Managed_SetVariableValue](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v, [Bidly_Edge](#) f)
 - Function Bidly_Managed_SetVariableValue sets variable's value.*
- EXTERN [Bidly_Edge Bidly_Managed_GetVariableValue](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetVariableValue gets variable's value.*
- EXTERN void [Bidly_Managed_ClearVariablesData](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_ClearVariablesData free memory used for all variable's data.*

- EXTERN void [Biddy_Managed_SetVariableData](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v, void *x)
Function Biddy_Managed_SetVariableData sets variable's data.
 - EXTERN void * [Biddy_Managed_GetVariableData](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v)
Function Biddy_Managed_GetVariableData gets variable's data.
 - EXTERN [Biddy_Boolean](#) [Biddy_Managed_Eval](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_Eval returns the value of a Boolean function for a given variable assignment.
- Description**
- Side effects**
- Variables must have values assigned. Variable is considered to be FALSE iff variable.value == biddyZero, otherwise it is considered to be TRUE.*
- More info**
- Macro Biddy_Eval(f) is defined for use with anonymous manager.*
- EXTERN double [Biddy_Managed_EvalProbability](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_EvalProbability evaluates BDD.
 - EXTERN [Biddy_Boolean](#) [Biddy_Managed_IsSmaller](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) fv, [Biddy_↔Variable](#) gv)
Function Biddy_Managed_IsSmaller returns TRUE if the first variable is smaller (= lower = previous = above = top-more).
 - EXTERN [Biddy_Boolean](#) [Biddy_Managed_IsLowest](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v)
Function Biddy_Managed_IsLowest returns TRUE if the variable is the lowest one (lowest == topmost).
 - EXTERN [Biddy_Boolean](#) [Biddy_Managed_IsHighest](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v)
Function Biddy_Managed_IsHighest returns TRUE if the variable is the highest one if terminal node is ignored (highest == bottommost).
 - EXTERN [Biddy_Variable](#) [Biddy_Managed_FoaVariable](#) ([Biddy_Manager](#) MNG, [Biddy_String](#) x, [Biddy_↔Boolean](#) varelem)
Function Biddy_Managed_FoaVariable finds variable/element or adds new variable (i.e. Boolean function f = x) and new element (i.e. it creates set {{x}}).
 - EXTERN void [Biddy_Managed_ChangeVariableName](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v, [Biddy_↔String](#) x)
Function Biddy_Managed_ChangeVariableName set new name to the given variable/element.
 - EXTERN [Biddy_Variable](#) [Biddy_Managed_AddVariableByName](#) ([Biddy_Manager](#) MNG, [Biddy_String](#) x)
Function Biddy_Managed_AddVariableByName adds variable.
 - EXTERN [Biddy_Variable](#) [Biddy_Managed_AddElementByName](#) ([Biddy_Manager](#) MNG, [Biddy_String](#) x)
Function Biddy_Managed_AddElementByName adds element.
 - EXTERN [Biddy_Edge](#) [Biddy_Managed_AddVariableBelow](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v)
Function Biddy_Managed_AddVariableBelow adds variable.
 - EXTERN [Biddy_Edge](#) [Biddy_Managed_AddVariableAbove](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v)
Function Biddy_Managed_AddVariableAbove adds variable.
 - EXTERN [Biddy_Edge](#) [Biddy_Managed_TransferMark](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Boolean](#) mark, [Biddy_Boolean](#) leftright)
Function Biddy_Managed_TransferMark returns edge with inverted complement bit iff the second parameter is TRUE and normalization rules require this.
 - EXTERN [Biddy_Edge](#) [Biddy_Managed_IncTag](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_IncTag returns edge with an incremented tag.
 - EXTERN [Biddy_Edge](#) [Biddy_Managed_TaggedFoaNode](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v, [Biddy_↔Edge](#) pf, [Biddy_Edge](#) pt, [Biddy_Variable](#) ptag, [Biddy_Boolean](#) garbageAllowed)
Function Biddy_Managed_TaggedFoaNode finds or adds new node with the given variable and successors.
 - EXTERN [Biddy_Boolean](#) [Biddy_Managed_IsOK](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_IsOK returns TRUE iff given node is not obsolete.
 - EXTERN void [Biddy_Managed_GC](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) targetLT, [Biddy_Variable](#) target↔GEQ, [Biddy_Boolean](#) purge, [Biddy_Boolean](#) total)
Function Biddy_Managed_GC performs garbage collection.

- EXTERN void [Bidly_Managed_Clean](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_Clean performs cleaning.
- EXTERN void [Bidly_Managed_Purge](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_Purge immediately removes all nodes which were not preserved or which are not preserved anymore.
- EXTERN void [Bidly_Managed_PurgeAndReorder](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Boolean](#) converge)

Function Bidly_Managed_PurgeAndReorder immediately removes non-preserved nodes and triggers reordering on function.
- EXTERN void [Bidly_Managed_Refresh](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)

Function Bidly_Managed_Refresh refreshes top node in a given function.
- EXTERN void [Bidly_Managed_AddCache](#) ([Bidly_Manager](#) MNG, [Bidly_GCFunction](#) gc)

Function Bidly_Managed_AddCache adds cache to the end of Cache list.
- EXTERN unsigned int [Bidly_Managed_AddFormula](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x, [Bidly_Edge](#) f, int c)

Function Bidly_Managed_AddFormula adds formula to Formula table.
- EXTERN [Bidly_Boolean](#) [Bidly_Managed_FindFormula](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x, unsigned int *idx, [Bidly_Edge](#) *f)

Function Bidly_Managed_FindFormula find formula in Formula table.
- EXTERN [Bidly_Boolean](#) [Bidly_Managed_DeleteFormula](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x)

Function Bidly_Managed_DeleteFormula delete formula from Formula table.
- EXTERN [Bidly_Boolean](#) [Bidly_Managed_DeletelthFormula](#) ([Bidly_Manager](#) MNG, unsigned int i)

Function Bidly_Managed_DeletelthFormula deletes formula from the table.
- EXTERN [Bidly_Edge](#) [Bidly_Managed_GetlthFormula](#) ([Bidly_Manager](#) MNG, unsigned int i)

Function Bidly_Managed_GetlthFormula returns ith formula in a Formula table.
- EXTERN [Bidly_String](#) [Bidly_Managed_GetlthFormulaName](#) ([Bidly_Manager](#) MNG, unsigned int i)

Function Bidly_Managed_GetlthFormulaName returns name of the ith formula in a Formula table.
- EXTERN [Bidly_String](#) [Bidly_Managed_GetOrdering](#) ([Bidly_Manager](#) MNG)
- EXTERN void [Bidly_Managed_SetOrdering](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) ordering)

Function Bidly_Managed_SetOrdering use variable swapping to create the ordering given by string.
- EXTERN void [Bidly_Managed_SetAlphabeticOrdering](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_SetAlphabeticOrdering use variable swapping to create the alphabetic ordering.
- EXTERN [Bidly_Variable](#) [Bidly_Managed_SwapWithHigher](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)

Function Bidly_Managed_SwapWithHigher swaps two adjacent variables.
- EXTERN [Bidly_Variable](#) [Bidly_Managed_SwapWithLower](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)

Function Bidly_Managed_SwapWithLower swaps two adjacent variables.
- EXTERN [Bidly_Boolean](#) [Bidly_Managed_Sifting](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Boolean](#) converge)

Function Bidly_Managed_Sifting reorders variables to minimize node number using Rudell's sifting algorithm for the whole system (if f = NULL) or for the given function (if f != NULL) .
- EXTERN void [Bidly_Managed_MinimizeBDD](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) name)

Function Bidly_Managed_MinimizeBDD reorders variables to minimize the node number of the given formula using an exhaustive search over all possible orderings.
- EXTERN void [Bidly_Managed_MaximizeBDD](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) name)

Function Bidly_Managed_MaximizeBDD reorders variables to maximize the node number of the given function using an exhaustive search over all possible orderings.
- EXTERN [Bidly_Edge](#) [Bidly_Managed_Copy](#) ([Bidly_Manager](#) MNG1, [Bidly_Manager](#) MNG2, [Bidly_Edge](#) f)

Function Bidly_Managed_Copy copies a graph from one manager to another manager which can use the same or different BDD type.

Description

The function takes a graph from one manager and creates the same graph in another manager. If the managers do not use the same BDD type then a graph is converted. The resulting graph will represent the same Boolean function assuming the domain from the target manager. If ($f == \text{bidlyZero}$) then only the complete domain (all variables) is copied.

Side effects

If source and target manager are the same then function does nothing. The variable ordering of the created BDD is trying to follow the original ordering, but if some variables already exist in the target manager then the final ordering is adapted to the target manager. Please note, that indices of variables in the target manager may not be the same as in the source manager (for example, if source manager does not use initial ordering the indices in the target manager will follow the variable's ordering and not variable's original indices)

More info

Macro `Bidly_Copy(MNG2,f)` is defined for use with anonymous manager.

- EXTERN void `Bidly_Managed_CopyFormula` (`Bidly_Manager` MNG1, `Bidly_Manager` MNG2, `Bidly_String` x)

Function `Bidly_Managed_CopyFormula` uses `Bidly_Managed_Copy` to copy a graph from one manager to another manager which can use the same or different BDD type.

Description

See `Bidly_Managed_Copy`.

Side effects

If source and target manager are the same then function does nothing. The variable ordering of created BDD is adapted to the target manager. The created formula is refreshed but not preserved.

More info

Macro `Bidly_CopyFormula(MNG2,x)` is defined for use with anonymous manager.

- EXTERN `Bidly_Edge Bidly_Managed_ConstructBDD` (`Bidly_Manager` MNG, int numV, `Bidly_String` varlist, int numN, `Bidly_String` nodelist)
- EXTERN `Bidly_Edge Bidly_Managed_Not` (`Bidly_Manager` MNG, `Bidly_Edge` f)
- EXTERN `Bidly_Edge Bidly_Managed_ITE` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g, `Bidly_Edge` h)

Function `Bidly_Managed_ITE` calculates ITE operation of three Boolean functions.

- EXTERN `Bidly_Edge Bidly_Managed_And` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Or` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Nand` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Nor` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Xor` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Xnor` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Leq` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Gt` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Boolean Bidly_Managed_IsLeq` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Edge` g)
- EXTERN `Bidly_Edge Bidly_Managed_Restrict` (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Variable` v, `Bidly_Boolean` value)

Function `Bidly_Managed_Restrict` calculates a restriction of Boolean function.

- EXTERN [Bidly_Edge Bidly_Managed_Compose](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g, [Bidly_Variable](#) v)
Function Bidly_Managed_Compose calculates a composition of two Boolean functions.
- EXTERN [Bidly_Edge Bidly_Managed_E](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v)
Function Bidly_Managed_E calculates an existential quantification of Boolean function.
- EXTERN [Bidly_Edge Bidly_Managed_A](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v)
Function Bidly_Managed_A calculates an universal quantification of Boolean function.
- EXTERN [Bidly_Boolean Bidly_Managed_IsVariableDependent](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v)
Function Bidly_Managed_IsVariableDependent returns TRUE iff variable is dependent on others in a function.
- EXTERN [Bidly_Edge Bidly_Managed_ExistAbstract](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) cube)
Function Bidly_Managed_ExistAbstract existentially abstracts all the variables in cube from f.
- EXTERN [Bidly_Edge Bidly_Managed_UnivAbstract](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) cube)
Function Bidly_Managed_UnivAbstract universally abstracts all the variables in cube from f.
- EXTERN [Bidly_Edge Bidly_Managed_AndAbstract](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g, [Bidly_Edge](#) cube)
Function Bidly_Managed_AndAbstract calculates the AND of two BDDs and simultaneously (existentially) abstracts the variables in cube.
- EXTERN [Bidly_Edge Bidly_Managed_Constrain](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) c)
Function Bidly_Managed_Constrain calculates Coudert and Madre's constrain function.
- EXTERN [Bidly_Edge Bidly_Managed_Simplify](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) c)
Function Bidly_Managed_Simplify calculates (a slightly) modified Coudert and Madre's restrict function.
- EXTERN [Bidly_Edge Bidly_Managed_Support](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_Support calculates a product of all dependent variables (OBDDs and TZBDDs) or the combination set containing a single subset which includes all dependent variables (ZBDDs).
- EXTERN [Bidly_Edge Bidly_Managed_ReplaceByKeyword](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_↔String](#) keyword)
Function Bidly_Managed_ReplaceByKeyword calculates Boolean function with one or more variables replaced.
- EXTERN [Bidly_Edge Bidly_Managed_Change](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v)
Function Bidly_Managed_Change change the form of the given variable (positive literal becomes negative and vice versa).
- EXTERN [Bidly_Edge Bidly_Managed_VarSubset](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v, [Bidly_Boolean](#) value)
Function Bidly_Managed_VarSubset calculates a division of Boolean function with a literal.
- EXTERN [Bidly_Edge Bidly_Managed_ElementAbstract](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_↔Variable](#) v)
Function Bidly_Managed_ElementAbstract remove element from all combinations in the set.
- EXTERN [Bidly_Edge Bidly_Managed_Product](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Product calculates operation product defined over combination sets.
- EXTERN [Bidly_Edge Bidly_Managed_SelectiveProduct](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g, [Bidly_Edge](#) cube)
Function Bidly_Managed_SelectiveProduct calculates operation selective product defined over combination sets.
- EXTERN [Bidly_Edge Bidly_Managed_Supset](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Supset calculates Coudert and Madre's operation SupSet.
- EXTERN [Bidly_Edge Bidly_Managed_Subset](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Subset calculates Coudert and Madre's operation SubSet.
- EXTERN [Bidly_Edge Bidly_Managed_Permitsym](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, unsigned int n)
Function Bidly_Managed_Permitsym return a subset of f where only combinations with up to n elements are included.
- EXTERN [Bidly_Edge Bidly_Managed_Stretch](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_Stretch calculates minimal combination set such that all elements in the original set has at least one superset in the new set.

- EXTERN [Biddy_Edge Biddy_Managed_CreateMinterm](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, long long unsigned int x)
Function Biddy_Managed_CreateMinterm generates one minterm.
- EXTERN [Biddy_Edge Biddy_Managed_CreateFunction](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, long long unsigned int x)
Function Biddy_Managed_CreateFunction generates one Boolean function.
- EXTERN [Biddy_Edge Biddy_Managed_RandomFunction](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, double r)
Function Biddy_Managed_RandomFunction generates a random BDD.
- EXTERN [Biddy_Edge Biddy_Managed_RandomSet](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) unit, double r)
Function Biddy_Managed_RandomSet generates a random BDD.
- EXTERN [Biddy_Edge Biddy_Managed_ExtractMinterm](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, [Biddy_Edge](#) f)
Function Biddy_Managed_ExtractMinterm ...
- EXTERN unsigned int [Biddy_Managed_CountNodes](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_CountNodes.
- EXTERN unsigned int [Biddy_MaxLevel](#) ([Biddy_Edge](#) f)
Function Biddy_MaxLevel.
- EXTERN float [Biddy_AvgLevel](#) ([Biddy_Edge](#) f)
Function Biddy_AvgLevel.
- EXTERN [Biddy_Variable Biddy_Managed_VariableTableNum](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_VariableTableNum returns number of used variables.
- EXTERN unsigned int [Biddy_Managed_NodeTableSize](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableSize returns the size of node table.
- EXTERN unsigned int [Biddy_Managed_NodeTableBlockNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableBlockNumber.
- EXTERN unsigned int [Biddy_Managed_NodeTableGenerated](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableGenerated.
- EXTERN unsigned int [Biddy_Managed_NodeTableMax](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableMax returns maximal (peek) number of nodes in node table.
- EXTERN unsigned int [Biddy_Managed_NodeTableNum](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableNum returns number of all nodes currently in node table.
- EXTERN unsigned int [Biddy_Managed_NodeTableNumVar](#) ([Biddy_Manager](#) MNG, [Biddy_Variable](#) v)
Function Biddy_Managed_NodeTableNumVar returns number of nodes with a given variable currently in node table.
- EXTERN unsigned int [Biddy_Managed_NodeTableResizeNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableResizeNumber.
- EXTERN unsigned long long int [Biddy_Managed_NodeTableFoaNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableFoaNumber.
- EXTERN unsigned long long int [Biddy_Managed_NodeTableFindNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableFindNumber.
- EXTERN unsigned long long int [Biddy_Managed_NodeTableCompareNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableCompareNumber.
- EXTERN unsigned long long int [Biddy_Managed_NodeTableAddNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableAddNumber.
- EXTERN unsigned int [Biddy_Managed_NodeTableGCNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableGCNumber.
- EXTERN unsigned int [Biddy_Managed_NodeTableGCTime](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableGCTime.
- EXTERN unsigned long long int [Biddy_Managed_NodeTableGCObsoleteNumber](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_NodeTableGCObsoleteNumber.
- EXTERN unsigned int [Biddy_Managed_NodeTableSwapNumber](#) ([Biddy_Manager](#) MNG)

- Function Bidly_Managed_NodeTableSwapNumber.*

 - EXTERN unsigned int [Bidly_Managed_NodeTableSiftingNumber](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableSiftingNumber.
- EXTERN unsigned int [Bidly_Managed_NodeTableDRTime](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableDRTime.
- EXTERN unsigned int [Bidly_Managed_NodeTableITENumber](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableITENumber.
- EXTERN unsigned long long int [Bidly_Managed_NodeTableITERecursiveNumber](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableITERecursiveNumber.
- EXTERN unsigned int [Bidly_Managed_NodeTableANDORNumber](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableANDORNumber.
- EXTERN unsigned long long int [Bidly_Managed_NodeTableANDORRecursiveNumber](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableANDORRecursiveNumber.
- EXTERN unsigned int [Bidly_Managed_NodeTableXORNumber](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableXORNumber.
- EXTERN unsigned long long int [Bidly_Managed_NodeTableXORRecursiveNumber](#) (Bidly_Manager MNG)

Function Bidly_Managed_NodeTableXORRecursiveNumber.
- EXTERN unsigned int [Bidly_Managed_FormulaTableNum](#) (Bidly_Manager MNG)

Function Bidly_Managed_FormulaTableNum returns number of known formulae.
- EXTERN unsigned int [Bidly_Managed_ListUsed](#) (Bidly_Manager MNG)

Function Bidly_Managed_ListUsed.
- EXTERN unsigned int [Bidly_Managed_ListMaxLength](#) (Bidly_Manager MNG)

Function Bidly_Managed_ListMaxLength.
- EXTERN float [Bidly_Managed_ListAvgLength](#) (Bidly_Manager MNG)

Function Bidly_Managed_ListAvgLength.
- EXTERN unsigned long long int [Bidly_Managed_OPCCacheSearch](#) (Bidly_Manager MNG)

Function Bidly_Managed_OPCCacheSearch.
- EXTERN unsigned long long int [Bidly_Managed_OPCCacheFind](#) (Bidly_Manager MNG)

Function Bidly_Managed_OPCCacheFind.
- EXTERN unsigned long long int [Bidly_Managed_OPCCacheInsert](#) (Bidly_Manager MNG)

Function Bidly_Managed_OPCCacheInsert.
- EXTERN unsigned long long int [Bidly_Managed_OPCCacheOverwrite](#) (Bidly_Manager MNG)

Function Bidly_Managed_OPCCacheOverwrite.
- EXTERN unsigned int [Bidly_Managed_CountNodesPlain](#) (Bidly_Manager MNG, Bidly_Edge f)

Function Bidly_Managed_CountNodesPlain.
- EXTERN unsigned int [Bidly_Managed_DependentVariableNumber](#) (Bidly_Manager MNG, Bidly_Edge f, Bidly_Boolean select)

Function Bidly_Managed_DependentVariableNumber.
- EXTERN unsigned int [Bidly_Managed_CountComplementedEdges](#) (Bidly_Manager MNG, Bidly_Edge f)

Function Bidly_Managed_CountComplementedEdges count the number of complemented edges.
- EXTERN unsigned long long int [Bidly_Managed_CountPaths](#) (Bidly_Manager MNG, Bidly_Edge f)

Function Bidly_Managed_CountPaths count the number of 1-paths.
- EXTERN double [Bidly_Managed_CountMinterms](#) (Bidly_Manager MNG, Bidly_Edge f, unsigned int nvars)

Function Bidly_Managed_CountMinterms.
- EXTERN double [Bidly_Managed_DensityOfFunction](#) (Bidly_Manager MNG, Bidly_Edge f, unsigned int nvars)

Function Bidly_Managed_DensityOfFunction calculates the ratio of the number of on-set minterms to the number of all minterms.
- EXTERN double [Bidly_Managed_DensityOfBDD](#) (Bidly_Manager MNG, Bidly_Edge f, unsigned int nvars)

Function Bidly_Managed_DensityOfBDD calculates the ratio of the number of on-set minterms to the number of nodes.

- EXTERN unsigned int [Biddy_Managed_MinNodes](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_MinNodes reports number of nodes in the optimal ordering.
- EXTERN unsigned int [Biddy_Managed_MaxNodes](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_MaxNodes reports number of nodes in the worst ordering.
- EXTERN unsigned long long int [Biddy_Managed_ReadMemoryInUse](#) ([Biddy_Manager](#) MNG)
Function Biddy_Managed_ReadMemoryInUse reports memory consumption of main data structures in bytes (nodes, node table, variable table, ordering table, formula table, ITE cache, EA cache, RC cache, REPLACEcache).
- EXTERN void [Biddy_Managed_PrintInfo](#) ([Biddy_Manager](#) MNG, FILE *f)
Function Biddy_Managed_PrintInfo prepares a file with stats.
- EXTERN [Biddy_String](#) [Biddy_Managed_Eval0](#) ([Biddy_Manager](#) MNG, [Biddy_String](#) s)
Function Biddy_Managed_Eval0 evaluates raw format.
- EXTERN [Biddy_Edge](#) [Biddy_Managed_Eval1x](#) ([Biddy_Manager](#) MNG, [Biddy_String](#) s, [Biddy_Lookup](#)↔
[Function](#) lf)
Function Biddy_Managed_Eval1x evaluates prefix AND-OR-EXOR-NOT format.
- EXTERN [Biddy_Edge](#) [Biddy_Managed_Eval2](#) ([Biddy_Manager](#) MNG, [Biddy_String](#) boolFunc)
Function Biddy_Managed_Eval2 evaluates infix format.
- EXTERN [Biddy_String](#) [Biddy_Managed_ReadBddview](#) ([Biddy_Manager](#) MNG, const char filename[],
[Biddy_String](#) name)
Function Biddy_Managed_ReadBddview reads bddview file and creates a Boolean function.
- EXTERN void [Biddy_Managed_ReadVerilogFile](#) ([Biddy_Manager](#) MNG, const char filename[], [Biddy_String](#)
prefix)
Function Biddy_Managed_ReadVerilogFile reads Verilog file and creates variables for all primary inputs and Boolean functions for all primary outputs.
- EXTERN void [Biddy_Managed_PrintfBDD](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_PrintfBDD writes raw format using printf.
- EXTERN void [Biddy_Managed_WriteBDD](#) ([Biddy_Manager](#) MNG, const char filename[], [Biddy_Edge](#) f,
[Biddy_String](#) label)
Function Biddy_Managed_WriteBDD writes raw format using fprintf.
- EXTERN void [Biddy_Managed_PrintfTable](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_PrintfTable writes truth table using printf.
- EXTERN void [Biddy_Managed_WriteTable](#) ([Biddy_Manager](#) MNG, const char filename[], [Biddy_Edge](#) f)
Function Biddy_Managed_WriteTable writes truth table using fprintf.
- EXTERN void [Biddy_Managed_PrintfSOP](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_PrintfSOP writes SOP using printf.
- EXTERN void [Biddy_Managed_WriteSOP](#) ([Biddy_Manager](#) MNG, const char filename[], [Biddy_Edge](#) f)
Function Biddy_Managed_WriteSOP writes SOP using fprintf.
- EXTERN void [Biddy_Managed_PrintfMinterms](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Boolean](#) neg-
ative)
Function Biddy_Managed_PrintfMinterms writes minterms using printf.
- EXTERN unsigned int [Biddy_Managed_WriteDot](#) ([Biddy_Manager](#) MNG, const char filename[], [Biddy_Edge](#)
f, const char label[], int id, [Biddy_Boolean](#) cudd)
Function Biddy_Managed_WriteDot writes dot/graphviz format using fprintf.
- EXTERN unsigned int [Biddy_Managed_WriteBddview](#) ([Biddy_Manager](#) MNG, const char filename[], [Biddy](#)↔
[_Edge](#) f, const char label[], void *xytable)
Function Biddy_Managed_WriteBDDView writes bddview format using fprintf.

3.1.1 Detailed Description

Description

PackageName [Biddy]

Synopsis [Biddy provides data structures and algorithms for the representation and manipulation of Boolean functions with

ROBDDs, 0-sup-BDDs, and TZBDDs. A hash table is used for quick search of nodes. Complement edges decreases the number of nodes. An automatic garbage collection with a system age is implemented. Variable swapping and sifting are implemented.]

```
FileName      [biddy.h]
Revision      [${Revision: 545 $}]
Date          [${Date: 2019-02-11 14:07:50 +0100 (pon, 11 feb 2019) $}]
Authors       [Robert Meolic (robert@meolic.com),
              Ales Casar (ales@homemade.net)]
```

Copyright

Copyright (C) 2006, 2019 UM FERi, Koroska cesta 46, SI-2000 Maribor, Slovenia

Biddy is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Biddy is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

More info

See also: [biddyInt.h](#)

3.1.2 Macro Definition Documentation

3.1.2.1 Biddy_A

```
#define Biddy_A(  
    f,  
    v ) Biddy_Managed_A (NULL, f, v)
```

Macro Biddy_A is defined for use with anonymous manager.

Definition at line 808 of file biddy.h.

3.1.2.2 Biddy_AddCache

```
#define Biddy_AddCache(  
    gc ) Biddy_Managed_AddCache (NULL, gc)
```

Macro Biddy_AddCache is defined for use with anonymous manager.

Definition at line 608 of file biddy.h.

3.1.2.3 Biddy_AddElementByName

```
#define Biddy_AddElementByName(  
    x ) Biddy_Managed_AddElementByName (NULL, x)
```

Macro Biddy_AddElementByName is defined for use with anonymous manager.

Definition at line 538 of file biddy.h.

3.1.2.4 Biddy_AddFormula

```
#define Biddy_AddFormula(  
    x,  
    f,  
    c ) Biddy_Managed_AddFormula (NULL, x, f, c)
```

Macro Biddy_AddFormula is defined for use with anonymous manager.

Macros Biddy_Managed_AddTmpFormula, Biddy_AddTmpFormula, Biddy_Managed_AddPersistentFormula, Biddy_AddPersistentFormula, Biddy_Managed_KeepFormula, Biddy_KeepFormula, Biddy_Managed_KeepFormulaProlonged, Biddy_KeepFormulaProlonged, Biddy_Managed_KeepFormulaUntilPurge, and Biddy_KeepFormulaUntilPurge are defined to simplify formulae management.

Definition at line 619 of file biddy.h.

3.1.2.5 Biddy_AddVariableAbove

```
#define Biddy_AddVariableAbove(  
    v ) Biddy_Managed_AddVariableAbove (NULL, v)
```

Macro Biddy_AddVariableAbove is defined for use with anonymous manager.

Definition at line 550 of file biddy.h.

3.1.2.6 Biddy_AddVariableBelow

```
#define Biddy_AddVariableBelow(  
    v ) Biddy_Managed_AddVariableBelow (NULL, v)
```

Macro Biddy_AddVariableBelow is defined for use with anonymous manager.

Definition at line 545 of file biddy.h.

3.1.2.7 Biddy_AddVariableByName

```
#define Biddy_AddVariableByName(  
    x ) Biddy_Managed_AddVariableByName (NULL, x)
```

Macro Biddy_AddVariableByName is defined for use with anonymous manager.

Macros Biddy_Managed_AddVariable(MNG) and Biddy_AddVariable() are defined for creating numbered variables.

Macros Biddy_Managed_AddVariableEdge(MNG) and Biddy_AddVariableEdge() also create numbered variables but return the variable edge.

Definition at line 529 of file biddy.h.

3.1.2.8 Biddy_And

```
#define Biddy_And(  
    f,  
    g ) Biddy_Managed_And (NULL, f, g)
```

Macro Biddy_And is defined for use with anonymous manager.

Macros Biddy_Managed_Intersect and Biddy_Intersect are defined for set manipulation.

Definition at line 740 of file biddy.h.

3.1.2.9 Biddy_AndAbstract

```
#define Biddy_AndAbstract(  
    f,  
    g,  
    cube ) Biddy_Managed_AndAbstract (NULL, f, g, cube)
```

Macro Biddy_AndAbstract is defined for use with anonymous manager.

Definition at line 828 of file biddy.h.

3.1.2.10 Biddy_Change

```
#define Biddy_Change(  
    f,  
    v ) Biddy_Managed_Change (NULL, f, v)
```

Macro Biddy_Change is defined for use with anonymous manager.

Definition at line 858 of file biddy.h.

3.1.2.11 Biddy_ChangeVariableName

```
#define Biddy_ChangeVariableName(  
    v,  
    x ) Biddy_Managed_ChangeVariableName (NULL, v, x)
```

Macro Biddy_ChangeVariableName is defined for use with anonymous manager.

Definition at line 522 of file biddy.h.

3.1.2.12 Biddy_Clean

```
#define Biddy_Clean( ) Biddy_Managed_Clean (NULL)
```

Macro Biddy_Clean is defined for use with anonymous manager.

Definition at line 588 of file biddy.h.

3.1.2.13 Biddy_ClearMark

```
#define Biddy_ClearMark(  
    f ) (f = (Biddy_Edge) ((uintptr_t) f & ~((uintptr_t) 1)))
```

Biddy_ClearMark makes given edge not-complemented.

Definition at line 174 of file biddy.h.

3.1.2.14 Biddy_ClearTag

```
#define Biddy_ClearTag(  
    f ) 0
```

Biddy_ClearTag removes tag from the given edge, since Biddy v1.7.

Definition at line 216 of file biddy.h.

3.1.2.15 Biddy_ClearVariablesData

```
#define Biddy_ClearVariablesData( ) Biddy_Managed_ClearVariablesData (NULL)
```

Macro Biddy_ClearVariablesData is defined for use with anonymous manager.

Definition at line 477 of file biddy.h.

3.1.2.16 Biddy_Complement

```
#define Biddy_Complement(  
    f ) ((Biddy_Edge) ((uintptr_t) f | (uintptr_t) 1))
```

Biddy_Complement returns complemented version of edge, since Biddy v1.4.

Definition at line 190 of file biddy.h.

3.1.2.17 Biddy_Compose

```
#define Biddy_Compose(  
    f,  
    g,  
    v ) Biddy_Managed_Compose(NULL, f, g, v)
```

Macro Biddy_Compose is defined for use with anonymous manager.

Definition at line 798 of file biddy.h.

3.1.2.18 Biddy_Constrain

```
#define Biddy_Constrain(  
    f,  
    c ) Biddy_Managed_Constrain(NULL, f, c)
```

Macro Biddy_Constrain is defined for use with anonymous manager.

Definition at line 833 of file biddy.h.

3.1.2.19 Biddy_ConstructBDD

```
#define Biddy_ConstructBDD(  
    numV,  
    varlist,  
    numN,  
    nodelist ) Biddy_Managed_ConstructBDD(NULL, numV, varlist, numV, nodelist)
```

Macro Biddy_ConstructBDD is defined for use with anonymous manager.

Definition at line 711 of file biddy.h.

3.1.2.20 **Bidly_Copy**

```
#define Bidly_Copy(  
    MNG2,  
    f ) Bidly_Managed_Copy (NULL, MNG2, f)
```

Macro `Bidly_Copy` is defined for use with anonymous manager.

Definition at line 699 of file `bidly.h`.

3.1.2.21 **Bidly_CopyFormulaFrom**

```
#define Bidly_CopyFormulaFrom(  
    MNG1,  
    x ) Bidly_Managed_CopyFormula (MNG1, NULL, x)
```

Macros `Bidly_CopyFormulaFrom` and `Bidly_CopyFormulaTo` are defined for use with anonymous manager.

Definition at line 705 of file `bidly.h`.

3.1.2.22 **Bidly_CountComplementedEdges**

```
#define Bidly_CountComplementedEdges(  
    f ) Bidly_Managed_CountComplementedEdges (NULL, f)
```

Macro `Bidly_CountComplementedEdges` is defined for use with anonymous manager.

Definition at line 1144 of file `bidly.h`.

3.1.2.23 **Bidly_CountMinterms**

```
#define Bidly_CountMinterms(  
    f,  
    nvars ) Bidly_Managed_CountMinterms (NULL, f, nvars)
```

Macro `Bidly_CountMinterms` is defined for use with anonymous manager.

Definition at line 1154 of file `bidly.h`.

3.1.2.24 **Bidly_CountNodes**

```
#define Bidly_CountNodes(  
    f ) Bidly_Managed_CountNodes (NULL, f)
```

Macro `Bidly_CountNodes(f)` is defined for use with anonymous manager.

Definition at line 959 of file `bidly.h`.

3.1.2.25 Biddy_CountNodesPlain

```
#define Biddy_CountNodesPlain(  
    f ) Biddy_Managed_CountNodesPlain(NULL, f)
```

Macro Biddy_CountNodesPlain is defined for use with anonymous manager.

Definition at line 1134 of file biddy.h.

3.1.2.26 Biddy_CountPaths

```
#define Biddy_CountPaths(  
    f ) Biddy_Managed_CountPaths(NULL, f)
```

Macro Biddy_CountPaths is defined for use with anonymous manager.

Definition at line 1149 of file biddy.h.

3.1.2.27 Biddy_CreateFunction

```
#define Biddy_CreateFunction(  
    support,  
    x ) Biddy_Managed_CreateFunction(NULL, support, x)
```

Macro Biddy_CreateFunction is defined for use with anonymous manager.

Definition at line 923 of file biddy.h.

3.1.2.28 Biddy_CreateMinterm

```
#define Biddy_CreateMinterm(  
    support,  
    x ) Biddy_Managed_CreateMinterm(NULL, support, x)
```

Macro Biddy_CreateMinterm is defined for use with anonymous manager.

Definition at line 918 of file biddy.h.

3.1.2.29 Biddy_DeleteFormula

```
#define Biddy_DeleteFormula(  
    x ) Biddy_Managed_DeleteFormula(NULL, x)
```

Macro Biddy_DeleteFormula is defined for use with anonymous manager.

Definition at line 639 of file biddy.h.

3.1.2.30 **Bidly_DeleteIthFormula**

```
#define Bidly_DeleteIthFormula(  
    x ) Bidly_Managed_DeleteIthFormula (NULL, x)
```

Macro Bidly_DeleteIthFormula is defined for use with anonymous manager.

Definition at line 644 of file bidly.h.

3.1.2.31 **Bidly_DensityOfBDD**

```
#define Bidly_DensityOfBDD(  
    f,  
    nvars ) Bidly_Managed_DensityOfBDD (NULL, f, nvars)
```

Macro Bidly_DensityOfBDD is defined for use with anonymous manager.

Definition at line 1166 of file bidly.h.

3.1.2.32 **Bidly_DensityOfFunction**

```
#define Bidly_DensityOfFunction(  
    f,  
    nvars ) Bidly_Managed_DensityOfFunction (NULL, f, nvars)
```

Macro Bidly_DensityOfFunction is defined for use with anonymous manager.

Definition at line 1161 of file bidly.h.

3.1.2.33 **Bidly_DependentVariableNumber**

```
#define Bidly_DependentVariableNumber(  
    f,  
    select ) Bidly_Managed_DependentVariableNumber (NULL, f, select)
```

Macro Bidly_DependentVariableNumber is defined for use with anonymous manager.

Definition at line 1139 of file bidly.h.

3.1.2.34 **Bidly_DeselectAll**

```
#define Bidly_DeselectAll( ) Bidly_Managed_DeselectAll (NULL)
```

Macro Bidly_DeselectAll is defined for use with anonymous manager.

Definition at line 378 of file bidly.h.

3.1.2.35 Biddy_DeselectNode

```
#define Biddy_DeselectNode(  
    f ) Biddy_Managed_DeselectNode (NULL, f)
```

Macro Biddy_DeselectNode is defined for use with anonymous manager.

Definition at line 363 of file biddy.h.

3.1.2.36 Biddy_E

```
#define Biddy_E(  
    f,  
    v ) Biddy_Managed_E (NULL, f, v)
```

Macro Biddy_E is defined for use with anonymous manager.

Definition at line 803 of file biddy.h.

3.1.2.37 Biddy_ElementAbstract

```
#define Biddy_ElementAbstract(  
    f,  
    v ) Biddy_Managed_ElementAbstract (NULL, f, v)
```

Macro Biddy_ElementAbstract is defined for use with anonymous manager.

Definition at line 881 of file biddy.h.

3.1.2.38 Biddy_Eval

```
#define Biddy_Eval(  
    f ) Biddy_Managed_Eval (NULL, f)
```

Macro Biddy_Eval is defined for use with anonymous manager.

Definition at line 492 of file biddy.h.

3.1.2.39 Biddy_Eval0

```
#define Biddy_Eval0(  
    s ) Biddy_Managed_Eval0 (NULL, s)
```

Macro Biddy_Eval0 is defined for use with anonymous manager.

Definition at line 1203 of file biddy.h.

3.1.2.40 **Bidly_Eval1x**

```
#define Bidly_Eval1x(  
    s,  
    lf ) Bidly_Managed_Eval1x (NULL, s, lf)
```

Macro Bidly_Eval1x is defined for use with anonymous manager.

Definition at line 1208 of file bidly.h.

3.1.2.41 **Bidly_Eval2**

```
#define Bidly_Eval2(  
    boolFunc ) Bidly_Managed_Eval2 (NULL, boolFunc)
```

Macro Bidly_Eval2 is defined for use with anonymous manager.

Definition at line 1215 of file bidly.h.

3.1.2.42 **Bidly_EvalProbability**

```
#define Bidly_EvalProbability(  
    f ) Bidly_Managed_EvalProbability (NULL, f)
```

Macro Bidly_EvalProbability is defined for use with anonymous manager.

Definition at line 497 of file bidly.h.

3.1.2.43 **Bidly_ExistAbstract**

```
#define Bidly_ExistAbstract(  
    f,  
    cube ) Bidly_Managed_ExistAbstract (NULL, f, cube)
```

Macro Bidly_ExistAbstract is defined for use with anonymous manager.

Definition at line 818 of file bidly.h.

3.1.2.44 **Bidly_Exit**

```
#define Bidly_Exit( ) Bidly_ExitMNG (NULL)
```

Macro Bidly_Exit will delete anonymous manager.

Definition at line 315 of file bidly.h.

3.1.2.45 Biddy_ExtractMinterm

```
#define Biddy_ExtractMinterm(  
    support,  
    f ) Biddy_Managed_ExtractMinterm(NULL, support, f)
```

Macro Biddy_ExtractMinterm is defined for use with anonymous manager.

Definition at line 938 of file biddy.h.

3.1.2.46 Biddy_FindFormula

```
#define Biddy_FindFormula(  
    x,  
    idx,  
    f ) Biddy_Managed_FindFormula(NULL, x, idx, f)
```

Macro Biddy_FindFormula is defined for use with anonymous manager.

Definition at line 634 of file biddy.h.

3.1.2.47 Biddy_FoaVariable

```
#define Biddy_FoaVariable(  
    x,  
    varelem ) Biddy_Managed_FoaVariable(NULL, x, varelem)
```

Macro Biddy_FoaVariable is defined for use with anonymous manager.

Definition at line 517 of file biddy.h.

3.1.2.48 Biddy_FormulaTableNum

```
#define Biddy_FormulaTableNum( ) Biddy_Managed_FormulaTableNum(NULL)
```

Macro Biddy_FormulaTableNum is defined for use with anonymous manager.

Definition at line 1094 of file biddy.h.

3.1.2.49 Biddy_GC

```
#define Biddy_GC(  
    targetLT,  
    targetGEQ,  
    purge,  
    total ) Biddy_Managed_GC(NULL, targetLT, targetGEQ, purge, total)
```

Macro Biddy_GC is defined for use with anonymous manager.

Macros Biddy_Managed_AutoGC, Biddy_AutoGC, Biddy_Managed_ForceGC, and Biddy_ForceGC are useful variants.

Definition at line 579 of file biddy.h.

3.1.2.50 **Bidly_GetBaseSet**

```
#define Bidly_GetBaseSet( ) Bidly_Managed_GetBaseSet(NULL)
```

Macro `Bidly_GetBaseSet` is defined for use with anonymous manager.

Definition at line 402 of file `bidly.h`.

3.1.2.51 **Bidly_GetConstantOne**

```
#define Bidly_GetConstantOne( ) Bidly_Managed_GetConstantOne(NULL)
```

Macro `Bidly_GetConstantOne` is defined for use with anonymous manager.

Definition at line 395 of file `bidly.h`.

3.1.2.52 **Bidly_GetConstantZero**

```
#define Bidly_GetConstantZero( ) Bidly_Managed_GetConstantZero(NULL)
```

Macro `Bidly_GetConstantZero` is defined for use with anonymous manager.

Definition at line 388 of file `bidly.h`.

3.1.2.53 **Bidly_GetElementEdge**

```
#define Bidly_GetElementEdge(  
    v ) Bidly_Managed_GetElementEdge(NULL, v)
```

Macro `Bidly_GetElementEdge` is defined for use with anonymous manager.

Definition at line 437 of file `bidly.h`.

3.1.2.54 **Bidly_GetIthFormula**

```
#define Bidly_GetIthFormula(  
    i ) Bidly_Managed_GetIthFormula(NULL, i)
```

Macro `Bidly_GetIthFormula` is defined for use with anonymous manager.

Definition at line 649 of file `bidly.h`.

3.1.2.55 Biddy_GetIthFormulaName

```
#define Biddy_GetIthFormulaName(  
    i ) Biddy_Managed_GetIthFormulaName(NULL, i)
```

Macro Biddy_GetIthFormulaName is defined for use with anonymous manager.

Definition at line 654 of file biddy.h.

3.1.2.56 Biddy_GetIthVariable

```
#define Biddy_GetIthVariable(  
    i ) Biddy_Managed_GetIthVariable(NULL, i)
```

Macro Biddy_GetIthVariable is defined for use with anonymous manager.

Definition at line 417 of file biddy.h.

3.1.2.57 Biddy_GetLowestVariable

```
#define Biddy_GetLowestVariable( ) Biddy_Managed_GetLowestVariable(NULL)
```

Macro Biddy_GetLowestVariable is defined for use with anonymous manager.

Definition at line 412 of file biddy.h.

3.1.2.58 Biddy_GetManagerName

```
#define Biddy_GetManagerName( ) Biddy_Managed_GetManagerName(NULL)
```

Macro Biddy_GetManagerName is defined for use with anonymous manager.

Definition at line 328 of file biddy.h.

3.1.2.59 Biddy_GetManagerType

```
#define Biddy_GetManagerType( ) Biddy_Managed_GetManagerType(NULL)
```

Macro Biddy_GetManagerType is defined for use with anonymous manager.

Definition at line 323 of file biddy.h.

3.1.2.60 Biddy_GetMark

```
#define Biddy_GetMark(  
    f ) (((uintptr_t) f & (uintptr_t) 1) != 0)
```

Biddy_GetMark returns TRUE iff given edge is complemented.

Definition at line 168 of file biddy.h.

3.1.2.61 Biddy_GetNextVariable

```
#define Biddy_GetNextVariable(  
    v ) Biddy_Managed_GetNextVariable(NULL, v)
```

Macro Biddy_GetNextVariable is defined for use with anonymous manager.

Definition at line 427 of file biddy.h.

3.1.2.62 Biddy_GetOrdering

```
#define Biddy_GetOrdering( ) Biddy_Managed_GetOrdering(NULL)
```

Macro Biddy_GetOrdering is defined for use with anonymous manager.

Definition at line 659 of file biddy.h.

3.1.2.63 Biddy_GetPrevVariable

```
#define Biddy_GetPrevVariable(  
    v ) Biddy_Managed_GetPrevVariable(NULL, v)
```

Macro Biddy_GetPrevVariable is defined for use with anonymous manager.

Definition at line 422 of file biddy.h.

3.1.2.64 Biddy_GetTag

```
#define Biddy_GetTag(  
    f ) 0
```

Biddy_GetTag returns tag used for the given edge, since Biddy v1.7.

Definition at line 199 of file biddy.h.

3.1.2.65 Biddy_GetTerminal

```
#define Biddy_GetTerminal( ) Biddy_Managed_GetTerminal(NULL)
```

Macro Biddy_GetTerminal is defined for use with anonymous manager.

Definition at line 383 of file biddy.h.

3.1.2.66 Biddy_GetTopVariableChar

```
#define Biddy_GetTopVariableChar(  
    f ) Biddy_Managed_GetTopVariableChar(NULL, f)
```

Macro Biddy_GetTopVariableChar is defined for use with anonymous manager.

Definition at line 457 of file biddy.h.

3.1.2.67 Biddy_GetTopVariableEdge

```
#define Biddy_GetTopVariableEdge(  
    f ) Biddy_Managed_GetTopVariableEdge(NULL, f)
```

Macro Biddy_GetTopVariableEdge is defined for use with anonymous manager.

Definition at line 447 of file biddy.h.

3.1.2.68 Biddy_GetTopVariableName

```
#define Biddy_GetTopVariableName(  
    f ) Biddy_Managed_GetTopVariableName(NULL, f)
```

Macro Biddy_GetTopVariableName is defined for use with anonymous manager.

Definition at line 452 of file biddy.h.

3.1.2.69 Biddy_GetVariable

```
#define Biddy_GetVariable(  
    x ) Biddy_Managed_GetVariable(NULL, x)
```

Macro Biddy_GetVariable is defined for use with anonymous manager.

Definition at line 407 of file biddy.h.

3.1.2.70 **Biddy_GetVariableData**

```
#define Biddy_GetVariableData(  
    v ) Biddy_Managed_GetVariableData (NULL, v)
```

Macro `Biddy_GetVariableData` is defined for use with anonymous manager.

Definition at line 487 of file `biddy.h`.

3.1.2.71 **Biddy_GetVariableEdge**

```
#define Biddy_GetVariableEdge(  
    v ) Biddy_Managed_GetVariableEdge (NULL, v)
```

Macro `Biddy_GetVariableEdge` is defined for use with anonymous manager.

Definition at line 432 of file `biddy.h`.

3.1.2.72 **Biddy_GetVariableName**

```
#define Biddy_GetVariableName(  
    v ) Biddy_Managed_GetVariableName (NULL, v)
```

Macro `Biddy_GetVariableName` is defined for use with anonymous manager.

Definition at line 442 of file `biddy.h`.

3.1.2.73 **Biddy_GetVariableValue**

```
#define Biddy_GetVariableValue(  
    v ) Biddy_Managed_GetVariableValue (NULL, v)
```

Macro `Biddy_GetVariableValue` is defined for use with anonymous manager.

Definition at line 472 of file `biddy.h`.

3.1.2.74 **Biddy_Gt**

```
#define Biddy_Gt(  
    f,  
    g ) Biddy_Managed_Gt (NULL, f, g)
```

Macro `Biddy_Gt` is defined for use with anonymous manager.

Definition at line 780 of file `biddy.h`.

3.1.2.75 Biddy_IncTag

```
#define Biddy_IncTag(  
    f ) Biddy_Managed_IncTag (NULL, f)
```

Macro Biddy_IncTag is defined for use with anonymous manager.

Definition at line 561 of file biddy.h.

3.1.2.76 Biddy_Init

```
#define Biddy_Init( ) Biddy_InitMNG (NULL, BIDDYTYPEOBDD)
```

Macros Biddy_Init and Biddy_InitAnonymous will initialize anonymous manager.

Definition at line 309 of file biddy.h.

3.1.2.77 Biddy_Inv

```
#define Biddy_Inv(  
    f ) ((Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1))
```

Biddy_Inv returns edge with changed complement bit.

Definition at line 181 of file biddy.h.

3.1.2.78 Biddy_InvCond

```
#define Biddy_InvCond(  
    f,  
    c ) (c ? ((Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1)) : f)
```

Biddy_InvCond returns edge with conditionally changed complement bit.

Definition at line 184 of file biddy.h.

3.1.2.79 Biddy_InvertMark

```
#define Biddy_InvertMark(  
    f ) (f = (Biddy_Edge) ((uintptr_t) f ^ (uintptr_t) 1))
```

Biddy_InvertMark changes complement bit of the given edge.

Definition at line 177 of file biddy.h.

3.1.2.80 **Bidly_IsEqv**

```
#define Bidly_IsEqv(  
    f1,  
    MNG2,  
    f2 ) Bidly_Managed_IsEqv (NULL, f1, MNG2, f2)
```

Macro `Bidly_IsEqv` is defined for use with anonymous manager.

Definition at line 353 of file `bidly.h`.

3.1.2.81 **Bidly_IsEqvPointer**

```
#define Bidly_IsEqvPointer(  
    f,  
    g ) (((uintptr_t) f & ~((uintptr_t) 1)) == ((uintptr_t) g & ~((uintptr_t) 1)))
```

`Bidly_IsEqvPointer` returns `TRUE` iff given edges points to the same node.

Definition at line 165 of file `bidly.h`.

3.1.2.82 **Bidly_IsHighest**

```
#define Bidly_IsHighest(  
    v ) Bidly_Managed_IsHighest (NULL, v)
```

Macro `Bidly_IsHighest` is defined for use with anonymous manager.

Definition at line 512 of file `bidly.h`.

3.1.2.83 **Bidly_IsLeq**

```
#define Bidly_IsLeq(  
    f,  
    g ) Bidly_Managed_IsLeq (NULL, f, g)
```

Macro `Bidly_IsLeq` is defined for use with anonymous manager.

Definition at line 787 of file `bidly.h`.

3.1.2.84 **Bidly_IsLowest**

```
#define Bidly_IsLowest(  
    v ) Bidly_Managed_IsLowest (NULL, v)
```

Macro `Bidly_IsLowest` is defined for use with anonymous manager.

Definition at line 507 of file `bidly.h`.

3.1.2.85 Biddy_IsNull

```
#define Biddy_IsNull(  
    f ) (f == NULL)
```

Biddy_IsNull returns TRUE iff given BDD is a null edge.

Definition at line 150 of file biddy.h.

3.1.2.86 Biddy_IsOK

```
#define Biddy_IsOK(  
    f ) Biddy_Managed_IsOK(NULL, f)
```

Macro Biddy_IsOK is defined for use with anonymous manager.

Definition at line 573 of file biddy.h.

3.1.2.87 Biddy_IsSelected

```
#define Biddy_IsSelected(  
    f ) Biddy_Managed_IsSelected(NULL, f)
```

Macro Biddy_IsSelected is defined for use with anonymous manager.

Definition at line 368 of file biddy.h.

3.1.2.88 Biddy_IsSmaller

```
#define Biddy_IsSmaller(  
    fv,  
    gv ) Biddy_Managed_IsSmaller(NULL, fv, gv)
```

Macro Biddy_IsSmaller is defined for use with anonymous manager.

Definition at line 502 of file biddy.h.

3.1.2.89 Biddy_IsTerminal

```
#define Biddy_IsTerminal(  
    f ) (((void*)((uintptr_t) f & ~((uintptr_t) 1))) [2] == NULL) && (((void*)((uintptr_t) f & ~((uintptr_t) 1))) [3] == NULL)
```

Biddy_IsTerminal returns TRUE iff given edge points to the terminal node.

Definition at line 160 of file biddy.h.

3.1.2.90 Bidly_IsVariableDependent

```
#define Bidly_IsVariableDependent(  
    f,  
    v ) Bidly_Managed_IsVariableDependent (NULL, f, v)
```

Macro Bidly_IsVariableDependent is defined for use with anonymous manager.

Definition at line 813 of file bidly.h.

3.1.2.91 Bidly_ITE

```
#define Bidly_ITE(  
    f,  
    g,  
    h ) Bidly_Managed_ITE (NULL, f, g, h)
```

Macro Bidly_ITE is defined for use with anonymous manager.

Definition at line 734 of file bidly.h.

3.1.2.92 Bidly_Leq

```
#define Bidly_Leq(  
    f,  
    g ) Bidly_Managed_Leq (NULL, f, g)
```

Macro Bidly_Leq is defined for use with anonymous manager.

Definition at line 775 of file bidly.h.

3.1.2.93 Bidly_ListAvgLength

```
#define Bidly_ListAvgLength( ) Bidly_Managed_ListAvgLength (NULL)
```

Macro Bidly_ListAvgLength is defined for use with anonymous manager.

Definition at line 1109 of file bidly.h.

3.1.2.94 Bidly_ListMaxLength

```
#define Bidly_ListMaxLength( ) Bidly_Managed_ListMaxLength (NULL)
```

Macro Bidly_ListMaxLength is defined for use with anonymous manager.

Definition at line 1104 of file bidly.h.

3.1.2.95 Biddy_ListUsed

```
#define Biddy_ListUsed( ) Biddy_Managed_ListUsed(NULL)
```

Macro Biddy_ListUsed is defined for use with anonymous manager.

Definition at line 1099 of file biddy.h.

3.1.2.96 Biddy_Managed_AvgLevel

```
#define Biddy_Managed_AvgLevel(  
    MNG,  
    f ) Biddy_AvgLevel(f)
```

Macro Biddy_Managed_AvgLevel(MNG,f) is defined for your convenience.

Definition at line 969 of file biddy.h.

3.1.2.97 Biddy_Managed_GetElse

```
#define Biddy_Managed_GetElse(  
    MNG,  
    f ) Biddy_GetElse(f)
```

Macro Biddy_Managed_GetElse is defined for your convenience.

Definition at line 343 of file biddy.h.

3.1.2.98 Biddy_Managed_GetThen

```
#define Biddy_Managed_GetThen(  
    MNG,  
    f ) Biddy_GetThen(f)
```

Macro Biddy_Managed_GetThen is defined for your convenience.

Definition at line 338 of file biddy.h.

3.1.2.99 Biddy_Managed_GetTopVariable

```
#define Biddy_Managed_GetTopVariable(  
    MNG,  
    f ) Biddy_GetTopVariable(f)
```

Macro Biddy_Managed_GetTopVariable is defined for your convenience.

Definition at line 348 of file biddy.h.

3.1.2.100 **Bidly_Managed_MaxLevel**

```
#define Bidly_Managed_MaxLevel(  
    MNG,  
    f ) Bidly_MaxLevel(f)
```

Macro [Bidly_Managed_MaxLevel\(MNG,f\)](#) is defined for your convenience.

Definition at line 964 of file bidly.h.

3.1.2.101 **Bidly_MaximizeBDD**

```
#define Bidly_MaximizeBDD(  
    f ) Bidly_Managed_MaximizeBDD(NULL, f)
```

Macro [Bidly_MaximizeBDD](#) is defined for use with anonymous manager.

Definition at line 694 of file bidly.h.

3.1.2.102 **Bidly_MaxNodes**

```
#define Bidly_MaxNodes(  
    f ) Bidly_Managed_MaxNodes(NULL, f)
```

Macro [Bidly_MaxNodes\(f\)](#) is defined for use with anonymous manager.

Definition at line 1176 of file bidly.h.

3.1.2.103 **Bidly_MinimizeBDD**

```
#define Bidly_MinimizeBDD(  
    f ) Bidly_Managed_MinimizeBDD(NULL, f)
```

Macro [Bidly_MinimizeBDD](#) is defined for use with anonymous manager.

Definition at line 689 of file bidly.h.

3.1.2.104 **Bidly_MinNodes**

```
#define Bidly_MinNodes(  
    f ) Bidly_Managed_MinNodes(NULL, f)
```

Macro [Bidly_MinNodes\(f\)](#) is defined for use with anonymous manager.

Definition at line 1171 of file bidly.h.

3.1.2.105 Biddy_Nand

```
#define Biddy_Nand(  
    f,  
    g ) Biddy_Managed_Nand(NULL, f, g)
```

Macro Biddy_Nand is defined for use with anonymous manager.

Definition at line 755 of file biddy.h.

3.1.2.106 Biddy_NodeTableAddNumber

```
#define Biddy_NodeTableAddNumber( ) Biddy_Managed_NodeTableAddNumber(NULL)
```

Macro Biddy_NodeTableAddNumber is defined for use with anonymous manager.

Definition at line 1029 of file biddy.h.

3.1.2.107 Biddy_NodeTableANDORNumber

```
#define Biddy_NodeTableANDORNumber( ) Biddy_Managed_NodeTableANDORNumber(NULL)
```

Macro Biddy_NodeTableANDORNumber is defined for use with anonymous manager.

Definition at line 1074 of file biddy.h.

3.1.2.108 Biddy_NodeTableANDORRecursiveNumber

```
#define Biddy_NodeTableANDORRecursiveNumber( ) Biddy_Managed_NodeTableANDORRecursiveNumber(NULL)
```

Macro Biddy_NodeTableANDORRecursiveNumber is defined for use with anonymous manager.

Definition at line 1079 of file biddy.h.

3.1.2.109 Biddy_NodeTableBlockNumber

```
#define Biddy_NodeTableBlockNumber( ) Biddy_Managed_NodeTableBlockNumber(NULL)
```

Macro Biddy_NodeTableBlockNumber is defined for use with anonymous manager.

Definition at line 984 of file biddy.h.

3.1.2.110 **Biddy_NodeTableCompareNumber**

```
#define Biddy_NodeTableCompareNumber( ) Biddy_Managed_NodeTableCompareNumber(NULL)
```

Macro `Biddy_NodeTableCompareNumber` is defined for use with anonymous manager.

Definition at line 1024 of file `bidly.h`.

3.1.2.111 **Biddy_NodeTableDRTime**

```
#define Biddy_NodeTableDRTime( ) Biddy_Managed_NodeTableDRTime(NULL)
```

Macro `Biddy_NodeTableDRTime` is defined for use with anonymous manager.

Definition at line 1059 of file `bidly.h`.

3.1.2.112 **Biddy_NodeTableFindNumber**

```
#define Biddy_NodeTableFindNumber( ) Biddy_Managed_NodeTableFindNumber(NULL)
```

Macro `Biddy_NodeTableFindNumber` is defined for use with anonymous manager.

Definition at line 1019 of file `bidly.h`.

3.1.2.113 **Biddy_NodeTableFoaNumber**

```
#define Biddy_NodeTableFoaNumber( ) Biddy_Managed_NodeTableFoaNumber(NULL)
```

Macro `Biddy_NodeTableFoaNumber` is defined for use with anonymous manager.

Definition at line 1014 of file `bidly.h`.

3.1.2.114 **Biddy_NodeTableGCNumber**

```
#define Biddy_NodeTableGCNumber( ) Biddy_Managed_NodeTableGCNumber(NULL)
```

Macro `Biddy_NodeTableGCNumber` is defined for use with anonymous manager.

Definition at line 1034 of file `bidly.h`.

3.1.2.115 Biddy_NodeTableGCObsoleteNumber

```
#define Biddy_NodeTableGCObsoleteNumber( ) Biddy_Managed_NodeTableGCObsoleteNumber(NULL)
```

Macro Biddy_NodeTableGCObsoleteNumber is defined for use with anonymous manager.

Definition at line 1044 of file biddy.h.

3.1.2.116 Biddy_NodeTableGCTime

```
#define Biddy_NodeTableGCTime( ) Biddy_Managed_NodeTableGCTime(NULL)
```

Macro Biddy_NodeTableGCTime is defined for use with anonymous manager.

Definition at line 1039 of file biddy.h.

3.1.2.117 Biddy_NodeTableGenerated

```
#define Biddy_NodeTableGenerated( ) Biddy_Managed_NodeTableGenerated(NULL)
```

Macro Biddy_NodeTableGenerated is defined for use with anonymous manager.

Definition at line 989 of file biddy.h.

3.1.2.118 Biddy_NodeTableITENumber

```
#define Biddy_NodeTableITENumber( ) Biddy_Managed_NodeTableITENumber(NULL)
```

Macro Biddy_NodeTableITENumber is defined for use with anonymous manager.

Definition at line 1064 of file biddy.h.

3.1.2.119 Biddy_NodeTableITERRecursiveNumber

```
#define Biddy_NodeTableITERRecursiveNumber( ) Biddy_Managed_NodeTableITERRecursiveNumber(NULL)
```

Macro Biddy_NodeTableITERRecursiveNumber is defined for use with anonymous manager.

Definition at line 1069 of file biddy.h.

3.1.2.120 Bidly_NodeTableMax

```
#define Bidly_NodeTableMax( ) Bidly_Managed_NodeTableMax(NULL)
```

Macro Bidly_NodeTableMax is defined for use with anonymous manager.

Definition at line 994 of file bidly.h.

3.1.2.121 Bidly_NodeTableNum

```
#define Bidly_NodeTableNum( ) Bidly_Managed_NodeTableNum(NULL)
```

Macro Bidly_NodeTableNum is defined for use with anonymous manager.

Definition at line 999 of file bidly.h.

3.1.2.122 Bidly_NodeTableNumVar

```
#define Bidly_NodeTableNumVar(  
    v ) Bidly_Managed_NodeTableNumVar(NULL, v)
```

Macro Bidly_NodeTableNumVar is defined for use with anonymous manager.

Definition at line 1004 of file bidly.h.

3.1.2.123 Bidly_NodeTableResizeNumber

```
#define Bidly_NodeTableResizeNumber( ) Bidly_Managed_NodeTableResizeNumber(NULL)
```

Macro Bidly_NodeTableResizeNumber is defined for use with anonymous manager.

Definition at line 1009 of file bidly.h.

3.1.2.124 Bidly_NodeTableSiftingNumber

```
#define Bidly_NodeTableSiftingNumber( ) Bidly_Managed_NodeTableSiftingNumber(NULL)
```

Macro Bidly_NodeTableSiftingNumber is defined for use with anonymous manager.

Definition at line 1054 of file bidly.h.

3.1.2.125 Bidly_NodeTableSize

```
#define Bidly_NodeTableSize( ) Bidly_Managed_NodeTableSize(NULL)
```

Macro Bidly_NodeTableSize is defined for use with anonymous manager.

Definition at line 979 of file bidly.h.

3.1.2.126 Bidly_NodeTableSwapNumber

```
#define Bidly_NodeTableSwapNumber( ) Bidly_Managed_NodeTableSwapNumber(NULL)
```

Macro Bidly_NodeTableSwapNumber is defined for use with anonymous manager.

Definition at line 1049 of file bidly.h.

3.1.2.127 Bidly_NodeTableXORNumber

```
#define Bidly_NodeTableXORNumber( ) Bidly_Managed_NodeTableXORNumber(NULL)
```

Macro Bidly_NodeTableXORNumber is defined for use with anonymous manager.

Definition at line 1084 of file bidly.h.

3.1.2.128 Bidly_NodeTableXORRecursiveNumber

```
#define Bidly_NodeTableXORRecursiveNumber( ) Bidly_Managed_NodeTableXORRecursiveNumber(NULL)
```

Macro Bidly_NodeTableXORRecursiveNumber is defined for use with anonymous manager.

Definition at line 1089 of file bidly.h.

3.1.2.129 Bidly_Nor

```
#define Bidly_Nor(  
    f,  
    g ) Bidly_Managed_Nor(NULL, f, g)
```

Macro Bidly_Nor is defined for use with anonymous manager.

Definition at line 760 of file bidly.h.

3.1.2.130 Bidly_Not

```
#define Bidly_Not(  
    f ) Bidly_Managed_Not (NULL, f)
```

Macro Bidly_Not is defined for use with anonymous manager.

For OBDD and OFDD, use macro Bidly_Inv.

Definition at line 729 of file bidly.h.

3.1.2.131 Bidly_OPCacheFind

```
#define Bidly_OPCacheFind( ) Bidly_Managed_OPCacheFind (NULL)
```

Macro Bidly_OPCacheFind is defined for use with anonymous manager.

Definition at line 1119 of file bidly.h.

3.1.2.132 Bidly_OPCacheInsert

```
#define Bidly_OPCacheInsert( ) Bidly_Managed_OPCacheInsert (NULL)
```

Macro Bidly_OPCacheInsert is defined for use with anonymous manager.

Definition at line 1124 of file bidly.h.

3.1.2.133 Bidly_OPCacheOverwrite

```
#define Bidly_OPCacheOverwrite( ) Bidly_Managed_OPCacheOverwrite (NULL)
```

Macro Bidly_OPCacheOverwrite is defined for use with anonymous manager.

Definition at line 1129 of file bidly.h.

3.1.2.134 Bidly_OPCacheSearch

```
#define Bidly_OPCacheSearch( ) Bidly_Managed_OPCacheSearch (NULL)
```

Macro Bidly_OPCacheSearch is defined for use with anonymous manager.

Definition at line 1114 of file bidly.h.

3.1.2.135 Biddy_Or

```
#define Biddy_Or(  
    f,  
    g ) Biddy_Managed_Or (NULL, f, g)
```

Macro Biddy_Or is defined for use with anonymous manager.

Macros Biddy_Managed_Union and Biddy_Union are defined for set manipulation.

Definition at line 748 of file biddy.h.

3.1.2.136 Biddy_Permitsym

```
#define Biddy_Permitsym(  
    f,  
    n ) Biddy_Managed_Permitsym (NULL, f, n)
```

Macro Biddy_Permitsym is defined for use with anonymous manager.

Definition at line 908 of file biddy.h.

3.1.2.137 Biddy_PrintfBDD

```
#define Biddy_PrintfBDD(  
    f ) Biddy_Managed_PrintfBDD (NULL, f)
```

Macro Biddy_PrintfBDD is defined for use with anonymous manager.

Definition at line 1230 of file biddy.h.

3.1.2.138 Biddy_PrintfMinterms

```
#define Biddy_PrintfMinterms(  
    f,  
    negative ) Biddy_Managed_PrintfMinterms (NULL, f, negative)
```

Macro Biddy_PrintfMinterms is defined for use with anonymous manager.

Definition at line 1260 of file biddy.h.

3.1.2.139 Biddy_PrintfSOP

```
#define Biddy_PrintfSOP(  
    f ) Biddy_Managed_PrintfSOP (NULL, f)
```

Macro Biddy_PrintfSOP is defined for use with anonymous manager.

Definition at line 1250 of file biddy.h.

3.1.2.140 Bidly_PrintfTable

```
#define Bidly_PrintfTable(  
    f ) Bidly_Managed_PrintfTable (NULL, f)
```

Macro Bidly_PrintfTable is defined for use with anonymous manager.

Definition at line 1240 of file bidly.h.

3.1.2.141 Bidly_PrintInfo

```
#define Bidly_PrintInfo(  
    f ) Bidly_Managed_PrintInfo (NULL, f)
```

Macro Bidly_PrintInfo is defined for use with anonymous manager.

Definition at line 1186 of file bidly.h.

3.1.2.142 Bidly_Product

```
#define Bidly_Product(  
    f,  
    g ) Bidly_Managed_Product (NULL, f, g)
```

Macro Bidly_Product is defined for use with anonymous manager.

Definition at line 886 of file bidly.h.

3.1.2.143 Bidly_Purge

```
#define Bidly_Purge( ) Bidly_Managed_Purge (NULL)
```

Macro Bidly_Purge is defined for use with anonymous manager.

Definition at line 593 of file bidly.h.

3.1.2.144 Bidly_PurgeAndReorder

```
#define Bidly_PurgeAndReorder(  
    f,  
    c ) Bidly_Managed_PurgeAndReorder (NULL, f, c)
```

Macro Bidly_PurgeAndReorder is defined for use with anonymous manager.

Definition at line 598 of file bidly.h.

3.1.2.145 Bidly_RandomFunction

```
#define Bidly_RandomFunction(  
    support,  
    r ) Bidly_Managed_RandomFunction(NULL, support, r)
```

Macro Bidly_RandomFunction is defined for use with anonymous manager.

Definition at line 928 of file bidly.h.

3.1.2.146 Bidly_RandomSet

```
#define Bidly_RandomSet(  
    unit,  
    r ) Bidly_Managed_RandomSet(NULL, unit, r)
```

Macro Bidly_RandomSet is defined for use with anonymous manager.

Definition at line 933 of file bidly.h.

3.1.2.147 Bidly_ReadBddview

```
#define Bidly_ReadBddview(  
    filename,  
    name ) Bidly_Managed_ReadBddview(NULL, filename, name)
```

Macro Bidly_ReadBddview is defined for use with anonymous manager.

Definition at line 1220 of file bidly.h.

3.1.2.148 Bidly_ReadMemoryInUse

```
#define Bidly_ReadMemoryInUse( ) Bidly_Managed_ReadMemoryInUse(NULL)
```

Macro Bidly_ReadMemoryInUse is defined for use with anonymous manager.

Definition at line 1181 of file bidly.h.

3.1.2.149 Bidly_ReadVerilogFile

```
#define Bidly_ReadVerilogFile(  
    filename,  
    prefix ) Bidly_Managed_ReadVerilogFile(NULL, filename, prefix)
```

Macro Bidly_ReadVerilogFile is defined for use with anonymous manager.

Definition at line 1225 of file bidly.h.

3.1.2.150 Bidly_Refresh

```
#define Bidly_Refresh(  
    f ) Bidly_Managed_Refresh(NULL, f)
```

Macro Bidly_Refresh is defined for use with anonymous manager.

Definition at line 603 of file bidly.h.

3.1.2.151 Bidly_Regular

```
#define Bidly_Regular(  
    f ) ((Bidly_Edge) ((uintptr_t) f & ~((uintptr_t) 1)))
```

Bidly_Regular returns not-complemented version of edge, since Bidly v1.4.

Definition at line 187 of file bidly.h.

3.1.2.152 Bidly_ReplaceByKeyword

```
#define Bidly_ReplaceByKeyword(  
    f,  
    keyword ) Bidly_Managed_ReplaceByKeyword(NULL, f, keyword)
```

Macro Bidly_ReplaceByKeyword is defined for use with anonymous manager.

Macros Bidly_Managed_Replace and Bidly_Replace are variants with less effective cache table

Definition at line 851 of file bidly.h.

3.1.2.153 Bidly_ResetVariablesValue

```
#define Bidly_ResetVariablesValue( ) Bidly_Managed_ResetVariablesValue(NULL)
```

Macro Bidly_ResetVariablesValue is defined for use with anonymous manager.

Definition at line 462 of file bidly.h.

3.1.2.154 Bidly_Restrict

```
#define Bidly_Restrict(  
    f,  
    v,  
    value ) Bidly_Managed_Restrict(NULL, f, v, value)
```

Macro Bidly_Restrict is defined for use with anonymous manager.

Definition at line 793 of file bidly.h.

3.1.2.155 Biddy_SelectFunction

```
#define Biddy_SelectFunction(  
    f ) Biddy_Managed_SelectFunction (NULL, f)
```

Macro Biddy_SelectFunction is defined for use with anonymous manager.

Definition at line 373 of file biddy.h.

3.1.2.156 Biddy_SelectiveProduct

```
#define Biddy_SelectiveProduct (  
    f,  
    g,  
    cube ) Biddy_Managed_SelectiveProduct (NULL, f, g, cube)
```

Macro Biddy_SelectiveProduct is defined for use with anonymous manager.

Definition at line 891 of file biddy.h.

3.1.2.157 Biddy_SelectNode

```
#define Biddy_SelectNode(  
    f ) Biddy_Managed_SelectNode (NULL, f)
```

Macro Biddy_SelectNode is defined for use with anonymous manager.

Definition at line 358 of file biddy.h.

3.1.2.158 Biddy_SetAlphabeticOrdering

```
#define Biddy_SetAlphabeticOrdering( ) Biddy_Managed_SetAlphabeticOrdering (NULL)
```

Macro Biddy_SetAlphabeticOrdering is defined for use with anonymous manager.

Definition at line 669 of file biddy.h.

3.1.2.159 Biddy_SetManagerParameters

```
#define Biddy_SetManagerParameters(  
    gcr,  
    gcrF,  
    gcrX,  
    rr,  
    rrF,  
    rrX,  
    st,  
    cst ) Biddy_Managed_SetManagerParameters (NULL, gcr, gcrF, gcrX, rr, rrF, rrX, st, cst)
```

Macro Biddy_SetManagerParameters is defined for use with anonymous manager.

Definition at line 333 of file biddy.h.

3.1.2.160 **Biddy_SetMark**

```
#define Biddy_SetMark(  
    f ) (f = (Biddy_Edge) ((uintptr_t) f | (uintptr_t) 1))
```

Biddy_SetMark makes given edge complemented.

Definition at line 171 of file `biddy.h`.

3.1.2.161 **Biddy_SetOrdering**

```
#define Biddy_SetOrdering(  
    ordering ) Biddy_Managed_SetOrdering (NULL, ordering)
```

Macro **Biddy_SetOrdering** is defined for use with anonymous manager.

Definition at line 664 of file `biddy.h`.

3.1.2.162 **Biddy_SetTag**

```
#define Biddy_SetTag(  
    f,  
    t ) 0
```

Biddy_SetTag adds tag to the given edge, since Biddy v1.7.

Definition at line 208 of file `biddy.h`.

3.1.2.163 **Biddy_SetVariableData**

```
#define Biddy_SetVariableData(  
    v,  
    x ) Biddy_Managed_SetVariableData (NULL, v, x)
```

Macro **Biddy_SetVariableData** is defined for use with anonymous manager.

Definition at line 482 of file `biddy.h`.

3.1.2.164 **Biddy_SetVariableValue**

```
#define Biddy_SetVariableValue(  
    v,  
    f ) Biddy_Managed_SetVariableValue (NULL, v, f)
```

Macro **Biddy_SetVariableValue** is defined for use with anonymous manager.

Definition at line 467 of file `biddy.h`.

3.1.2.165 Biddy_Sifting

```
#define Biddy_Sifting(  
    f,  
    c ) Biddy_Managed_Sifting (NULL, f, c)
```

Macro Biddy_Sifting is defined for use with anonymous manager.

Definition at line 684 of file biddy.h.

3.1.2.166 Biddy_Simplify

```
#define Biddy_Simplify(  
    f,  
    c ) Biddy_Managed_Simplify (NULL, f, c)
```

Macro Biddy_Simplify is defined for use with anonymous manager.

Definition at line 839 of file biddy.h.

3.1.2.167 Biddy_Stretch

```
#define Biddy_Stretch(  
    f ) Biddy_Managed_Stretch (NULL, f)
```

Macro Biddy_Stretch is defined for use with anonymous manager.

Definition at line 913 of file biddy.h.

3.1.2.168 Biddy_Subset

```
#define Biddy_Subset(  
    f,  
    g ) Biddy_Managed_Subset (NULL, f, g)
```

Macro Biddy_Subset is defined for use with anonymous manager.

Definition at line 903 of file biddy.h.

3.1.2.169 Biddy_Support

```
#define Biddy_Support(  
    f ) Biddy_Managed_Support (NULL, f)
```

Macro Biddy_Support is defined for use with anonymous manager.

Definition at line 844 of file biddy.h.

3.1.2.170 **Bidly_Supset**

```
#define Bidly_Supset(  
    f,  
    g ) Bidly_Managed_Supset (NULL, f, g)
```

Macro `Bidly_Supset` is defined for use with anonymous manager.

Definition at line 897 of file `bidly.h`.

3.1.2.171 **Bidly_SwapWithHigher**

```
#define Bidly_SwapWithHigher(  
    v ) Bidly_Managed_SwapWithHigher (NULL, v)
```

Macro `Bidly_SwapWithHigher` is defined for use with anonymous manager.

Definition at line 674 of file `bidly.h`.

3.1.2.172 **Bidly_SwapWithLower**

```
#define Bidly_SwapWithLower(  
    v ) Bidly_Managed_SwapWithLower (NULL, v)
```

Macro `Bidly_SwapWithLower` is defined for use with anonymous manager.

Definition at line 679 of file `bidly.h`.

3.1.2.173 **Bidly_TaggedFoaNode**

```
#define Bidly_TaggedFoaNode(  
    v,  
    pf,  
    pt,  
    ptag,  
    garbageAllowed ) Bidly_Managed_TaggedFoaNode (NULL, v, pf, pt, ptag, garbageAllowed)
```

Macro `Bidly_TaggedFoaNode` is defined for use with anonymous manager.

Definition at line 566 of file `bidly.h`.

3.1.2.174 Biddy_TransferMark

```
#define Biddy_TransferMark(  
    f,  
    mark,  
    leftright ) Biddy_Managed_TransferMark(NULL, f, mark, leftright)
```

Macro Biddy_TransferMark is defined for use with anonymous manager.

For OBDD, use macro Biddy_InvCond.

Definition at line 556 of file biddy.h.

3.1.2.175 Biddy_UnivAbstract

```
#define Biddy_UnivAbstract(  
    f,  
    cube ) Biddy_Managed_UnivAbstract(NULL, f, cube)
```

Macro Biddy_UnivAbstract is defined for use with anonymous manager.

Definition at line 823 of file biddy.h.

3.1.2.176 Biddy_Untagged

```
#define Biddy_Untagged(  
    f ) 0
```

Biddy_Untagged returns untagged version of edge, since Biddy v1.7.

Definition at line 224 of file biddy.h.

3.1.2.177 Biddy_VariableTableNum

```
#define Biddy_VariableTableNum( ) Biddy_Managed_VariableTableNum(NULL)
```

Macro Biddy_VariableTableNum is defined for use with anonymous manager.

Definition at line 974 of file biddy.h.

3.1.2.178 Biddy_VarSubset

```
#define Biddy_VarSubset(  
    f,  
    v,  
    value ) Biddy_Managed_VarSubset(NULL, f, v, value)
```

Macro Biddy_VarSubset is defined for use with anonymous manager.

Definition at line 868 of file biddy.h.

3.1.2.179 Bidly_WriteBDD

```
#define Bidly_WriteBDD(  
    filename,  
    f,  
    label ) Bidly_Managed_WriteBDD(NULL, filename, f, label)
```

Macro Bidly_WriteBDD is defined for use with anonymous manager.

Definition at line 1235 of file bidly.h.

3.1.2.180 Bidly_WriteBddview

```
#define Bidly_WriteBddview(  
    filename,  
    f,  
    label,  
    table ) Bidly_Managed_WriteBddview(NULL, filename, f, label, table)
```

Macro Bidly_WriteBddview is defined for use with anonymous manager.

Definition at line 1270 of file bidly.h.

3.1.2.181 Bidly_WriteDot

```
#define Bidly_WriteDot(  
    filename,  
    f,  
    label,  
    id,  
    cudd ) Bidly_Managed_WriteDot(NULL, filename, f, label, id, cudd)
```

Macro Bidly_WriteDot is defined for use with anonymous manager.

Definition at line 1265 of file bidly.h.

3.1.2.182 Bidly_WriteSOP

```
#define Bidly_WriteSOP(  
    filename,  
    f ) Bidly_Managed_WriteSOP(NULL, filename, f)
```

Macro Bidly_WriteSOP is defined for use with anonymous manager.

Definition at line 1255 of file bidly.h.

3.1.2.183 Biddy_WriteTable

```
#define Biddy_WriteTable(  
    filename,  
    f ) Biddy_Managed_WriteTable(NULL, filename, f)
```

Macro Biddy_WriteTable is defined for use with anonymous manager.

Definition at line 1245 of file biddy.h.

3.1.2.184 Biddy_Xnor

```
#define Biddy_Xnor(  
    f,  
    g ) Biddy_Managed_Xnor(NULL, f, g)
```

Macro Biddy_Xnor is defined for use with anonymous manager.

Definition at line 770 of file biddy.h.

3.1.2.185 Biddy_Xor

```
#define Biddy_Xor(  
    f,  
    g ) Biddy_Managed_Xor(NULL, f, g)
```

Macro Biddy_Xor is defined for use with anonymous manager.

Definition at line 765 of file biddy.h.

3.1.3 Function Documentation

3.1.3.1 Biddy_About()

```
EXTERN Biddy_String Biddy_About ( )
```

Description

Side effects

More info

Definition at line 1127 of file biddyMain.c.

3.1.3.2 Biddy_AvgLevel()

```
EXTERN float Biddy_AvgLevel (
    Biddy_Edge f )
```

Description

Side effects

The result may not be compatible with your definition of Average Level for DAG. The result is especially problematic if there exist nodes with two equal descendants (e.g for ZBDDs and TZBDDs).

More info

Macro [Biddy_Managed_AvgLevel\(f\)](#) is defined for user convenience.

Definition at line 198 of file `biddyStat.c`.

3.1.3.3 Biddy_ExitMNG()

```
EXTERN void Biddy_ExitMNG (
    Biddy_Manager * mng )
```

Description

Deallocates all memory allocated by `Biddy_InitMNG`, `Biddy_FoaVariable`, `Biddy_FoaNode` etc.

Side effects

More info

Macro [Biddy_Exit\(\)](#) will delete anonymous manager.

Definition at line 904 of file `biddyMain.c`.

3.1.3.4 Biddy_GetElse()

```
EXTERN Biddy_Edge Biddy_GetElse (
    Biddy_Edge fun )
```

Description

Input mark is not transferred! External use, only.

Side effects

For terminal nodes, function returns the same node.

More info

Macro BidyE(fun) is defined for internal use.

Definition at line 1303 of file biddyMain.c.

3.1.3.5 Bidy_GetThen()

```
EXTERN Bidy_Edge Bidy_GetThen (  
    Bidy_Edge fun )
```

Description

Input mark is not transferred! External use, only.

Side effects

For terminal nodes, function returns the same node.

More info

Macro BidyT(fun) is defined for internal use.

Definition at line 1264 of file biddyMain.c.

3.1.3.6 Bidy_GetTopVariable()

```
EXTERN Bidy_Variable Bidy_GetTopVariable (  
    Bidy_Edge fun )
```

Description

External use, only.

Side effects**More info**

Macro BidyV(fun) is defined for internal use.

Definition at line 1341 of file biddyMain.c.

3.1.3.7 Bidy_InitMNG()

```
EXTERN void Bidy_InitMNG (  
    Bidy_Manager * mng,  
    int bdtype )
```

Description

Biddy_InitMNG creates and initializes a manager. Initialization consists of creating manager structure (MNG), node table (biddyNodeTable), variable table (biddyVariableTable), formula table (biddyFormulaTable), four basic caches (biddyOPCache, biddyEACache, biddyRCCache, and biddyReplaceCache), and cache list (biddyCacheList). Biddy_InitMNG also initializes constant edges (biddyOne, biddyZero), memory management and automatic garbage collection.

Side effects

Allocates a lot of memory.

More info

Macro Biddy_InitAnonymous() will initialize anonymous manager. Macro [Biddy_Init\(\)](#) will initialize anonymous manager for ROBDDs.

Definition at line 179 of file biddyMain.c.

3.1.3.8 Biddy_Managed_A()

```
EXTERN Biddy_Edge Biddy_Managed_A (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Variable v )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD. Be careful: $Ax A F \neq \text{NOT}(Ax A (\text{NOT } F))$. Counterexample: $Axb (\text{AND} (\text{NOT } a) b c)$.

More info

Macro [Biddy_A\(f,v\)](#) is defined for use with anonymous manager.

Definition at line 3826 of file biddyOp.c.

3.1.3.9 Biddy_Managed_AddCache()

```
EXTERN void Biddy_Managed_AddCache (
    Biddy_Manager MNG,
    Biddy_GCFunction gc )
```

Description

If Cache list does not exist, function creates it.

Side effects**More info**

Macro [Bidly_AddCache\(gc\)](#) is defined for use with anonymous manager.

Definition at line 4437 of file `bidlyMain.c`.

3.1.3.10 Bidly_Managed_AddElementByName()

```
EXTERN Bidly_Variable Bidly_Managed_AddElementByName (
    Bidly_Manager MNG,
    Bidly_String x )
```

Description

`Bidly_Managed_AddElementByName` uses `Bidly_Managed_FoaVariable` to find or add element. Function returns element edge. If element already exists, function returns the existing element edge. For more details see `Bidly_Managed_FoaVariable`.

Side effects

See `Bidly_Managed_FoaVariable`.

More info

Macro [Bidly_AddElementByName\(x\)](#) is defined for use with anonymous manager. Macros `Bidly_Managed_AddElement(MNG)` and `Bidly_AddElement()` are defined for creating numbered elements.

Definition at line 2928 of file `bidlyMain.c`.

3.1.3.11 Bidly_Managed_AddFormula()

```
EXTERN unsigned int Bidly_Managed_AddFormula (
    Bidly_Manager MNG,
    Bidly_String x,
    Bidly_Edge f,
    int c )
```

Description

Given BDD becomes a formula. If ($x \neq \text{NULL}$) then formula is accessible by its name. If ($x \neq \text{NULL}$) then index of the formula in the Formulae Table is returned, otherwise function returns 0. Nodes of the given BDD will be preserved for the given number of clearings. If ($c == -1$) then formula is refreshed but not preserved. If ($c == 0$) then formula is persistently preserved. You have to use `Bidly_DeleteFormula` and `Bidly_Purge` to remove nodes of persistently preserved formulae. There are five macros defined to simplify formulae management: `Bidly_Managed_AddTmpFormula(mng,name,bdd) := Bidly_Managed_AddFormula(mng,name,bdd,-1)` `Bidly_Managed_AddPersistentFormula(mng,name,bdd) := Bidly_Managed_AddFormula(mng,name,bdd,0)` `Bidly_Managed_KeepFormula(mng,bdd) := Bidly_Managed_AddFormula(mng,NULL,bdd,1)` `Bidly_Managed_KeepFormulaProlonged(mng,bdd,c) := Bidly_Managed_AddFormula(mng,NULL,bdd,c)` `Bidly_Managed_KeepFormulaUntilPurge(mng,bdd) := Bidly_Managed_AddFormula(mng,NULL,bdd,0)`

Side effects

Function is prolonged or fortified. Formulae with name are ordered by name. If formula with the same name already exists, it will be overwritten - preserved (i.e. not obsolete and not fresh) and persistently preserved formulae will be deleted at the original index and recreated at new index!

More info

Macros `Biddy_AddFormula(x,f,c)`, `Biddy_AddTmpFormula(x,f)`, `Biddy_AddPersistentFormula(x,f)`, `Biddy_KeepFormula(f)`, `Biddy_KeepFormulaProlonged(f,c)`, and `Biddy_KeepFormulaUntilPurge(f)` are defined for use with anonymous manager.

Definition at line 4511 of file `biddyMain.c`.

3.1.3.12 Biddy_Managed_AddVariableAbove()

```
EXTERN Biddy_Edge Biddy_Managed_AddVariableAbove (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

`Biddy_Managed_AddVariableAbove` uses `Biddy_Managed_AddVariableByName` to add new variable. Then, the order of the new variable is changed to become immediately above the given variable (above = previous = topmore in BDD) Function returns variable edge.

Side effects**More info**

Macro `Biddy_AddVariableAbove(v)` is defined for use with anonymous manager.

Definition at line 3040 of file `biddyMain.c`.

3.1.3.13 Biddy_Managed_AddVariableBelow()

```
EXTERN Biddy_Edge Biddy_Managed_AddVariableBelow (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

`Biddy_Managed_AddVariableBelow` uses `Biddy_Managed_AddVariableByName` to add new variable. Then, the order of the new variable is changed to become immediately below the given variable (below = next = bottommore in BDD) Function returns variable edge.

Side effects**More info**

Macro [Bidly_AddVariableBelow\(v\)](#) is defined for use with anonymous manager.

Definition at line 2957 of file bidlyMain.c.

3.1.3.14 Bidly_Managed_AddVariableByName()

```
EXTERN Bidly_Variable Bidly_Managed_AddVariableByName (
    Bidly_Manager MNG,
    Bidly_String x )
```

Description

[Bidly_Managed_AddVariableByName](#) uses [Bidly_Managed_FoaVariable](#) to find or add variable. Function returns variable edge. If variable already exists, function returns the existing variable edge. For more details see [Bidly_Managed_FoaVariable](#).

Side effects

See [Bidly_Managed_FoaVariable](#).

More info

Macro [Bidly_AddVariableByName\(x\)](#) is defined for use with anonymous manager. Macros [Bidly_Managed_AddVariable\(MNG\)](#) and [Bidly_AddVariable\(\)](#) are defined for creating numbered variables.

Definition at line 2896 of file bidlyMain.c.

3.1.3.15 Bidly_Managed_And()

```
EXTERN Bidly_Edge Bidly_Managed_And (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

For combination sets, this function coincides with Intersection.

Side Effects

Used by ITE (for OBDD). Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F' * H$. For all other BDD types, results are cached as (f,g,bidlyZero).

More Info

Macro [Bidly_And\(f,g\)](#) is defined for use with anonymous manager. Macros [Bidly_Managed_Intersect\(MNG,f,g\)](#) and [Bidly_Intersect\(f,g\)](#) are defined for manipulation of combination sets.

Definition at line 812 of file `bidlyOp.c`.

3.1.3.16 Bidly_Managed_AndAbstract()

```
EXTERN Bidly_Edge Bidly_Managed_AndAbstract (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g,
    Bidly_Edge cube )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro [Bidly_AndAbstract\(f,g,cube\)](#) is defined for use with anonymous manager.

Definition at line 4335 of file `bidlyOp.c`.

3.1.3.17 Bidly_Managed_Change()

```
EXTERN Bidly_Edge Bidly_Managed_Change (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters `(f,bidlyNull,v)`.

More info

Macro [Bidly_Change\(\)](#) is defined for use with anonymous manager.

Definition at line 5692 of file `bidlyOp.c`.

3.1.3.18 Biddy_Managed_ChangeVariableName()

```
EXTERN void Biddy_Managed_ChangeVariableName (
    Biddy_Manager MNG,
    Biddy_Variable v,
    Biddy_String x )
```

Description

Side effects

It is not checked that the same variable/element does not already exist. The ordering of the variable is not changed.

More info

Macro [Biddy_ChangeVariableName\(v,x\)](#) is defined for use with anonymous manager.

Definition at line 2860 of file `biddyMain.c`.

3.1.3.19 Biddy_Managed_Clean()

```
EXTERN void Biddy_Managed_Clean (
    Biddy_Manager MNG )
```

Description

Discard all nodes which are not preserved or which are not preserved anymore. Obsolete nodes are not immediately removed, they will be removed during the first garbage collection.

Side effects

Field deleted is not considered and thus no fortified node and no prolonged node is discarded.

More info

Macro [Biddy_Clean\(\)](#) is defined for use with anonymous manager.

Definition at line 4283 of file `biddyMain.c`.

3.1.3.20 Biddy_Managed_ClearVariablesData()

```
EXTERN void Biddy_Managed_ClearVariablesData (
    Biddy_Manager MNG )
```

Description

Side effects

Only active (used) variables are considered.

More info

Macro `Biddy_ClearVariablesData()` is defined for use with anonymous manager.

Definition at line 2168 of file `biddyMain.c`.

3.1.3.21 Biddy_Managed_Compose()

```
EXTERN Biddy_Edge Biddy_Managed_Compose (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g,
    Biddy_Variable v )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDs, recursive calls use optimization: $F(a=G) == \text{NOT}((\text{NOT } F)(a=G))$.

More info

Macro `Biddy_Compose(f,g,v)` is defined for use with anonymous manager.

Definition at line 3383 of file `biddyOp.c`.

3.1.3.22 Biddy_Managed_Constrain()

```
EXTERN Biddy_Edge Biddy_Managed_Constrain (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge c )
```

Description

Coudert and Madre's constrain function is also called a generalized cofactor of function f with respect to function c . Here is an explanation from <http://gauss.ececs.uc.edu/Courses/c626/lectures/BD↔D/bdd-desc.pdf> "BDD g is a generalized co-factor of f and c if for any truth assignment t , $g(t)$ has the same value as $f(t')$ where t' is the "nearest" truth assignment to t that maps c to 1. By definition, the result of this operation depends on the underlying BDD variable ordering so it cannot be regarded as a logical operation."

Side effects

Original BDD is not changed. Implemented only for OBDD and OBDDC. Cache table is not used.

More info

Macro `Bidly_Constrain(f,c)` is defined for use with anonymous manager.

Definition at line 4775 of file `bidlyOp.c`.

3.1.3.23 Bidly_Managed_ConstructBDD()

```
EXTERN Bidly_Edge Bidly_Managed_ConstructBDD (
    Bidly_Manager MNG,
    int numV,
    Bidly_String varlist,
    int numN,
    Bidly_String nodelist )
```

Function***** **Synopsis** [Function `Bidly_Managed_ConstructBDD`.] **Description** [`Bidly_Managed_ConstructBDD` constructs BDD from lists of nodes and edges. In both lists, elements are separated with spaces. Elements of node list has the following format: terminal node zero = (type=0,l=-1,r=-1), terminal node one = (type=1,l=-1,r=-1), regular label = (type=2,r=-1), complemented label = (type=3,r=-1), regular node = (type=4), node with complemented successor(s) = (type=5) The first element in nodelist is ignored. An example node list is: 'Bidly 0 Bidly 2 1 -1 1 B 4 2 3 2 0 0 -1 -1 3 i 4 4 9 4 d 4 5 6 5 0 0 -1 -1 6 y 4 7 8 7 0 0 -1 -1 8 1 1 -1 -1 9 d 4 6 10 10 1 1 -1 -1' which is constructed as: '{Bidly} {0 Bidly 2 1 -1} {1 B 4 2 3} {2 0 0 -1 -1} {3 i 4 4 9} {4 d 4 5 6} {5 0 0 -1 -1} {6 y 4 7 8} {7 0 0 -1 -1} {8 1 1 -1 -1} {9 d 4 6 10} {10 1 1 -1 -1}'.] **SideEffects** [If variable ordering in the file is not compatible with the active ordering then the result will be wrong.] **SeeAlso** []

Definition at line 6203 of file `bidlyMain.c`.

3.1.3.24 Bidly_Managed_CountComplementedEdges()

```
EXTERN unsigned int Bidly_Managed_CountComplementedEdges (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Count number of complemented edges in a given BDD.

Side effects

Terminal 0 is represented by complemented edge to terminal 1 with all BDD types but this edge is counted as a complemented one only if complemented edges are explicitly used.

More info

Macro `Bidly_Managed_CountComplementedEdges(f)` is defined for use with anonymous manager.

Definition at line 1407 of file `bidlyStat.c`.

3.1.3.25 Bidly_Managed_CountMinterms()

```
EXTERN double Bidly_Managed_CountMinterms (
    Bidly_Manager MNG,
    Bidly_Edge f,
    unsigned int nvars )
```

Description

For combination sets, this function coincides with combinations counting. Parameter `nvars` is a user-defined number of dependent variables. If (`nvars == 0`) then all noticeable variables are considered.

Side effects

We are using GNU Multiple Precision Arithmetic Library (GMP). For ZBDDs, this function coincides with the 1-path count. For ZBDDs, result does not depend on the number of dependent variables. For OBDD, noticeable variables are all variables existing in the graph. For TZBDD, noticeable variables are all variables equal or below a top variable (considering the tag). For OBDDs and TZBDDs, if (`nvars == 0`) the result may not be consistent with `Bidly_PrintfMinterms` because this function considers noticeable variables, while `Bidly_PrintfMinterms` considers all created variables.

More info

Macro `Bidly_CountMinterms(f,nvars)` is defined for use with anonymous manager. Macros `Bidly_Managed_CountCombination(MNG,f,nvars)` and `Bidly_CountCombinations(f,nvars)` are defined for use with combination sets.

Definition at line 1539 of file `bidlyStat.c`.

3.1.3.26 Bidly_Managed_CountNodes()

```
EXTERN unsigned int Bidly_Managed_CountNodes (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Count number of nodes in a BDD.

Side effects

This function must be managed because node selection is used.

More info

Macro `Bidly_CountNodes(f)` is defined for use with anonymous manager.

Definition at line 85 of file `bidlyStat.c`.

3.1.3.27 Biddy_Managed_CountNodesPlain()

```
EXTERN unsigned int Biddy_Managed_CountNodesPlain (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Count number of nodes in a corresponding BDD without complement edges.

Side effects

More info

Macro Biddy_Managed_CountNodesPlain(f) is defined for use with anonymous manager.

Definition at line 1195 of file biddyStat.c.

3.1.3.28 Biddy_Managed_CountPaths()

```
EXTERN unsigned long long int Biddy_Managed_CountPaths (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. TO DO: implement this using GNU Multiple Precision Arithmetic Library (GMP).

More info

Macro Biddy_CountPaths(f) is defined for use with anonymous manager.

Definition at line 1463 of file biddyStat.c.

3.1.3.29 Biddy_Managed_CreateFunction()

```
EXTERN Biddy_Edge Biddy_Managed_CreateFunction (
    Biddy_Manager MNG,
    Biddy_Edge support,
    long long unsigned int x )
```

Description

The represented Boolean function depends on the variables given with parameter support whilst the parameter n determines the generated function.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro `Bidly_CreateFunction(support,x)` is defined for use with anonymous manager.

Definition at line 7670 of file `bidlyOp.c`.

3.1.3.30 Bidly_Managed_CreateMinterm()

```
EXTERN Bidly_Edge Bidly_Managed_CreateMinterm (  
    Bidly_Manager MNG,  
    Bidly_Edge support,  
    long long unsigned int x )
```

Description

The represented Boolean function depends on the variables given with parameter `support` whilst the parameter `n` determines the generated minterm.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro `Bidly_CreateMinterm(support,x)` is defined for use with anonymous manager.

Definition at line 7590 of file `bidlyOp.c`.

3.1.3.31 Bidly_Managed_DeleteFormula()

```
EXTERN Bidly_Boolean Bidly_Managed_DeleteFormula (  
    Bidly_Manager MNG,  
    Bidly_String x )
```

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. Formulae representing constants and variables will not be deleted.

More info

Macro [Bidly_DeleteFormula\(x\)](#) is defined for use with anonymous manager.

Definition at line 4964 of file bidlyMain.c.

3.1.3.32 Bidly_Managed_DeleteIthFormula()

```
EXTERN Bidly_Boolean Bidly_Managed_DeleteIthFormula (  
    Bidly_Manager MNG,  
    unsigned int i )
```

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. The first two formulae ("0" and "1") will not be deleted. Formulae representing variables will not be deleted.

More info

Macro [Bidly_DeleteIthFormula\(x\)](#) is defined for use with anonymous manager.

Definition at line 5027 of file bidlyMain.c.

3.1.3.33 Bidly_Managed_DensityOfBDD()

```
EXTERN double Bidly_Managed_DensityOfBDD (  
    Bidly_Manager MNG,  
    Bidly_Edge f,  
    unsigned int nvars )
```

Description

If nvars == 0 then number of dependent variables is used.

Side effects

More info

Macro [Bidly_DensityOfBDD\(f,nvars\)](#) is defined for use with anonymous manager.

Definition at line 1756 of file bidlyStat.c.

3.1.3.34 Bidly_Managed_DensityOfFunction()

```
EXTERN double Bidly_Managed_DensityOfFunction (
    Bidly_Manager MNG,
    Bidly_Edge f,
    unsigned int nvars )
```

Description

If `nvars == 0` then number of dependent variables is used.

Side effects

More info

Macro [Bidly_DensityOfFunction\(f,nvars\)](#) is defined for use with anonymous manager.

Definition at line 1683 of file `bidlyStat.c`.

3.1.3.35 Bidly_Managed_DependentVariableNumber()

```
EXTERN unsigned int Bidly_Managed_DependentVariableNumber (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Boolean select )
```

Description

Count number of dependent variables. For OBDDs, the number of dependent variables is the same as the number of variables in the graph. For ZBDDs and TZBDDs, this is not true. If (`select == TRUE`) then dependent variables remain selected otherwise the function will unselect them.

Side effects

For ZBDDs, variables above the top variable (which are always all dependent) are also counted and selected!

More info

Macro [Bidly_DependentVariableNumber\(f\)](#) is defined for use with anonymous manager.

Definition at line 1283 of file `bidlyStat.c`.

3.1.3.36 Bidly_Managed_DeselectAll()

```
EXTERN void Bidly_Managed_DeselectAll (
    Bidly_Manager MNG )
```

Description

Deselect all nodes.

Side effects**More info**

Macro [Bidly_DeselectAll\(\)](#) is defined for use with anonymous manager.

Definition at line 1536 of file `bidlyMain.c`.

3.1.3.37 Bidly_Managed_DeselectNode()

```
EXTERN void Bidly_Managed_DeselectNode (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description**Side effects****More info**

Macro [Bidly_DeselectNode\(f\)](#) is defined for use with anonymous manager.

Definition at line 1426 of file `bidlyMain.c`.

3.1.3.38 Bidly_Managed_E()

```
EXTERN Bidly_Edge Bidly_Managed_E (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TzBDD. Be careful: $ExA \ F \neq \text{NOT}(ExA \ (\text{NOT } F))$. Counterexample: $Exb \ (\text{AND} \ (\text{NOT } a) \ b \ c)$.

More info

Macro [Bidly_E\(f,v\)](#) is defined for use with anonymous manager.

Definition at line 3618 of file `bidlyOp.c`.

3.1.3.39 Bidly_Managed_ElementAbstract()

```
EXTERN Bidly_Edge Bidly_Managed_ElementAbstract (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v )
```

Description

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro [Bidly_ElementAbstract\(f,v\)](#) is defined for use with anonymous manager.

Definition at line 6141 of file bidlyOp.c.

3.1.3.40 Bidly_Managed_Eval0()

```
EXTERN Bidly_String Bidly_Managed_Eval0 (
    Bidly_Manager MNG,
    Bidly_String s )
```

Description

First word is a name. It is followed by raw format. Function return name of the formula.

Side effects

All variables should already exists in the correct ordering! Not reentrant.

More info

Macro [Bidly_Eval0\(s\)](#) is defined for use with anonymous manager.

Definition at line 171 of file bidlyInOut.c.

3.1.3.41 Bidly_Managed_Eval1x()

```
EXTERN Bidly_Edge Bidly_Managed_Eval1x (
    Bidly_Manager MNG,
    Bidly_String s,
    Bidly_LookupFunction lf )
```

Description

Parameter `If` is a lookup function in the user-defined cache table.

Side effects

Not reentrant.

More info

Macro `Biddy_Eval1x(s,If)` is defined for use with anonymous manager. Macros `Biddy_Managed_Eval1(s)` and `Biddy_Eval1(s)` are defined for use without searching in the user-defined cache.

Definition at line 248 of file `biddyInOut.c`.

3.1.3.42 Biddy_Managed_Eval2()

```
EXTERN Biddy_Edge Biddy_Managed_Eval2 (
    Biddy_Manager MNG,
    Biddy_String boolFunc )
```

Description

Parenthesis are implemented. Operators' priority is implemented. Formula Tree is supported (global table, only). Boolean constants are '0' and '1'. Boolean operators are NOT (~!), AND (&*), OR (|+), XOR (^%), XNOR (-), IMPLIES (><), NAND (@), NOR (#), BUTNOT (), NOTBUT (/).

Side effects**More info**

Original author: Volodymyr Mihav (mihaw.wolodymyr@gmail.com) Original implementation of this function is on <https://github.com/sungmaster/libDD>. Macro `Biddy_Eval2(boolFunc)` is defined for use with anonymous manager.

Definition at line 347 of file `biddyInOut.c`.

3.1.3.43 Biddy_Managed_EvalProbability()

```
EXTERN double Biddy_Managed_EvalProbability (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Each variable should be associated with a probability q (0-1) via data element in `BiddyVariable`. For each 1-path a product of variable's probability is calculated (q for 'then' successor and $(1-q)$ for 'else' successor). The result of the function is the sum of all such products.

Side effects

Implemented for OBDD, ZBDD, and TZBDD.

More info

Macro [Bidly_EvalProbability\(f\)](#) is defined for use with anonymous manager.

Definition at line 2388 of file bidlyMain.c.

3.1.3.44 Bidly_Managed_ExistAbstract()

```
EXTERN Bidly_Edge Bidly_Managed_ExistAbstract (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge cube )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro [Bidly_ExistAbstract\(f,cube\)](#) is defined for use with anonymous manager.

Definition at line 3987 of file bidlyOp.c.

3.1.3.45 Bidly_Managed_ExtractMinterm()

```
EXTERN Bidly_Edge Bidly_Managed_ExtractMinterm (
    Bidly_Manager MNG,
    Bidly_Edge support,
    Bidly_Edge f )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Cache table is not used.

More info

Macro [Bidly_ExtractMinterm\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 8067 of file bidlyOp.c.

3.1.3.46 Biddy_Managed_FindFormula()

```
EXTERN Biddy_Boolean Biddy_Managed_FindFormula (
    Biddy_Manager MNG,
    Biddy_String x,
    unsigned int * idx,
    Biddy_Edge * f )
```

Description

Return TRUE/FALSE, index, and the formula. If formula is constant or variable then `idx = 0` and `f != biddyNull`. If formula is not found then `idx` is a position where the formulae should exist and `f == biddyNull`.

Side effects

More info

Macro `Biddy_FindFormula(x,f)` is defined for use with anonymous manager.

Definition at line 4790 of file `biddyMain.c`.

3.1.3.47 Biddy_Managed_FoaVariable()

```
EXTERN Biddy_Variable Biddy_Managed_FoaVariable (
    Biddy_Manager MNG,
    Biddy_String x,
    Biddy_Boolean varelem )
```

Description

If variable/element already exists, function returns the existing one. If `x == NULL` then numbered variable/element is added. Numbered variables/elements have only digits in its name. The current number of numbered variables/elements is stored in `numnum`. If numbered variable/element is requested then function increments `numnum` and creates a new (non-existing) variable/element. Parameter `varelem` is used to determine how to adapt the existing BDD base to keep the current formula valid (use `varelem = TRUE` if formulae represent Boolean functions and `varelem = FALSE` if they represent combination sets). The ordering of the new variable/element is determined in `Biddy_InitMNG`. Function always returns variable.

Side effects

Adding new variable/element may change the meaning of the existing BDDs. Variables and elements are always repaired. Formulae are repaired with regards to the parameter `varelem`. BDDs without external references are not repaired. For OBDDs, OFDDs, TZBDDs, and TZFDDs, it is safe to add new variables/elements if BDDs are used to represent Boolean functions. For ZBDDs and ZFDDs, it is safe to add new variables/elements if BDDs are used to represent combination sets. User should not add numbered variables/elements with some other function. TO DO: Formulae in user's formula tables are not repaired, yet! TO DO: Variables cannot be deleted, yet!

More info

Macro `Biddy_FoaVariable(x)` is defined for use with anonymous manager.

Definition at line 2573 of file `biddyMain.c`.

3.1.3.48 Bidly_Managed_FormulaTableNum()

```
EXTERN unsigned int Bidly_Managed_FormulaTableNum (
    Bidly_Manager MNG )
```

Description**Side effects**

Formulae '0' and '1' are included.

More info

Macro [Bidly_FormulaTableNum\(\)](#) is defined for use with anonymous manager.

Definition at line 917 of file `bidlyStat.c`.

3.1.3.49 Bidly_Managed_GC()

```
EXTERN void Bidly_Managed_GC (
    Bidly_Manager MNG,
    Bidly_Variable targetLT,
    Bidly_Variable targetGEQ,
    Bidly_Boolean purge,
    Bidly_Boolean total )
```

Description

All obsolete nodes are deleted. Parameter `purge` should not be true during automatic garbage collection. If parameter `purge` is true then all formulae without name are deleted. If parameter `purge` is true then all nodes which are not part of non-obsolete non-deleted formulae are removed even if they are fresh or fortified. If parameter `total` is true then all unnecessary nodes are immediately deleted, otherwise they are deleted only when there are enough of them. If (`targetLT != 0`) then node table resizing is disabled. If (`targetLT != 0`) then there should not exist obsolete formulae. If (`targetLT != 0`) then there should not exist obsolete nodes which are not part of any non-obsolete non-deleted formulae. If (`targetLT != 0`) then there should not exist obsolete nodes with variable equal or higher (bottom-more) than `target` and smaller (top-more) than `targetGEQ`.

Side effects

The first element of each chain in a node table should have a special value for its 'prev' element to allow tricky but efficient deleting. Moreover, 'prev' and 'next' should be the first and the second element in the structure `Bidly_↔Node`, respectively. Garbage collection is reported by `bidlyNodeTable.garbage` only if some bad nodes are purged! Parameters `targetLT` and `targetGEQ` are used during sifting, only, in all other cases 0 is used.

More info

Macro [Bidly_GC\(targetLT,targetGEQ,purge,total\)](#) is defined for use with anonymous manager. Macros [Bidly_↔Managed_AutoGC\(MNG\)](#) and [Bidly_AutoGC\(\)](#) are useful variants with `targetLT = targetGEQ = 0`, `purge = FA↔LSE`, and `total = FALSE`. Macros [Bidly_Managed_ForceGC\(MNG\)](#) and [Bidly_ForceGC\(\)](#) are useful variants with `targetLT = targetGEQ = 0`, `purge = FALSE`, and `total = TRUE`.

Definition at line 3788 of file `bidlyMain.c`.

3.1.3.50 Biddy_Managed_GetBaseSet()

```
EXTERN Biddy_Edge Biddy_Managed_GetBaseSet (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_GetBaseSet\(\)](#) is defined for use with anonymous manager.

Definition at line 1668 of file `biddyMain.c`.

3.1.3.51 Biddy_Managed_GetConstantOne()

```
EXTERN Biddy_Edge Biddy_Managed_GetConstantOne (
    Biddy_Manager MNG )
```

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 1 coincides with universal set.

Side effects

For ZBDDs and ZFDDs, you should always obtain constant 1 through the call of this function!

More info

Internally, use macro `biddyOne` (also for ZBDDs and ZFDDs!). Macro [Biddy_GetConstantOne\(\)](#) is defined for use with anonymous manager. Macros `Biddy_Managed_GetUniversalSet(MNG)` and `Biddy_GetUniversalSet()` are defined for manipulation of combination sets.

Definition at line 1642 of file `biddyMain.c`.

3.1.3.52 Biddy_Managed_GetConstantZero()

```
EXTERN Biddy_Edge Biddy_Managed_GetConstantZero (
    Biddy_Manager MNG )
```

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 0 coincides with empty set.

Side effects**More info**

Internally, use macro `bidlyZero`. Macro `Bidly_GetConstantZero()` is defined for use with anonymous manager. Macros `Bidly_Managed_GetEmptySet(MNG)` and `Bidly_GetEmptySet()` are defined for manipulation of combination sets.

Definition at line 1610 of file `bidlyMain.c`.

3.1.3.53 Bidly_Managed_GetElementEdge()

```
EXTERN Bidly_Edge Bidly_Managed_GetElementEdge (  
    Bidly_Manager MNG,  
    Bidly_Variable v )
```

Description**Side effects****More info**

Macro `Bidly_GetElementEdge(v)` is defined for use with anonymous manager.

Definition at line 1947 of file `bidlyMain.c`.

3.1.3.54 Bidly_Managed_GetIthFormula()

```
EXTERN Bidly_Edge Bidly_Managed_GetIthFormula (  
    Bidly_Manager MNG,  
    unsigned int i )
```

Description

Return `bidlyNull` if `ith` formulae does not exist.

Side effects

After adding new formula the index of others may change!

More info

Macro `Bidly_GetIthFormula(i)` is defined for use with anonymous manager.

Definition at line 5087 of file `bidlyMain.c`.

3.1.3.55 Biddy_Managed_GetIthFormulaName()

```
EXTERN Biddy_String Biddy_Managed_GetIthFormulaName (
    Biddy_Manager MNG,
    unsigned int i )
```

Description

Return NULL if ith formulae does not exist.

Side effects

After adding new formula the index of others may change!

More info

Macro [Biddy_GetIthFormulaName\(i\)](#) is defined for use with anonymous manager.

Definition at line 5118 of file biddyMain.c.

3.1.3.56 Biddy_Managed_GetIthVariable()

```
EXTERN Biddy_Variable Biddy_Managed_GetIthVariable (
    Biddy_Manager MNG,
    Biddy_Variable i )
```

Description

The lowest (topmost) variable has global ordering 1. The highest (bottommost) variable is '1' and has global ordering equal to numUsedVariables.

Side effects

If argument is 0, function returns 0. If argument is larger than the number of variables, function returns 0.

More info

Macro [Biddy_GetIthVariable\(x\)](#) is defined for use with anonymous manager.

Definition at line 1825 of file biddyMain.c.

3.1.3.57 Biddy_Managed_GetLowestVariable()

```
EXTERN Biddy_Variable Biddy_Managed_GetLowestVariable (
    Biddy_Manager MNG )
```

Description

The lowest variable is the tompost variable.

Side effects**More info**

Macro [Bidly_GetLowestVariable\(x\)](#) is defined for use with anonymous manager.

Definition at line 1787 of file bidlyMain.c.

3.1.3.58 Bidly_Managed_GetManagerName()

```
EXTERN Bidly_String Bidly_Managed_GetManagerName (
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_GetManagerName\(\)](#) is defined for use with anonymous manager.

Definition at line 1176 of file bidlyMain.c.

3.1.3.59 Bidly_Managed_GetManagerType()

```
EXTERN int Bidly_Managed_GetManagerType (
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_GetManagerType\(\)](#) is defined for use with anonymous manager.

Definition at line 1150 of file bidlyMain.c.

3.1.3.60 Bidly_Managed_GetNextVariable()

```
EXTERN Bidly_Variable Bidly_Managed_GetNextVariable (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description**Side effects****More info**

Macro [Bidly_GetNextVariable\(v\)](#) is defined for use with anonymous manager.

Definition at line 1893 of file `biddyMain.c`.

3.1.3.61 Bidly_Managed_GetPrevVariable()

```
EXTERN Bidly_Variable Bidly_Managed_GetPrevVariable (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description**Side effects****More info**

Macro [Bidly_GetPrevVariable\(v\)](#) is defined for use with anonymous manager.

Definition at line 1863 of file `biddyMain.c`.

3.1.3.62 Bidly_Managed_GetTerminal()

```
EXTERN Bidly_Edge Bidly_Managed_GetTerminal (
    Bidly_Manager MNG )
```

Description

Terminal node depends on a manager.

Side effects**More info**

Internally, use macro `biddyTerminal`. Macro [Bidly_GetTerminal\(\)](#) is defined for use with anonymous manager.

Definition at line 1580 of file `biddyMain.c`.

3.1.3.63 Bidly_Managed_GetTopVariableChar()

```
EXTERN char Bidly_Managed_GetTopVariableChar (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Side effects

More info

Macro `Bidly_GetTopVariableChar(f)` is defined for use with anonymous manager.

Definition at line 2056 of file `bidlyMain.c`.

3.1.3.64 `Bidly_Managed_GetTopVariableEdge()`

```
EXTERN Bidly_Edge Bidly_Managed_GetTopVariableEdge (  
    Bidly_Manager MNG,  
    Bidly_Edge f )
```

Description

Side effects

TO DO: For ZBDDs, element edge is sometimes preferred over variable edge.

More info

Macro `Bidly_GetTopVariableEdge(f)` is defined for use with anonymous manager.

Definition at line 2000 of file `bidlyMain.c`.

3.1.3.65 `Bidly_Managed_GetTopVariableName()`

```
EXTERN Bidly_String Bidly_Managed_GetTopVariableName (  
    Bidly_Manager MNG,  
    Bidly_Edge f )
```

Description

Side effects

More info

Macro `Bidly_GetTopVariableName(f)` is defined for use with anonymous manager.

Definition at line 2028 of file `bidlyMain.c`.

3.1.3.66 Biddy_Managed_GetVariable()

```
EXTERN Biddy_Variable Biddy_Managed_GetVariable (
    Biddy_Manager MNG,
    Biddy_String x )
```

Description

Side effects

If variable is not found function returns 0!

More info

Macro [Biddy_GetVariable\(x\)](#) is defined for use with anonymous manager.

Definition at line 1733 of file `biddyMain.c`.

3.1.3.67 Biddy_Managed_GetVariableData()

```
EXTERN void* Biddy_Managed_GetVariableData (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

Side effects

It is not checked that the given variable is valid.

More info

Macro [Biddy_GetVariableData\(v\)](#) is defined for use with anonymous manager.

Definition at line 2227 of file `biddyMain.c`.

3.1.3.68 Biddy_Managed_GetVariableEdge()

```
EXTERN Biddy_Edge Biddy_Managed_GetVariableEdge (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

Side effects

More info

Macro [Biddy_GetVariableEdge\(v\)](#) is defined for use with anonymous manager.

Definition at line 1922 of file `biddyMain.c`.

3.1.3.69 **Biddy_Managed_GetVariableName()**

```
EXTERN Biddy_String Biddy_Managed_GetVariableName (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

Side effects

More info

Macro [Biddy_GetVariableName\(v\)](#) is defined for use with anonymous manager.

Definition at line 1972 of file `biddyMain.c`.

3.1.3.70 **Biddy_Managed_GetVariableValue()**

```
EXTERN Biddy_Edge Biddy_Managed_GetVariableValue (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

Side effects

It is not checked that the given variable is valid.

More info

Macro [Biddy_GetVariableValue\(v\)](#) is defined for use with anonymous manager.

Definition at line 2141 of file `biddyMain.c`.

3.1.3.71 **Biddy_Managed_Gt()**

```
EXTERN Biddy_Edge Biddy_Managed_Gt (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

Boolean function $gt(f,g) = \text{and}(f,\text{not}(g)) = \text{not}(\text{leq}(f,g))$. For combination sets, this function coincides with Diff.

Side Effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TzBDD. For OBDDC, results are cached as parameters to $\text{ITE}(F,G,H) = F * G \text{ XOR } F' * H$. For all other BDD types, results are cached as (f,g,g) .

More Info

Macro [Bidly_Gt\(f,g\)](#) is defined for use with anonymous manager. Macros [Bidly_Managed_Diff\(MNG,f,g\)](#) and [Bidly_Diff\(f,g\)](#) are defined for manipulation of combination sets.

Definition at line 2767 of file `bidlyOp.c`.

3.1.3.72 Bidly_Managed_IncTag()

```
EXTERN Bidly_Edge Bidly_Managed_IncTag (  
    Bidly_Manager MNG,  
    Bidly_Edge f )
```

Description

Used for TZBDDs and TZFDDs, only.

Side effects

It is not checked, that the resulting tag is not greater than top variable. Function may return non-fresh node even if `f` is fresh.

More info

Macro [Bidly_IncTag\(\)](#) is defined for use with anonymous manager.

Definition at line 3187 of file `bidlyMain.c`.

3.1.3.73 Bidly_Managed_IsEqv()

```
EXTERN Bidly_Boolean Bidly_Managed_IsEqv (  
    Bidly_Manager MNG1,  
    Bidly_Edge f1,  
    Bidly_Manager MNG2,  
    Bidly_Edge f2 )
```

Description

It is assumed that `f1` and `f2` have the same ordering.

Side effects

More info

Macro [Bidly_IsEqv\(f1,MNG2,f2\)](#) is defined for use with anonymous manager.

Definition at line 1365 of file `bidlyMain.c`.

3.1.3.74 Bidly_Managed_IsHighest()

```
EXTERN Bidly_Boolean Bidly_Managed_IsHighest (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Side effects

More info

Macro BidlyIsHighest(v) is defined for internal use. Macro [Bidly_IsHighest\(v\)](#) is defined for use with anonymous manager.

Definition at line 2515 of file bidlyMain.c.

3.1.3.75 Bidly_Managed_IsLeq()

```
EXTERN Bidly_Boolean Bidly_Managed_IsLeq (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

Side Effects

Prototyped for OBDDs, ZBDDs, and TZBDDs (via calculating full implication, this is less efficient as implementation in CUDD).

More Info

Macro [Bidly_IsLeq\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 3091 of file bidlyOp.c.

3.1.3.76 Bidly_Managed_IsLowest()

```
EXTERN Bidly_Boolean Bidly_Managed_IsLowest (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description**Side effects****More info**

Macro `BiddyIsLowest(v)` is defined for internal use. Macro `Biddy_IsLowest(v)` is defined for use with anonymous manager.

Definition at line 2478 of file `biddyMain.c`.

3.1.3.77 Biddy_Managed_IsOK()

```
EXTERN Biddy_Boolean Biddy_Managed_IsOK (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

This is needed for implementation of user caches.

Side effects**More info**

Macro `BiddyIsOK(f)` is defined for debugging. It will check more properties and not only the expiry value. Macro `Biddy_IsOK(f)` is defined for use with anonymous manager.

Definition at line 3736 of file `biddyMain.c`.

3.1.3.78 Biddy_Managed_IsSelected()

```
EXTERN Biddy_Boolean Biddy_Managed_IsSelected (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects****More info**

Macro `Biddy_IsSelected(f)` is defined for use with anonymous manager.

Definition at line 1452 of file `biddyMain.c`.

3.1.3.79 Biddy_Managed_IsSmaller()

```
EXTERN Biddy_Boolean Biddy_Managed_IsSmaller (
    Biddy_Manager MNG,
    Biddy_Variable fv,
    Biddy_Variable gv )
```

Description

Side effects

More info

Macro BiddyIsSmaller(fv,gv) is defined for internal use. Macro [Biddy_IsSmaller\(fv,gv\)](#) is defined for use with anonymous manager.

Definition at line 2451 of file biddyMain.c.

3.1.3.80 Biddy_Managed_IsVariableDependent()

```
EXTERN Biddy_Boolean Biddy_Managed_IsVariableDependent (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Variable v )
```

Description

A variable is dependent on others in a function iff universal quantification of this variable returns constant FALSE.

Side effects

Prototyped for OBDDs (via xA, calculating full universal quantification is less efficient as direct implementation in CUDD).

More info

Macro [Biddy_IsVariableDependent\(f,v\)](#) is defined for use with anonymous manager.

Definition at line 3924 of file biddyOp.c.

3.1.3.81 Biddy_Managed_ITE()

```
EXTERN Biddy_Edge Biddy_Managed_ITE (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g,
    Biddy_Edge h )
```

Description**Side Effects**

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F * H$. For all other BDD types, results are cached as (f,g,h) where $f,g,h \neq 0$, $f \neq g$, $f \neq h$, and $g \neq h$.

More info

Macro `Biddy_ITE(f,g,h)` is defined for use with anonymous manager.

Definition at line 370 of file `biddyOp.c`.

3.1.3.82 Biddy_Managed_Leq()

```
EXTERN Biddy_Edge Biddy_Managed_Leq (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

Boolean function $leq(f,g) = or(not(f),g) = not(gt(f,g))$. This function coincides with implication $f \rightarrow g$.

Side Effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F * H$. For all other BDD types, results are cached as (f,f,g) .

More Info

Macro `Biddy_Leq(f,g)` is defined for use with anonymous manager.

Definition at line 2452 of file `biddyOp.c`.

3.1.3.83 Biddy_Managed_ListAvgLength()

```
EXTERN float Biddy_Managed_ListAvgLength (
    Biddy_Manager MNG )
```

Description**Side effects****More info**

Macro `Biddy_ListAvgLength()` is defined for use with anonymous manager.

Definition at line 1039 of file `biddyStat.c`.

3.1.3.84 Biddy_Managed_ListMaxLength()

```
EXTERN unsigned int Biddy_Managed_ListMaxLength (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_ListMaxLength\(\)](#) is defined for use with anonymous manager.

Definition at line 975 of file `biddyStat.c`.

3.1.3.85 Biddy_Managed_ListUsed()

```
EXTERN unsigned int Biddy_Managed_ListUsed (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_ListUsed\(\)](#) is defined for use with anonymous manager.

Definition at line 942 of file `biddyStat.c`.

3.1.3.86 Biddy_Managed_MaximizeBDD()

```
EXTERN void Biddy_Managed_MaximizeBDD (
    Biddy_Manager MNG,
    Biddy_String name )
```

Description

Steinhaus–Johnson–Trotter algorithm is used to generate all possible permutations. An optimized version of Bubble Sort is used to setup the final ordering. Variables are reordered globally. All obsolete nodes will be removed.

Side effects

Indeed, this function may take a lot of time! For TZBDD, all unreferenced nodes (not part of registered formulae) will be removed. For TZBDD, this function may change top edge or even a top node of any function/formula - this is a problem, because functions referenced by local variables only may become wrong. Consequently, for TZBDDs, sifting is not safe to start automatically!

More info

Macro `Biddy_MaximizeBDD(f)` is defined for use with anonymous manager.

Definition at line 5719 of file `biddyMain.c`.

3.1.3.87 Biddy_Managed_MaxNodes()

```
EXTERN unsigned int Biddy_Managed_MaxNodes (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

BDD is copied into new empty manager and then Steinhaus–Johnson–Trotter algorithm is used to check the node number for all possible orderings.

Side effects

Function will finish in a good time only for small number of variables.

More info

Macro `Biddy_MaxNodes()` is defined for use with anonymous manager.

Definition at line 1995 of file `biddyStat.c`.

3.1.3.88 Biddy_Managed_MinimizeBDD()

```
EXTERN void Biddy_Managed_MinimizeBDD (
    Biddy_Manager MNG,
    Biddy_String name )
```

Description

Steinhaus–Johnson–Trotter algorithm is used to generate all possible permutations. An optimized version of Bubble Sort is used to setup the final ordering. Variables are reordered globally. All obsolete nodes will be removed.

Side effects

Indeed, this function may take a lot of time! For TZBDD, all unreferenced nodes (not part of registered formulae) will be removed. For TZBDD, this function may change top edge or even a top node of any function/formula - this is a problem, because functions referenced by local variables only may become wrong. Consequently, for TZBDDs, sifting is not safe to start automatically!

More info

Macro [Bidly_MinimizeBDD\(f\)](#) is defined for use with anonymous manager.

Definition at line 5620 of file bidlyMain.c.

3.1.3.89 Bidly_Managed_MinNodes()

```
EXTERN unsigned int Bidly_Managed_MinNodes (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

BDD is copied into new empty manager and then Steinhaus–Johnson–Trotter algorithm is used to check the node number for all possible orderings.

Side effects

Function will finish in a good time only for small number of variables.

More info

Macro [Bidly_MinNodes\(\)](#) is defined for use with anonymous manager.

Definition at line 1852 of file bidlyStat.c.

3.1.3.90 Bidly_Managed_Nand()

```
EXTERN Bidly_Edge Bidly_Managed_Nand (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description**Side Effects**

Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD (via and-not). For $O \leftrightarrow$ BDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F' * H$. For OBDD, ZBDD, ZBDDC and TZBDD, results could be cached as (f,g,bidlyNull).

More Info

Macro [Bidly_Nand\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 1718 of file bidlyOp.c.

3.1.3.91 Biddy_Managed_NodeTableAddNumber()

```
EXTERN unsigned long long int Biddy_Managed_NodeTableAddNumber (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableAddNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 538 of file bidyStat.c.

3.1.3.92 Biddy_Managed_NodeTableANDORNumber()

```
EXTERN unsigned int Biddy_Managed_NodeTableANDORNumber (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableANDORNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 798 of file bidyStat.c.

3.1.3.93 Biddy_Managed_NodeTableANDORRecursiveNumber()

```
EXTERN unsigned long long int Biddy_Managed_NodeTableANDORRecursiveNumber (
    Biddy_Manager MNG )
```

Description

Side effects

Recursive AND/OR calls are counted only if Biddy is compiled using directive `BIDDYEXTENDEDSTATS_YES`.

More info

Macro [Biddy_NodeTableANDORRecursiveNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 826 of file bidyStat.c.

3.1.3.94 Bidly_Managed_NodeTableBlockNumber()

```
EXTERN unsigned int Bidly_Managed_NodeTableBlockNumber (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableBlockNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 287 of file bidlyStat.c.

3.1.3.95 Bidly_Managed_NodeTableCompareNumber()

```
EXTERN unsigned long long int Bidly_Managed_NodeTableCompareNumber (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableCompareNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 506 of file bidlyStat.c.

3.1.3.96 Bidly_Managed_NodeTableDRTime()

```
EXTERN unsigned int Bidly_Managed_NodeTableDRTime (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableDRTime\(\)](#) is defined for use with anonymous manager.

Definition at line 713 of file bidlyStat.c.

3.1.3.97 Biddy_Managed_NodeTableFindNumber()

```
EXTERN unsigned long long int Biddy_Managed_NodeTableFindNumber (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableFindNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 474 of file `biddyStat.c`.

3.1.3.98 Biddy_Managed_NodeTableFoaNumber()

```
EXTERN unsigned long long int Biddy_Managed_NodeTableFoaNumber (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableFoaNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 442 of file `biddyStat.c`.

3.1.3.99 Biddy_Managed_NodeTableGCNumber()

```
EXTERN unsigned int Biddy_Managed_NodeTableGCNumber (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableGCNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 569 of file `biddyStat.c`.

3.1.3.100 Bidly_Managed_NodeTableGCObsoleteNumber()

```
EXTERN unsigned long long int Bidly_Managed_NodeTableGCObsoleteNumber (
    Bidly_Manager MNG )
```

Description

Return the number of nodes deleted by GC.

Side effects

Obsolete nodes deleted by GC are counted only if Bidly is compiled using directive `BIDDYEXTENDEDSTATS_↔` YES.

More info

Macro [Bidly_NodeTableGCObsoleteNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 623 of file `bidlyStat.c`.

3.1.3.101 Bidly_Managed_NodeTableGCTime()

```
EXTERN unsigned int Bidly_Managed_NodeTableGCTime (
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_NodeTableGCTime\(\)](#) is defined for use with anonymous manager.

Definition at line 594 of file `bidlyStat.c`.

3.1.3.102 Bidly_Managed_NodeTableGenerated()

```
EXTERN unsigned int Bidly_Managed_NodeTableGenerated (
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_NodeTableGenerated\(\)](#) is defined for use with anonymous manager.

Definition at line 312 of file `bidlyStat.c`.

3.1.3.103 Bidly_Managed_NodeTableITENumber()

```
EXTERN unsigned int Bidly_Managed_NodeTableITENumber (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableITENumber\(\)](#) is defined for use with anonymous manager.

Definition at line 738 of file `biddyStat.c`.

3.1.3.104 Bidly_Managed_NodeTableITERecursiveNumber()

```
EXTERN unsigned long long int Bidly_Managed_NodeTableITERecursiveNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

Recursive ITE calls are counted only if Bidly is compiled using directive `BIDDYEXTENDEDSTATS_YES`.

More info

Macro [Bidly_NodeTableITERecursiveNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 766 of file `biddyStat.c`.

3.1.3.105 Bidly_Managed_NodeTableMax()

```
EXTERN unsigned int Bidly_Managed_NodeTableMax (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableMax\(\)](#) is defined for use with anonymous manager.

Definition at line 338 of file `biddyStat.c`.

3.1.3.106 Bidly_Managed_NodeTableNum()

```
EXTERN unsigned int Bidly_Managed_NodeTableNum (  
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_NodeTableNum\(\)](#) is defined for use with anonymous manager.

Definition at line 364 of file `bidlyStat.c`.

3.1.3.107 Bidly_Managed_NodeTableNumVar()

```
EXTERN unsigned int Bidly_Managed_NodeTableNumVar (  
    Bidly_Manager MNG,  
    Bidly_Variable v )
```

Description**Side effects****More info**

Macro [Bidly_NodeTableNumVar\(v\)](#) is defined for use with anonymous manager.

Definition at line 390 of file `bidlyStat.c`.

3.1.3.108 Bidly_Managed_NodeTableResizeNumber()

```
EXTERN unsigned int Bidly_Managed_NodeTableResizeNumber (  
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_NodeTableResizeNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 416 of file `bidlyStat.c`.

3.1.3.109 Bidly_Managed_NodeTableSiftingNumber()

```
EXTERN unsigned int Bidly_Managed_NodeTableSiftingNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableSiftingNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 688 of file bidlyStat.c.

3.1.3.110 Bidly_Managed_NodeTableSize()

```
EXTERN unsigned int Bidly_Managed_NodeTableSize (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableSize\(\)](#) is defined for use with anonymous manager.

Definition at line 257 of file bidlyStat.c.

3.1.3.111 Bidly_Managed_NodeTableSwapNumber()

```
EXTERN unsigned int Bidly_Managed_NodeTableSwapNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableSwapNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 662 of file bidlyStat.c.

3.1.3.112 Biddy_Managed_NodeTableXORNumber()

```
EXTERN unsigned int Biddy_Managed_NodeTableXORNumber (
    Biddy_Manager MNG )
```

Description**Side effects****More info**

Macro [Biddy_NodeTableXORNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 857 of file biddyStat.c.

3.1.3.113 Biddy_Managed_NodeTableXORRecursiveNumber()

```
EXTERN unsigned long long int Biddy_Managed_NodeTableXORRecursiveNumber (
    Biddy_Manager MNG )
```

Description**Side effects**

Recursive XOR calls are counted only if Biddy is compiled using directive BIDDYEXTENDEDSTATS_YES.

More info

Macro [Biddy_NodeTableXORRecursiveNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 885 of file biddyStat.c.

3.1.3.114 Biddy_Managed_Nor()

```
EXTERN Biddy_Edge Biddy_Managed_Nor (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description**Side Effects**

Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD (via or-not). For $O \leftrightarrow$ BDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F' * H$. For OBDD, ZBDD, ZBDDC and TZBDD, results could be cached as (biddyNull,f,g).

More Info

Macro [Bidly_Nor\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 1816 of file `bidlyOp.c`.

3.1.3.115 Bidly_Managed_Not()

```
EXTERN Bidly_Edge Bidly_Managed_Not (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC and OFDDC, it is better to use macro `Bidly_Inv`. For OBDDC, cache table is not needed. For ZBDD and ZBDDC, recursive calls are via `Xor`, its cache table is used. For OBDD and TZBDD, results are cached as `(f,bidlyZero,bidlyOne)`.

More info

Macro [Bidly_Not\(\)](#) is defined for use with anonymous manager.

Definition at line 100 of file `bidlyOp.c`.

3.1.3.116 Bidly_Managed_OPCCacheFind()

```
EXTERN unsigned long long int Bidly_Managed_OPCCacheFind (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_OPCCacheFind\(\)](#) is defined for use with anonymous manager.

Definition at line 1106 of file `bidlyStat.c`.

3.1.3.117 Bidly_Managed_OPCCacheInsert()

```
EXTERN unsigned long long int Bidly_Managed_OPCCacheInsert (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro `Bidly_OPCCacheInsert()` is defined for use with anonymous manager.

Definition at line 1131 of file `bidlyStat.c`.

3.1.3.118 `Bidly_Managed_OPCCacheOverwrite()`

```
EXTERN unsigned long long int Bidly_Managed_OPCCacheOverwrite (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro `Bidly_OPCCacheOverwrite()` is defined for use with anonymous manager.

Definition at line 1162 of file `bidlyStat.c`.

3.1.3.119 `Bidly_Managed_OPCCacheSearch()`

```
EXTERN unsigned long long int Bidly_Managed_OPCCacheSearch (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro `Bidly_OPCCacheSearch()` is defined for use with anonymous manager.

Definition at line 1081 of file `bidlyStat.c`.

3.1.3.120 `Bidly_Managed_Or()`

```
EXTERN Bidly_Edge Bidly_Managed_Or (  
    Bidly_Manager MNG,  
    Bidly_Edge f,  
    Bidly_Edge g )
```

Description

For combination sets, this function coincides with Union.

Side Effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to ITE(F,G,H)= F*G XOR F'*H. For all other BDD types, results are cached as (biddyZero,f,g).

More Info

Macro [Biddy_Or\(f,g\)](#) is defined for use with anonymous manager. Macros [Biddy_Managed_Union\(MNG,f,g\)](#) and [Biddy_Union\(f,g\)](#) are defined for manipulation of combination sets.

Definition at line 1243 of file biddyOp.c.

3.1.3.121 Biddy_Managed_Permitsym()

```
EXTERN Biddy_Edge Biddy_Managed_Permitsym (
    Biddy_Manager MNG,
    Biddy_Edge f,
    unsigned int n )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Cache table is not used.

More info

Macro [Biddy_Permitsym\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 7261 of file biddyOp.c.

3.1.3.122 Biddy_Managed_PrintfBDD()

```
EXTERN void Biddy_Managed_PrintfBDD (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

This function is intended for writing to an output channel via macro which overrides the meaning of standard printf calls. For writing raw format into the file, use [Biddy_Managed_WriteBDD](#).

Side effects

More info

Macro `Biddy_PrintfBDD(f)` is defined for use with anonymous manager.

Definition at line 1080 of file `biddyInOut.c`.

3.1.3.123 `Biddy_Managed_PrintfSOP()`

```
EXTERN void Biddy_Managed_PrintfSOP (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Side effects

More info

Definition at line 1350 of file `biddyInOut.c`.

3.1.3.124 `Biddy_Managed_PrintfTable()`

```
EXTERN void Biddy_Managed_PrintfTable (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

This function is intended for writing to an output channel via macro which overrides the meaning of standard printf calls. For writing truth table into the file, use `Biddy_Managed_WriteTable`.

Side effects

More info

Thanks to Jan Kraner and Ziga Kobale for prototype implementation. Macro `Biddy_PrintfTable(f)` is defined for use with anonymous manager.

Definition at line 1156 of file `biddyInOut.c`.

3.1.3.125 `Biddy_Managed_PrintInfo()`

```
EXTERN void Biddy_Managed_PrintInfo (
    Biddy_Manager MNG,
    FILE * f )
```

Description**Side effects****More info**

Macro [Bidly_PrintInfo\(f\)](#) is defined for use with anonymous manager.

Definition at line 2234 of file bidlyStat.c.

3.1.3.126 Bidly_Managed_Product()

```
EXTERN Bidly_Edge Bidly_Managed_Product (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

Product is also called Multiplication.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TzBDD. RC Cache is used with parameters (f,g,bidlyTerminal).

More info

Macro [Bidly_Product\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 6307 of file bidlyOp.c.

3.1.3.127 Bidly_Managed_Purge()

```
EXTERN void Bidly_Managed_Purge (
    Bidly_Manager MNG )
```

Description

All formulae without name are deleted. All deleted formulae (including prolonged/fortified formulae) are removed. All fresh and obsolete nodes are immediately removed. Moreover, all prolonged and fortified nodes are immediately removed if they are not needed by some of the remaining formula. Call to Bidly_Purge does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

Removes all fresh nodes!

More info

Macro [Biddy_Purge\(f\)](#) is defined for use with anonymous manager.

Definition at line 4334 of file `biddyMain.c`.

3.1.3.128 Biddy_Managed_PurgeAndReorder()

```
EXTERN void Biddy_Managed_PurgeAndReorder (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Boolean converge )
```

Description

All obsolete nodes are immediately removed. Moreover, nodes from deleted prolonged formulae and nodes from deleted fortified formulae are removed if they are not needed by other formulae. If BDD is given ($f \neq \text{NULL}$), reordering on function is used. Otherwise ($f == \text{NULL}$) global reordering is used. Call to `Biddy_PurgeAndReorder` does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

Removes all fresh nodes.

More info

Macro [Biddy_PurgeAndReorder\(f\)](#) is defined for use with anonymous manager.

Definition at line 4382 of file `biddyMain.c`.

3.1.3.129 Biddy_Managed_RandomFunction()

```
EXTERN Biddy_Edge Biddy_Managed_RandomFunction (
    Biddy_Manager MNG,
    Biddy_Edge support,
    double r )
```

Description

The represented Boolean function depends on the variables given with parameter `support` whilst the parameter `r` determines the ratio between the number of function's minterms and the number of all possible minterms. Parameter `support` is a product of positive variables - it can be generated with function `Biddy_Support`.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Parameter `r` must be a number from $[0, 1]$. Otherwise, function returns `biddyNull`.

More info

Macro `Bidly_RandomFunction(support,r)` is defined for use with anonymous manager.

Definition at line 7767 of file `bidlyOp.c`.

3.1.3.130 Bidly_Managed_RandomSet()

```
EXTERN Bidly_Edge Bidly_Managed_RandomSet (
    Bidly_Manager MNG,
    Bidly_Edge unit,
    double r )
```

Description

The represented set is a random combination set determined by the parameter `unit` whilst the parameter `r` determines the ratio between the number of set's subsets and the number of all possible subsets. Parameter `unit` is a set containing a single subset which consist of all elements, i.e. it is a set $\{\{x_1, x_2, \dots, x_n\}\}$ - this is encoded as a product of positive variables and can be generated with function `Bidly_Support`.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Parameter `r` must be a number from $[0, 1]$. Otherwise, function returns `bidlyNull`.

More info

Macro `Bidly_RandomSet(unit,r)` is defined for use with anonymous manager.

Definition at line 7908 of file `bidlyOp.c`.

3.1.3.131 Bidly_Managed_ReadBddview()

```
EXTERN Bidly_String Bidly_Managed_ReadBddview (
    Bidly_Manager MNG,
    const char filename[],
    Bidly_String name )
```

Description

If `(name != NULL)` then `name` will be used for the resulting BDD. If `(name == NULL)` then the resulting BDD will have name given in the file. Resulting BDD will not be preserved.

Side effects

`Bidly_Managed_ConstructBDD` is used and thus, if variable ordering in the file is not compatible with the active ordering then the result will be wrong! To improve efficiency, only the first "type", "var", and "label" are used.

More info

Macro [Bidly_ReadBddview\(filename\)](#) is defined for use with anonymous manager.

Definition at line 543 of file `biddyInOut.c`.

3.1.3.132 Bidly_Managed_ReadMemoryInUse()

```
EXTERN unsigned long long int Bidly_Managed_ReadMemoryInUse (  
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_ReadMemoryInUse\(\)](#) is defined for use with anonymous manager.

Definition at line 2137 of file `biddyStat.c`.

3.1.3.133 Bidly_Managed_ReadVerilogFile()

```
EXTERN void Bidly_Managed_ReadVerilogFile (  
    Bidly_Manager MNG,  
    const char filename[],  
    Bidly_String prefix )
```

Description

If (`prefix != NULL`) then the created BDD variables and formulae will get it.

Side effects**More info**

Original author: David Kebo Houngrinou, Southern Methodist University Original implementation of this function is on <https://github.com/davidkebo/verilog-parser> Macro [Bidly_ReadVerilogFile\(filename,prefix\)](#) is defined for use with anonymous manager.

Definition at line 979 of file `biddyInOut.c`.

3.1.3.134 Bidly_Managed_Refresh()

```
EXTERN void Bidly_Managed_Refresh (  
    Bidly_Manager MNG,  
    Bidly_Edge f )
```

Description

This is an external variant of internal macro BiddyRefresh This is needed for implementing user caches.

Side effects**More info**

Macro [Biddy_Refresh\(f\)](#) is defined for use with anonymous manager.

Definition at line 4411 of file biddyMain.c.

3.1.3.135 Biddy_Managed_ReplaceByKeyword()

```
EXTERN Biddy_Edge Biddy_Managed_ReplaceByKeyword (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_String keyword )
```

Description

Original BDD is not changed. Implemented for OBDD, OBDDC, and TZBDD. Prototyped for ZBDD and ZBDDC (via And-Xor-Not-Restrict). Replacing is controlled by variable's values (which are edges!). For OBDDs and TZBD↔Ds control values must be variables, but for ZBDDs they must be elements! Use Biddy_ResetVariablesValue and Biddy_SetVariableValue to prepare control values. Parameter keyword is used to maintain cache table. If (keyword == NULL) then entries in the cache table from previous calculations are deleted.

Side effects

For ZBDD and ZBDDC the function is prototyped and not implemented, yet. Therefore, for ZBDD and ZBDDC, the sets of current and new variables should be disjoint.

More info

Macro [Biddy_ReplaceByKeyword\(f,keyword\)](#) is defined for use with anonymous manager. Macros Biddy_↔Managed_Replace(MNG,f) and Biddy_Replace(f) are variants with less effective cache table.

Definition at line 5265 of file biddyOp.c.

3.1.3.136 Biddy_Managed_ResetVariablesValue()

```
EXTERN void Biddy_Managed_ResetVariablesValue (
    Biddy_Manager MNG )
```

Description**Side effects**

Only active (used) variables are reinitialized.

More info

Macro [Biddy_ResetVariablesValue\(\)](#) is defined for use with anonymous manager.

Definition at line 2085 of file `biddyMain.c`.

3.1.3.137 Biddy_Managed_Restrict()

```
EXTERN Biddy_Edge Biddy_Managed_Restrict (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Variable v,
    Biddy_Boolean value )
```

Description

This is not Coudert and Madre's restrict function (use [Biddy_Simplify](#) if you need that one).

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDs, recursive calls use optimization: $F(a=x) == \text{NOT}(\text{NOT } F)(a=x)$.

More info

Macro [Biddy_Restrict\(f,v,value\)](#) is defined for use with anonymous manager.

Definition at line 3163 of file `biddyOp.c`.

3.1.3.138 Biddy_Managed_SelectFunction()

```
EXTERN void Biddy_Managed_SelectFunction (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects**

Terminal node must be selected before starting this function!

More info

Macro [Biddy_SelectFunction\(f\)](#) is defined for use with anonymous manager.

Definition at line 1479 of file `biddyMain.c`.

3.1.3.139 Biddy_Managed_SelectiveProduct()

```

EXTERN Biddy_Edge Biddy_Managed_SelectiveProduct (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g,
    Biddy_Edge cube )

```

Description

Selective product is also called Selective multiplication. Combinations are composed only if they agree on variables from cube. This is a non-commutative operation. Variables in Boolean function cube, which are presented in g, must exist in f while, viceversa, this is not required. Moreover, variables which are missing in Boolean function cube and are presented in g must not exist in f (they will be included in the result!) while, viceversa, this is not required. Combinations from f, which are not composed with any combination from g are not included in the result.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (f,g,cube). Because the usage of RC Cache coincide with functions Biddy_Restrict and Biddy_Compose they should not be used together with Biddy_SelectiveProduct in the same manager (the allowed exception is Restrict to zero).

More info

Macro [Biddy_SelectiveProduct\(f,g,cube\)](#) is defined for use with anonymous manager.

Definition at line 6614 of file biddyOp.c.

3.1.3.140 Biddy_Managed_SelectNode()

```

EXTERN void Biddy_Managed_SelectNode (
    Biddy_Manager MNG,
    Biddy_Edge f )

```

Description**Side effects****More info**

Macro [Biddy_SelectNode\(f\)](#) is defined for use with anonymous manager.

Definition at line 1400 of file biddyMain.c.

3.1.3.141 Biddy_Managed_SetAlphabeticOrdering()

```

EXTERN void Biddy_Managed_SetAlphabeticOrdering (
    Biddy_Manager MNG )

```

Description

Named variables are ordered according to their names. Numbered variables are ordered according to their numbers and are always smaller (topmore) as any named variable.

Side effects**More info**

Macro [Bidly_SetAlphabeticOrdering\(\)](#) is defined for use with anonymous manager.

Definition at line 5256 of file bidlyMain.c.

3.1.3.142 Bidly_Managed_SetManagerParameters()

```
EXTERN void Bidly_Managed_SetManagerParameters (
    Bidly_Manager MNG,
    float gcr,
    float gcrF,
    float gcrX,
    float rr,
    float rrF,
    float rrX,
    float st,
    float cst )
```

Description

Function expect 6 float values. If the value is < 0 then the parameter is not modified. The parameters are: bidlyNodeTable.gcratio (do not delete nodes if the effect is to small), bidlyNodeTable.gcratioF (do not delete nodes if the effect is to small), bidlyNodeTable.gcratioX (do not delete nodes if the effect is to small), bidlyNodeTable.resizeratio (resize Node table if there are to many nodes), bidlyNodeTable.resizeratioF (resize Node table if there are to many nodes), bidlyNodeTable.resizeratioX (resize Node table if there are to many nodes), bidlyNodeTable.siftingreshold (stop sifting if the size of the system grows to much), bidlyNodeTable.fsiftingreshold (stop sifting if the size of the function grows to much), bidlyNodeTable.convergesiftingreshold (stop one step of converging sifting if the size of the system grows to much), bidlyNodeTable.fconvergesiftingreshold (stop one step of converging sifting if the size of the function grows to much).

Side effects

Initial values are given in Bidly_InitMNG.

More info

Macro [Bidly_SetManagerParameters\(\)](#) is defined for use with anonymous manager.

Definition at line 1219 of file bidlyMain.c.

3.1.3.143 Biddy_Managed_SetOrdering()

```
EXTERN void Biddy_Managed_SetOrdering (
    Biddy_Manager MNG,
    Biddy_String ordering )
```

Description

Non-existing variables included in the string are simply ignored. Not all variables need to be given.

Side effects

String should be formatted in the same way as returned by Biddy_GetOrdering. It should not be prefixed with spaces.

More info

Macro [Biddy_SetOrdering\(ordering\)](#) is defined for use with anonymous manager.

Definition at line 5202 of file biddyMain.c.

3.1.3.144 Biddy_Managed_SetVariableData()

```
EXTERN void Biddy_Managed_SetVariableData (
    Biddy_Manager MNG,
    Biddy_Variable v,
    void * x )
```

Description

Side effects

It is not checked that the given variable is valid.

More info

Macro [Biddy_SetVariableData\(v,x\)](#) is defined for use with anonymous manager.

Definition at line 2201 of file biddyMain.c.

3.1.3.145 Biddy_Managed_SetVariableValue()

```
EXTERN void Biddy_Managed_SetVariableValue (
    Biddy_Manager MNG,
    Biddy_Variable v,
    Biddy_Edge f )
```

Description**Side effects**

It is not checked that the given variable is valid.

More info

Macro `Biddy_SetVariableValue(v,f)` is defined for use with anonymous manager.

Definition at line 2115 of file `biddyMain.c`.

3.1.3.146 `Biddy_Managed_Sifting()`

```
EXTERN Biddy_Boolean Biddy_Managed_Sifting (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Boolean converge )
```

Description

Variables are reordered globally. All obsolete nodes will be removed.

Side effects

For TZBDD, all unreferenced nodes (not part of registered formulae) will be removed. For TZBDD, sifting may change top edge or even a top node of any function/formula - this is a problem, because functions referenced by local variables only may become wrong. Consequently, for TZBDDs, sifting is not safe to start automatically!

More info

Macro `Biddy_Sifting(f)` is defined for use with anonymous manager.

Definition at line 5431 of file `biddyMain.c`.

3.1.3.147 `Biddy_Managed_Simplify()`

```
EXTERN Biddy_Edge Biddy_Managed_Simplify (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge c )
```

Description

Coudert and Madre's restrict function tries to simplify function `f` by restricting it to the domain covered by function `c`. No checks are done to see if the result is actually smaller than the input. Here is an explanation from <http://gauss.eecs.uc.edu/Courses/c626/lectures/BDD/bdd-desc.pdf> "Consider the truth tables corresponding to two BDDs `f` and `c` over the union of variable sets of both `f` and `c`. Build a new BDD `g` with variable set no larger than the union of the variable sets of `f` and `c` and with a truth table such that on rows which `c` maps to 1 `g` maps to the same value that `f` maps to, and on other rows `g` maps to any value, independent of `f`. It should be clear that $(f \text{ AND } c)$ and $(g \text{ AND } c)$ are identical so `g` can replace `f` in a collection of BDDs without changing its solution space."

Side effects

Original BDD is not changed. Implemented only for OBDD and OBDDC. Cache table is not used.

More info

Macro `Biddy_Simplify(f,c)` is defined for use with anonymous manager.

Definition at line 4925 of file `biddyOp.c`.

3.1.3.148 Biddy_Managed_Stretch()

```
EXTERN Biddy_Edge Biddy_Managed_Stretch (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (f,biddyNull,biddyNull).

More info

Macro `Biddy_Stretch(f,g)` is defined for use with anonymous manager.

Definition at line 7432 of file `biddyOp.c`.

3.1.3.149 Biddy_Managed_Subset()

```
EXTERN Biddy_Edge Biddy_Managed_Subset (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

The set `SubSet(F,G)` is the set of products of F that is contained in at least one product of G. For combination sets, function `Subset` coincides with a function which is called a Permission operation. It extracts the product terms from F such that the item combination is a subset of at least one item combination in G.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (f,g,biddyNull).

More info

Macro [Bidly_Subset\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 7076 of file bidlyOp.c.

3.1.3.150 Bidly_Managed_Support()

```
EXTERN Bidly_Edge Bidly_Managed_Support (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDs, dependent variables are exactly those variables existing in the graph. For ZBDDs and TZBDDs, this is not true.

Side effects

For ZBDDs and ZFDDs, variables above the top variable (which are always all dependent) are also included.

More info

Macro [Bidly_Support\(f\)](#) is defined for use with anonymous manager.

Definition at line 5065 of file bidlyOp.c.

3.1.3.151 Bidly_Managed_Supset()

```
EXTERN Bidly_Edge Bidly_Managed_Supset (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

The set $\text{SupSet}(F,G)$ is the set of products of F that contain at least one product of G . For combination sets, function Supset coincides with a function which is called a Restriction operation. It extracts the product terms from F such that the item combination is a superset of at least one item combination in G .

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (bidlyNull,f,g).

More info

Macro [Bidly_Supset\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 6855 of file bidlyOp.c.

3.1.3.152 Bidly_Managed_SwapWithHigher()

```
EXTERN Bidly_Variable Bidly_Managed_SwapWithHigher (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Higher (greater) variable is the bottommore one! The highest variable is constant variable "1". Global ordering is number of zeros in corresponding line of orderingTable. Constant variable '1' has global ordering greater than all others.

Side effects

All obsolete nodes will be removed.

More info

Macro [Bidly_SwapWithHigher\(v\)](#) is defined for use with anonymous manager.

Definition at line 5295 of file bidlyMain.c.

3.1.3.153 Bidly_Managed_SwapWithLower()

```
EXTERN Bidly_Variable Bidly_Managed_SwapWithLower (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Lower (smaller) variable is the topmore one! The lowest (topmost) element is not fixed. Topmost variable has global ordering 1 (smaller than all except itself). Global ordering is the number of zeros in corresponding line of orderingTable.

Side effects

All obsolete nodes will be removed.

More info

Macro [Bidly_SwapWithLower\(v\)](#) is defined for use with anonymous manager.

Definition at line 5361 of file bidlyMain.c.

3.1.3.154 Biddy_Managed_TaggedFoaNode()

```

EXTERN Biddy_Edge Biddy_Managed_TaggedFoaNode (
    Biddy_Manager MNG,
    Biddy_Variable v,
    Biddy_Edge pf,
    Biddy_Edge pt,
    Biddy_Variable ptag,
    Biddy_Boolean garbageAllowed )

```

Description

If such node already exists, function returns it and does not create the new one. For OBDDs, ZBDDs, OFDDs, and ZFDDs, the returned edge is not tagged (i.e. tag == 0). For TZBDDs and TZFDDs, the returned edge is tagged with the given ptag. There are two special cases:

1. If (pf == pt == NULL) then new variable (for OBDDs, OFDDs, TZBDDs, and TZFDDs) or new element (for ZBDDs and ZFDDs) is created.
2. (If ptag == 0) then the reduction rule and the normalization of complemented edges is not used and the node is added exactly as specified (be careful, this may create a wrong node!).

Side effects

This function should not be called directly to add new variables and elements, you must use `Biddy_Managed_↔_FoaVariable`, `Biddy_Managed_AddVariableByName`, or `Biddy_Managed_AddElementByName`. Using `Biddy_↔_Managed_TaggedFoaNode` you can create node with an arbitrary ordering. It is much more safe to use Boolean operators, e.g. `Biddy_Managed_ITE`.

More info

Macro `Biddy_Managed_FoaNode(MNG,v,pf,pt,garbageAllowed)` is defined for use without tags. Macros `Biddy_↔_TaggedFoaNode(v,pf,pt,tag,garbageAllowed)` and `Biddy_FoaNode(v,pf,pt,garbageAllowed)` are defined for use with anonymous manager.

Definition at line 3269 of file `biddyMain.c`.

3.1.3.155 Biddy_Managed_TransferMark()

```

EXTERN Biddy_Edge Biddy_Managed_TransferMark (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Boolean mark,
    Biddy_Boolean leftright )

```

Description

Parameter `leftright` should be TRUE for left and FALSE for right. For OBDDC, it is better to use macro `Biddy_Inv_↔_Cond`. For OBDDC, parameter `leftright` is ignored.

Side effects

TO DO: swap the meaning of parameter `leftright` (left should be FALSE)

More info

Macro [Bidly_TransferMark\(\)](#) is defined for use with anonymous manager.

Definition at line 3121 of file `bidlyMain.c`.

3.1.3.156 Bidly_Managed_UnivAbstract()

```
EXTERN Bidly_Edge Bidly_Managed_UnivAbstract (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge cube )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD.

More info

Macro [Bidly_UnivAbstract\(f,cube\)](#) is defined for use with anonymous manager.

Definition at line 4240 of file `bidlyOp.c`.

3.1.3.157 Bidly_Managed_VariableTableNum()

```
EXTERN Bidly_Variable Bidly_Managed_VariableTableNum (
    Bidly_Manager MNG )
```

Description**Side effects**

Variable '1' is included.

More info

Macro [Bidly_VariableTableNum\(\)](#) is defined for use with anonymous manager.

Definition at line 232 of file `bidlyStat.c`.

3.1.3.158 Bidly_Managed_VarSubset()

```

EXTERN Bidly_Edge Bidly_Managed_VarSubset (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v,
    Bidly_Boolean value )

```

Description

If (value == FALSE) then calculates $(f|_{\{v=0\}}) * (\text{NOT } v)$. If (value == TRUE) then calculates $(f|_{\{v=1\}}) * v$. This is NOT a Couderet and Madre's SubSet operation (which is also called a Permission operation and is a counterpart of a SupSet operation). For combination sets, function Subset coincides with functions Subset0 and Subset1. Moreover, function Offset (also called Modulo or Remainder) is the same as function Subset0, while function Onset (also called Division or Quotient) can be calculated as $\text{Change}(\text{Subset1}(f,v),v)$.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Cache table for AND is used.

More info

Macro `Bidly_VarSubset(f,v,value)` is defined for use with anonymous manager. Macros `Bidly_Managed_Subset0(MNG,f,v)`, `Bidly_Subset0(f,v)`, `Bidly_Managed_Subset1(MNG,f,v)`, and `Bidly_Subset1(f,v)` are defined for manipulation of combination sets. Macros `Bidly_Managed_Quotient(MNG,f,v)`, `Bidly_Quotient(f,v)`, `Bidly_Managed_Remainder(MNG,f,v)`, and `Bidly_Remainder(f,v)` are defined for manipulation of combination sets. Using the provided macros, `Bidly_Managed_Quotient` and `Bidly_Quotient` are not implemented optimally.

Definition at line 5891 of file `bidlyOp.c`.

3.1.3.159 Bidly_Managed_WriteBDD()

```

EXTERN void Bidly_Managed_WriteBDD (
    Bidly_Manager MNG,
    const char filename[],
    Bidly_Edge f,
    Bidly_String label )

```

Description**Side effects****More info**

Macro `Bidly_WriteBDD(f)` is defined for use with anonymous manager.

Definition at line 1110 of file `bidlyInOut.c`.

3.1.3.160 Biddy_Managed_WriteBddview()

```

EXTERN unsigned int Biddy_Managed_WriteBddview (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f,
    const char label[],
    void * xytable )

```

Description

Output bddview format. Parameter table is optional, if not NULL then it must contain node names and coordinates. If (filename == NULL) then output is to stdout.

Side effects

A conservative approach to list all the existing variables in the manager (and not only the dependent ones) is used. Variable names containing # are adapted. To support EST, also variable names containing <> are adapted. The optional table is type-casted to BiddyXYTable.

More info

Macro [Biddy_WriteBddview\(filename,f,label\)](#) is defined for use with anonymous manager.

Definition at line 1916 of file biddyInOut.c.

3.1.3.161 Biddy_Managed_WriteDot()

```

EXTERN unsigned int Biddy_Managed_WriteDot (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f,
    const char label[],
    int id,
    Biddy_Boolean cudd )

```

Description

Output dot format. Two approaches are implemented. The CUDD-like implementation is copied from CUDD 3.0.

Side effects

If (id != -1) then id is used instead of <...> for variable names. If (filename == NULL) then output is to stdout. Function resets all variables value. Variable names containing # are adapted.

More info

Macro [Biddy_WriteDot\(filename,f,label,id,cudd\)](#) is defined for use with anonymous manager.

Definition at line 1693 of file biddyInOut.c.

3.1.3.162 Biddy_Managed_WriteSOP()

```
EXTERN void Biddy_Managed_WriteSOP (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f )
```

Description

Side effects

More info

Definition at line 1434 of file biddyInOut.c.

3.1.3.163 Biddy_Managed_WriteTable()

```
EXTERN void Biddy_Managed_WriteTable (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f )
```

Description

Side effects

More info

Thanks to Jan Kraner and Ziga Kobale for prototype implementation. Macro [Biddy_WriteTable\(f\)](#) is defined for use with anonymous manager.

Definition at line 1253 of file biddyInOut.c.

3.1.3.164 Biddy_Managed_Xnor()

```
EXTERN Biddy_Edge Biddy_Managed_Xnor (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

Side Effects

Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD (via xor-not). For $O \leftrightarrow$ BDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F' * H$. For OBDD, ZBDD, ZBDDC and TZBDD, results could be cached as $(f, \text{biddyNull}, g)$.

More Info

Macro [Bidly_Xnor\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 2354 of file bidlyOp.c.

3.1.3.165 Bidly_Managed_Xor()

```
EXTERN Bidly_Edge Bidly_Managed_Xor (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description**Side Effects**

Used by ITE (for OBDDC). Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to ITE(F,G,H)= F*G XOR F'*H. For all other BDD types, results are cached as (f,bidlyZero,g).

More Info

Macro [Bidly_Xor\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 1913 of file bidlyOp.c.

3.1.3.166 Bidly_MaxLevel()

```
EXTERN unsigned int Bidly_MaxLevel (
    Bidly_Edge f )
```

Description**Side effects****More info**

Macro [Bidly_Managed_MaxLevel\(f\)](#) is defined for user convenience.

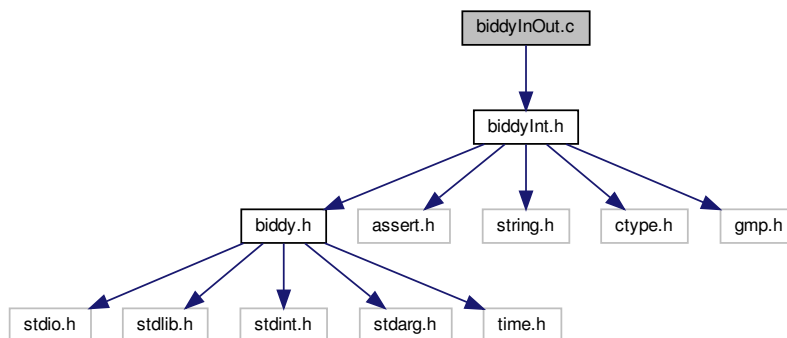
Definition at line 164 of file bidlyStat.c.

3.2 biddylnOut.c File Reference

File `biddylnOut.c` contains various parsers and generators.

```
#include "biddyInt.h"
```

Include dependency graph for `biddylnOut.c`:



Macros

- `#define LINESIZE 999` /* maximum length of each input line read */
- `#define BUFSIZE 99999` /* maximum length of a buffer */
- `#define INOUTNUM 999` /* maximum number of inputs and outputs */
- `#define REGNUM 9999` /* maximum number of registers */
- `#define TOKENNUM 9999` /* maximum number of tokens */
- `#define GATENUM 9999` /* maximum number of gates */
- `#define LINENUM 99999` /* maximum number of lines */
- `#define WIRENUM 99999` /* maximum number of wires */
- `#define GATESNUM 9`
- `#define oAND 1`
- `#define oOR 2`
- `#define oEXOR 3`

Functions

- `Biddy_String Biddy_Managed_Eval0 (Biddy_Manager MNG, Biddy_String s)`
Function `Biddy_Managed_Eval0` evaluates raw format.
- `Biddy_Edge Biddy_Managed_Eval1x (Biddy_Manager MNG, Biddy_String s, Biddy_LookupFunction lf)`
Function `Biddy_Managed_Eval1x` evaluates prefix AND-OR-EXOR-NOT format.
- `Biddy_Edge Biddy_Managed_Eval2 (Biddy_Manager MNG, Biddy_String boolFunc)`
Function `Biddy_Managed_Eval2` evaluates infix format.
- `Biddy_String Biddy_Managed_ReadBddview (Biddy_Manager MNG, const char filename[], Biddy_String name)`
Function `Biddy_Managed_ReadBddview` reads `bddview` file and creates a Boolean function.
- `void Biddy_Managed_ReadVerilogFile (Biddy_Manager MNG, const char filename[], Biddy_String prefix)`
Function `Biddy_Managed_ReadVerilogFile` reads Verilog file and creates variables for all primary inputs and Boolean functions for all primary outputs.

- void `Bidly_Managed_PrintfBDD` (`Bidly_Manager` MNG, `Bidly_Edge` f)
Function Bidly_Managed_PrintfBDD writes raw format using printf.
- void `Bidly_Managed_WriteBDD` (`Bidly_Manager` MNG, const char filename[], `Bidly_Edge` f, `Bidly_String` label)
Function Bidly_Managed_WriteBDD writes raw format using fprintf.
- void `Bidly_Managed_PrintfTable` (`Bidly_Manager` MNG, `Bidly_Edge` f)
Function Bidly_Managed_PrintfTable writes truth table using printf.
- void `Bidly_Managed_WriteTable` (`Bidly_Manager` MNG, const char filename[], `Bidly_Edge` f)
Function Bidly_Managed_WriteTable writes truth table using fprintf.
- void `Bidly_Managed_PrintfSOP` (`Bidly_Manager` MNG, `Bidly_Edge` f)
Function Bidly_Managed_PrintfSOP writes SOP using printf.
- void `Bidly_Managed_WriteSOP` (`Bidly_Manager` MNG, const char filename[], `Bidly_Edge` f)
Function Bidly_Managed_WriteSOP writes SOP using fprintf.
- void **`Bidly_Managed_PrintfMinterms`** (`Bidly_Manager` MNG, `Bidly_Edge` f, `Bidly_Boolean` negative)
- unsigned int `Bidly_Managed_WriteDot` (`Bidly_Manager` MNG, const char filename[], `Bidly_Edge` f, const char label[], int id, `Bidly_Boolean` cudd)
Function Bidly_Managed_WriteDot writes dot/graphviz format using fprintf.
- unsigned int `Bidly_Managed_WriteBddview` (`Bidly_Manager` MNG, const char filename[], `Bidly_Edge` f, const char label[], void *xytable)
Function Bidly_Managed_WriteBDDView writes bddview format using fprintf.

Variables

- const char * **`VerilogFileGateName`** []

3.2.1 Detailed Description

Description

PackageName [Bidly]

Synopsis [Bidly provides data structures and algorithms for the representation and manipulation of Boolean functions with ROBDDs, 0-sup-BDDs, and TZBDDs. A hash table is used for quick search of nodes. Complement edges decreases the number of nodes. An automatic garbage collection with a system age is implemented. Variable swapping and sifting are implemented.]

FileName [bidlyInOut.c]

Revision [\${Revision}: 545 \$]

Date [\${Date}: 2019-02-11 14:07:50 +0100 (pon, 11 feb 2019) \$]

Authors [Robert Meolic (robert@meolic.com),
Ales Casar (ales@homemade.net),
Volodymyr Mihav (mihav.wolodymyr@gmail.com),
David Kebo Houngninou (dhoungninou@smu.edu)]

Copyright

Copyright (C) 2006, 2019 UM FERi, Koroska cesta 46, SI-2000 Maribor, Slovenia

Bidly is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Bidly is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

More info

See also: [bidy.h](#), [bidyInt.h](#)

3.2.2 Function Documentation

3.2.2.1 `Biddy_Managed_Eval0()`

```
Biddy_String Biddy_Managed_Eval0 (
    Biddy_Manager MNG,
    Biddy_String s )
```

Description

First word is a name. It is followed by raw format. Function return name of the formula.

Side effects

All variables should already exists in the correct ordering! Not reentrant.

More info

Macro `Biddy_Eval0(s)` is defined for use with anonymous manager.

Definition at line 171 of file `bidyInOut.c`.

3.2.2.2 `Biddy_Managed_Eval1x()`

```
Biddy_Edge Biddy_Managed_Eval1x (
    Biddy_Manager MNG,
    Biddy_String s,
    Biddy_LookupFunction lf )
```

Description

Parameter `lf` is a lookup function in the user-defined cache table.

Side effects

Not reentrant.

More info

Macro `Biddy_Eval1x(s,lf)` is defined for use with anonymous manager. Macros `Biddy_Managed_Eval1(s)` and `Biddy_Eval1(s)` are defined for use without searching in the user-defined cache.

Definition at line 248 of file `bidyInOut.c`.

3.2.2.3 Biddy_Managed_Eval2()

```
Biddy_Edge Biddy_Managed_Eval2 (
    Biddy_Manager MNG,
    Biddy_String boolFunc )
```

Description

Parenthesis are implemented. Operators' priority is implemented. Formula Tree is supported (global table, only). Boolean constants are '0' and '1'. Boolean operators are NOT (~!), AND (&*), OR (|+), XOR (^%), XNOR (-), IMPLIES (><), NAND (@), NOR (#), BUTNOT (), NOTBUT (/).

Side effects

More info

Original author: Volodymyr Mihav (mihaw.wolodymyr@gmail.com) Original implementation of this function is on <https://github.com/sungmaster/liBDD>. Macro `Biddy_Eval2(boolFunc)` is defined for use with anonymous manager.

Definition at line 347 of file `biddyInOut.c`.

3.2.2.4 Biddy_Managed_PrintfBDD()

```
void Biddy_Managed_PrintfBDD (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

This function is intended for writing to an output channel via macro which overrides the meaning of standard printf calls. For writing raw format into the file, use `Biddy_Managed_WriteBDD`.

Side effects

More info

Macro `Biddy_PrintfBDD(f)` is defined for use with anonymous manager.

Definition at line 1080 of file `biddyInOut.c`.

3.2.2.5 Biddy_Managed_PrintfSOP()

```
void Biddy_Managed_PrintfSOP (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects****More info**

Definition at line 1350 of file biddyInOut.c.

3.2.2.6 Bidy_Managed_PrintfTable()

```
void Bidy_Managed_PrintfTable (
    Bidy_Manager MNG,
    Bidy_Edge f )
```

Description

This function is intended for writing to an output channel via macro which overrides the meaning of standard printf calls. For writing truth table into the file, use Bidy_Managed_WriteTable.

Side effects**More info**

Thanks to Jan Kraner and Ziga Kobale for prototype implementation. Macro Bidy_PrintfTable(f) is defined for use with anonymous manager.

Definition at line 1156 of file biddyInOut.c.

3.2.2.7 Bidy_Managed_ReadBddview()

```
Bidy_String Bidy_Managed_ReadBddview (
    Bidy_Manager MNG,
    const char filename[],
    Bidy_String name )
```

Description

If (name != NULL) then name will be used for the resulting BDD. If (name == NULL) then the resulting BDD will have name given in the file. Resulting BDD will not be preserved.

Side effects

Bidy_Managed_ConstructBDD is used and thus, if variable ordering in the file is not compatible with the active ordering then the result will be wrong! To improve efficiency, only the first "type", "var", and "label" are used.

More info

Macro [Bidly_ReadBddview\(filename\)](#) is defined for use with anonymous manager.

Definition at line 543 of file `biddyInOut.c`.

3.2.2.8 Bidly_Managed_ReadVerilogFile()

```
void Bidly_Managed_ReadVerilogFile (
    Bidly_Manager MNG,
    const char filename[],
    Bidly_String prefix )
```

Description

If (prefix != NULL) then the created BDD variables and formulae will get it.

Side effects

More info

Original author: David Kebo Houngrinou, Southern Methodist University Original implementation of this function is on <https://github.com/davidkebo/verilog-parser> Macro [Bidly_ReadVerilogFile\(filename,prefix\)](#) is defined for use with anonymous manager.

Definition at line 979 of file `biddyInOut.c`.

3.2.2.9 Bidly_Managed_WriteBDD()

```
void Bidly_Managed_WriteBDD (
    Bidly_Manager MNG,
    const char filename[],
    Bidly_Edge f,
    Bidly_String label )
```

Description

Side effects

More info

Macro [Bidly_WriteBDD\(f\)](#) is defined for use with anonymous manager.

Definition at line 1110 of file `biddyInOut.c`.

3.2.2.10 Biddy_Managed_WriteBddview()

```
unsigned int Biddy_Managed_WriteBddview (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f,
    const char label[],
    void * xytable )
```

Description

Output bddview format. Parameter table is optional, if not NULL then it must contain node names and coordinates. If (filename == NULL) then output is to stdout.

Side effects

A conservative approach to list all the existing variables in the manager (and not only the dependent ones) is used. Variable names containing # are adapted. To support EST, also variable names containing <> are adapted. The optional table is type-casted to BiddyXYTable.

More info

Macro [Biddy_WriteBddview\(filename,f,label\)](#) is defined for use with anonymous manager.

Definition at line 1916 of file biddyInOut.c.

3.2.2.11 Biddy_Managed_WriteDot()

```
unsigned int Biddy_Managed_WriteDot (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f,
    const char label[],
    int id,
    Biddy_Boolean cudd )
```

Description

Output dot format. Two approaches are implemented. The CUDD-like implementation is copied from CUDD 3.0.

Side effects

If (id != -1) then id is used instead of <...> for variable names. If (filename == NULL) then output is to stdout. Function resets all variables value. Variable names containing # are adapted.

More info

Macro [Biddy_WriteDot\(filename,f,label,id,cudd\)](#) is defined for use with anonymous manager.

Definition at line 1693 of file biddyInOut.c.

3.2.2.12 Biddy_Managed_WriteSOP()

```
void Biddy_Managed_WriteSOP (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f )
```

Description

Side effects

More info

Definition at line 1434 of file biddyInOut.c.

3.2.2.13 Biddy_Managed_WriteTable()

```
void Biddy_Managed_WriteTable (
    Biddy_Manager MNG,
    const char filename[],
    Biddy_Edge f )
```

Description

Side effects

More info

Thanks to Jan Kraner and Ziga Kobale for prototype implementation. Macro [Biddy_WriteTable\(f\)](#) is defined for use with anonymous manager.

Definition at line 1253 of file biddyInOut.c.

3.2.3 Variable Documentation

3.2.3.1 VerilogFileGateName

```
const char* VerilogFileGateName[]
```

Initial value:

```
= {
    "buf",
    "and",
    "nand",
    "or",
    "nor",
    "xor",
    "xnor",
    "not", "inv"
}
```

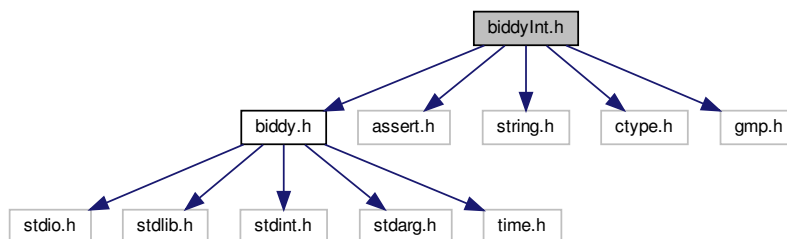
Definition at line 64 of file biddyInOut.c.

3.3 biddyInt.h File Reference

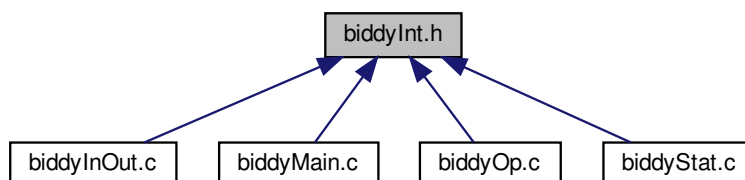
File [biddyInt.h](#) contains declaration of internal data structures.

```
#include "biddy.h"
#include <assert.h>
#include <string.h>
#include <ctype.h>
#include <gmp.h>
```

Include dependency graph for biddyInt.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define **BIDDYEXTENDEDSTATS_NO**
- #define **BIDDYEVENTLOG_NO**
- #define **ZF_LOGI(x)**
- #define **LEGACY_DOT**
- #define **UINTPTR** uintptr_t
- #define **UINTPTRSIZE** (8*sizeof(UINTPTR))
- #define **BIDDYVARMAX** 4096
- #define **LINESIZE** 999 /* maximum length of each input line read */
- #define **BUFSIZE** 99999 /* maximum length of a buffer */
- #define **INOUTNUM** 999 /* maximum number of inputs and outputs */
- #define **REGNUM** 9999 /* maximum number of registers */
- #define **TOKENNUM** 9999 /* maximum number of tokens */
- #define **GATENUM** 9999 /* maximum number of gates */

- #define **LINENUM** 99999 /* maximum number of lines */
- #define **WIREFNUM** 99999 /* maximum number of wires */
- #define **biddyNull** ((**Biddy_Edge**)NULL)
- #define **GET_ORDER**(orderingtable, X, Y) ((orderingtable)[X][Y/UINTPTRSIZE]&(((UINTPTR) 1)<<(Y%U←INTPTRSIZE)))!=0
- #define **SET_ORDER**(orderingtable, X, Y) (orderingtable)[X][Y/UINTPTRSIZE] |= (((UINTPTR) 1)<<(Y%U←INTPTRSIZE))
- #define **CLEAR_ORDER**(orderingtable, X, Y) (orderingtable)[X][Y/UINTPTRSIZE] &= (~(((UINTPTR) 1)<<(Y%UINTPTRSIZE)))
- #define **biddyManagerName** ((**Biddy_String**)(MNG[0]))
- #define **biddyManagerName1** ((**Biddy_String**)(MNG1[0]))
- #define **biddyManagerName2** ((**Biddy_String**)(MNG2[0]))
- #define **biddyManagerType** *((short int*)(MNG[1]))
- #define **biddyManagerType1** *((short int*)(MNG1[1]))
- #define **biddyManagerType2** *((short int*)(MNG2[1]))
- #define **biddyTerminal** ((void*)(MNG[2]))
- #define **biddyTerminal1** ((void*)(MNG1[2]))
- #define **biddyTerminal2** ((void*)(MNG2[2]))
- #define **biddyZero** ((**Biddy_Edge**)(MNG[3]))
- #define **biddyZero1** ((**Biddy_Edge**)(MNG1[3]))
- #define **biddyZero2** ((**Biddy_Edge**)(MNG2[3]))
- #define **biddyOne** ((**Biddy_Edge**)(MNG[4]))
- #define **biddyOne1** ((**Biddy_Edge**)(MNG1[4]))
- #define **biddyOne2** ((**Biddy_Edge**)(MNG2[4]))
- #define **biddyNodeTable** *((**BiddyNodeTable***)(MNG[5]))
- #define **biddyNodeTable1** *((**BiddyNodeTable***)(MNG1[5]))
- #define **biddyNodeTable2** *((**BiddyNodeTable***)(MNG2[5]))
- #define **biddyVariableTable** *((**BiddyVariableTable***)(MNG[6]))
- #define **biddyVariableTable1** *((**BiddyVariableTable***)(MNG1[6]))
- #define **biddyVariableTable2** *((**BiddyVariableTable***)(MNG2[6]))
- #define **biddyFormulaTable** *((**BiddyFormulaTable***)(MNG[7]))
- #define **biddyFormulaTable1** *((**BiddyFormulaTable***)(MNG1[7]))
- #define **biddyFormulaTable2** *((**BiddyFormulaTable***)(MNG2[7]))
- #define **biddyOPCache** *((**BiddyOp3CacheTable***)(MNG[8]))
- #define **biddyOPCache1** *((**BiddyOp3CacheTable***)(MNG1[8]))
- #define **biddyOPCache2** *((**BiddyOp3CacheTable***)(MNG2[8]))
- #define **biddyEACache** *((**BiddyOp3CacheTable***)(MNG[9]))
- #define **biddyEACache1** *((**BiddyOp3CacheTable***)(MNG1[9]))
- #define **biddyEACache2** *((**BiddyOp3CacheTable***)(MNG2[9]))
- #define **biddyRCCache** *((**BiddyOp3CacheTable***)(MNG[10]))
- #define **biddyRCCache1** *((**BiddyOp3CacheTable***)(MNG1[10]))
- #define **biddyRCCache2** *((**BiddyOp3CacheTable***)(MNG2[10]))
- #define **biddyReplaceCache** *((**BiddyKeywordCacheTable***)(MNG[11]))
- #define **biddyReplaceCache1** *((**BiddyKeywordCacheTable***)(MNG1[11]))
- #define **biddyReplaceCache2** *((**BiddyKeywordCacheTable***)(MNG2[11]))
- #define **biddyCacheList** *((**BiddyCacheList***)(MNG[12]))
- #define **biddyCacheList1** *((**BiddyCacheList***)(MNG1[12]))
- #define **biddyCacheList2** *((**BiddyCacheList***)(MNG2[12]))
- #define **biddyFreeNodes** *((**BiddyNode***)(MNG[13]))
- #define **biddyFreeNodes1** *((**BiddyNode***)(MNG1[13]))
- #define **biddyFreeNodes2** *((**BiddyNode***)(MNG2[13]))
- #define **biddyOrderingTable** *((**BiddyOrderingTable***)(MNG[14]))
- #define **biddyOrderingTable1** *((**BiddyOrderingTable***)(MNG1[14]))
- #define **biddyOrderingTable2** *((**BiddyOrderingTable***)(MNG2[14]))
- #define **biddySystemAge** *((unsigned int*)(MNG[15]))

- `#define bidlySystemAge1 *((unsigned int*)(MNG1[15]))`
- `#define bidlySystemAge2 *((unsigned int*)(MNG2[15]))`
- `#define bidlySelect *((unsigned short int*)(MNG[16]))`
- `#define bidlySelect1 *((unsigned short int*)(MNG1[16]))`
- `#define bidlySelect2 *((unsigned short int*)(MNG2[16]))`

Variables

- [Bidly_Manager](#) **bidlyAnonymousManager**
- BidlyLocalInfo * **bidlyLocalInfo**

3.3.1 Detailed Description

Description

```
PackageName [Bidly]
Synopsis [Bidly provides data structures and algorithms for the
         representation and manipulation of Boolean functions with
         ROBDDs, 0-sup-BDDs, and TZBDDs. A hash table is used for quick
         search of nodes. Complement edges decreases the number of
         nodes. An automatic garbage collection with a system age is
         implemented. Variable swapping and sifting are implemented.]

FileName [bidlyInt.h]
Revision [{$Revision: 545 $}]
Date [{$Date: 2019-02-11 14:07:50 +0100 (pon, 11 feb 2019) $}]
Authors [Robert Meolic (robert@meolic.com),
        Ales Casar (ales@homemade.net)]
```

Copyright

Copyright (C) 2006, 2019 UM FERi, Koroska cesta 46, SI-2000 Maribor, Slovenia

Bidly is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Bidly is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

More info

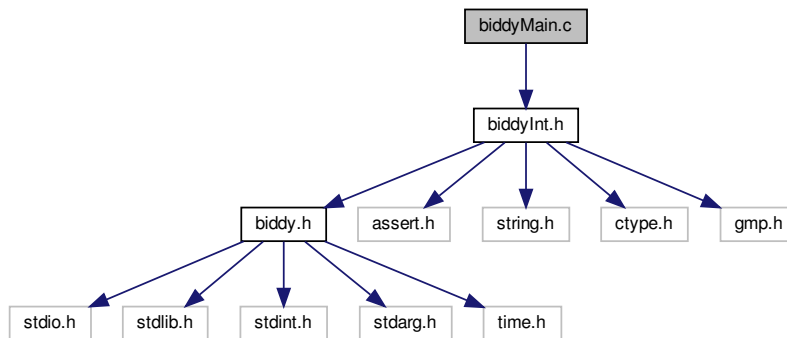
See also: [bidly.h](#)

3.4 biddyMain.c File Reference

File [biddyMain.c](#) contains main functions for representation and manipulation of boolean functions with various types of Binary Decision Diagrams.

```
#include "biddyInt.h"
```

Include dependency graph for [biddyMain.c](#):



Macros

- #define **TINY_SIZE** 1023
- #define **SMALL_SIZE** 65535
- #define **MEDIUM_SIZE** 262143
- #define **LARGE_SIZE** 1048575
- #define **XLARGE_SIZE** 2097151
- #define **XXLARGE_SIZE** 4194303
- #define **XXXLARGE_SIZE** 8388607
- #define **HUGE_SIZE** 16777215
- #define **SYSTEMREPORTVERBOSE**
- #define **SYSTEMREPORTDETAILS**
- #define **FUNCTIONREPORTVERBOSE**
- #define **FUNCTIONREPORTDETAILS**

Functions

- void [Bidly_InitMNG](#) ([Bidly_Manager](#) *mng, int bddtype)
Function Bidly_InitMNG initialize a manager.
- void [Bidly_ExitMNG](#) ([Bidly_Manager](#) *mng)
Function Bidly_ExitMNG deletes a manager.
- [Bidly_String](#) [Bidly_About](#) ()
Function Bidly_About reports version of Bidly package.
- int [Bidly_Managed_GetManagerType](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetManagerType reports BDD type used in the manager.
- [Bidly_String](#) [Bidly_Managed_GetManagerName](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetManagerName reports the name of the BDD type used in the manager.

- void [Bidly_Managed_SetManagerParameters](#) ([Bidly_Manager](#) MNG, float gcr, float gcrF, float gcrX, float rr, float rrF, float rrX, float st, float cst)
Function Bidly_Managed_SetManagerParameters set modifiable parameters.
- [Bidly_Edge Bidly_GetThen](#) ([Bidly_Edge](#) fun)
Function Bidly_GetThen returns THEN successor.
- [Bidly_Edge Bidly_GetElse](#) ([Bidly_Edge](#) fun)
Function Bidly_GetElse returns ELSE successor.
- [Bidly_Variable Bidly_GetTopVariable](#) ([Bidly_Edge](#) fun)
Function Bidly_GetTopVariable returns the top variable.
- [Bidly_Boolean Bidly_Managed_IsEqv](#) ([Bidly_Manager](#) MNG1, [Bidly_Edge](#) f1, [Bidly_Manager](#) MNG2, [Bidly_Edge](#) f2)
Function Bidly_Managed_IsEqv returns TRUE iff two BDDs are equal.
- void [Bidly_Managed_SelectNode](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_SelectNode selects the top node of the given function.
- void [Bidly_Managed_DeselectNode](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_DeselectNode deselects the top node of the given function.
- [Bidly_Boolean Bidly_Managed_IsSelected](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_IsSelected returns TRUE iff the top node of the given function is selected.
- void [Bidly_Managed_SelectFunction](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_SelectFunction recursively selects all nodes of a given function.
- void [Bidly_Managed_DeselectAll](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_DeselectAll deselects all nodes.
- [Bidly_Edge Bidly_Managed_GetTerminal](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetTerminal returns unmarked and untagged edge pointing to terminal node 1.
- [Bidly_Edge Bidly_Managed_GetConstantZero](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetConstantZero returns constant 0.
- [Bidly_Edge Bidly_Managed_GetConstantOne](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetConstantOne returns constant 1.
- [Bidly_Edge Bidly_Managed_GetBaseSet](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetBaseSet returns set containing only a null combination, i.e. it returns {{{}}.
- [Bidly_Variable Bidly_Managed_GetVariable](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x)
Function Bidly_Managed_GetVariable returns variable with the given name.
- [Bidly_Variable Bidly_Managed_GetLowestVariable](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_GetLowestVariable returns the lowest variable in the current ordering.
- [Bidly_Variable Bidly_Managed_GetIthVariable](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) i)
Function Bidly_Managed_GetIthVariable returns ith variable in the current global ordering.
- [Bidly_Variable Bidly_Managed_GetPrevVariable](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
Function Bidly_Managed_GetPrevVariable returns previous variable in the global ordering (lower, topmore).
- [Bidly_Variable Bidly_Managed_GetNextVariable](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
Function Bidly_Managed_GetNextVariable returns next variable in the global ordering (higher, bottommore).
- [Bidly_Edge Bidly_Managed_GetVariableEdge](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
Function Bidly_Managed_GetVariableEdge returns variable's edge.
- [Bidly_Edge Bidly_Managed_GetElementEdge](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
Function Bidly_Managed_GetElementEdge returns element's edge.
- [Bidly_String Bidly_Managed_GetVariableName](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
Function Bidly_Managed_GetVariableName returns the name of a variable.
- [Bidly_Edge Bidly_Managed_GetTopVariableEdge](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_GetTopVariableEdge returns variable's edge of top variable.
- [Bidly_String Bidly_Managed_GetTopVariableName](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_GetTopVariableName returns the name of top variable.
- char [Bidly_Managed_GetTopVariableChar](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)

- Function Bidly_Managed_GetTopVariableChar returns the first character in the name of top variable.*

 - void [Bidly_Managed_ResetVariablesValue](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_ResetVariablesValue sets all variable's value to biddyZero.*
 - void [Bidly_Managed_SetVariableValue](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v, [Bidly_Edge](#) f)
 - Function Bidly_Managed_SetVariableValue sets variable's value.*
 - [Bidly_Edge Bidly_Managed_GetVariableValue](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetVariableValue gets variable's value.*
 - void [Bidly_Managed_ClearVariablesData](#) ([Bidly_Manager](#) MNG)
 - Function Bidly_Managed_ClearVariablesData free memory used for all variable's data.*
 - void [Bidly_Managed_SetVariableData](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v, void *x)
 - Function Bidly_Managed_SetVariableData sets variable's data.*
 - void * [Bidly_Managed_GetVariableData](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_GetVariableData gets variable's data.*
 - [Bidly_Boolean Bidly_Managed_Eval](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_Eval returns the value of a Boolean function for a given variable assignment.*

Description

Side effects

Variables must have values assigned. Variable is considered to be FALSE iff variable.value == biddyZero, otherwise it is considered to be TRUE.

More info

- Macro Bidly_Eval(f) is defined for use with anonymous manager.*

 - double [Bidly_Managed_EvalProbability](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_EvalProbability evaluates BDD.*
 - [Bidly_Boolean Bidly_Managed_IsSmaller](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) fv, [Bidly_Variable](#) gv)
 - Function Bidly_Managed_IsSmaller returns TRUE if the first variable is smaller (= lower = previous = above = top-more).*
 - [Bidly_Boolean Bidly_Managed_IsLowest](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_IsLowest returns TRUE if the variable is the lowest one (lowest == topmost).*
 - [Bidly_Boolean Bidly_Managed_IsHighest](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_IsHighest returns TRUE if the variable is the highest one if terminal node is ignored (highest == bottommost).*
 - [Bidly_Variable Bidly_Managed_FoaVariable](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x, [Bidly_Boolean](#) varelem)
 - Function Bidly_Managed_FoaVariable finds variable/element or adds new variable (i.e. Boolean function f = x) and new element (i.e. it creates set {{x}}).*
 - void [Bidly_Managed_ChangeVariableName](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v, [Bidly_String](#) x)
 - Function Bidly_Managed_ChangeVariableName set new name to the given variable/element.*
 - [Bidly_Variable Bidly_Managed_AddVariableByName](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x)
 - Function Bidly_Managed_AddVariableByName adds variable.*
 - [Bidly_Variable Bidly_Managed_AddElementByName](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x)
 - Function Bidly_Managed_AddElementByName adds element.*
 - [Bidly_Edge Bidly_Managed_AddVariableBelow](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_AddVariableBelow adds variable.*
 - [Bidly_Edge Bidly_Managed_AddVariableAbove](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
 - Function Bidly_Managed_AddVariableAbove adds variable.*
 - [Bidly_Edge Bidly_Managed_TransferMark](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Boolean](#) mark, [Bidly_Boolean](#) leftright)
 - Function Bidly_Managed_TransferMark returns edge with inverted complement bit iff the second parameter is TRUE and normalization rules require this.*
 - [Bidly_Edge Bidly_Managed_IncTag](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
 - Function Bidly_Managed_IncTag returns edge with an incremented tag.*

- [Bidly_Edge Bidly_Managed_TaggedFoaNode](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v, [Bidly_Edge](#) pf, [Bidly_Edge](#) pt, [Bidly_Variable](#) ptag, [Bidly_Boolean](#) garbageAllowed)
Function Bidly_Managed_TaggedFoaNode finds or adds new node with the given variable and successors.
- [Bidly_Boolean Bidly_Managed_IsOK](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_IsOK returns TRUE iff given node is not obsolete.
- void [Bidly_Managed_GC](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) targetLT, [Bidly_Variable](#) targetGEQ, [Bidly_Boolean](#) purge, [Bidly_Boolean](#) total)
Function Bidly_Managed_GC performs garbage collection.
- void [Bidly_Managed_Clean](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_Clean performs cleaning.
- void [Bidly_Managed_Purge](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_Purge immediately removes all nodes which were not preserved or which are not preserved anymore.
- void [Bidly_Managed_PurgeAndReorder](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Boolean](#) converge)
Function Bidly_Managed_PurgeAndReorder immediately removes non-preserved nodes and triggers reordering on function.
- void [Bidly_Managed_Refresh](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_Refresh refreshes top node in a given function.
- void [Bidly_Managed_AddCache](#) ([Bidly_Manager](#) MNG, [Bidly_GCFunction](#) gc)
Function Bidly_Managed_AddCache adds cache to the end of Cache list.
- unsigned int [Bidly_Managed_AddFormula](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x, [Bidly_Edge](#) f, int c)
Function Bidly_Managed_AddFormula adds formula to Formula table.
- [Bidly_Boolean Bidly_Managed_FindFormula](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x, unsigned int *idx, [Bidly_Edge](#) *f)
Function Bidly_Managed_FindFormula find formula in Formula table.
- [Bidly_Boolean Bidly_Managed_DeleteFormula](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) x)
Function Bidly_Managed_DeleteFormula delete formula from Formula table.
- [Bidly_Boolean Bidly_Managed_DeletelthFormula](#) ([Bidly_Manager](#) MNG, unsigned int i)
Function Bidly_Managed_DeletelthFormula deletes formula from the table.
- [Bidly_Edge Bidly_Managed_GetlthFormula](#) ([Bidly_Manager](#) MNG, unsigned int i)
Function Bidly_Managed_GetlthFormula returns ith formula in a Formula table.
- [Bidly_String Bidly_Managed_GetlthFormulaName](#) ([Bidly_Manager](#) MNG, unsigned int i)
Function Bidly_Managed_GetlthFormulaName returns name of the ith formula in a Formula table.
- [Bidly_String Bidly_Managed_GetOrdering](#) ([Bidly_Manager](#) MNG)
- void [Bidly_Managed_SetOrdering](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) ordering)
Function Bidly_Managed_SetOrdering use variable swapping to create the ordering given by string.
- void [Bidly_Managed_SetAlphabeticOrdering](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_SetAlphabeticOrdering use variable swapping to create the alphabetic ordering.
- [Bidly_Variable Bidly_Managed_SwapWithHigher](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
Function Bidly_Managed_SwapWithHigher swaps two adjacent variables.
- [Bidly_Variable Bidly_Managed_SwapWithLower](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)
Function Bidly_Managed_SwapWithLower swaps two adjacent variables.
- [Bidly_Boolean Bidly_Managed_Sifting](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Boolean](#) converge)
Function Bidly_Managed_Sifting reorders variables to minimize node number using Rudell's sifting algorithm for the whole system (if f = NULL) or for the given function (if f != NULL) .
- void [Bidly_Managed_MinimizeBDD](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) name)
Function Bidly_Managed_MinimizeBDD reorders variables to minimize the node number of the given formula using an exhaustive search over all possible orderings.
- void [Bidly_Managed_MaximizeBDD](#) ([Bidly_Manager](#) MNG, [Bidly_String](#) name)
Function Bidly_Managed_MaximizeBDD reorders variables to maximize the node number of the given function using an exhaustive search over all possible orderings.
- [Bidly_Edge Bidly_Managed_Copy](#) ([Bidly_Manager](#) MNG1, [Bidly_Manager](#) MNG2, [Bidly_Edge](#) f)
Function Bidly_Managed_Copy copies a graph from one manager to another manager which can use the same or different BDD type.

Description

The function takes a graph from one manager and creates the same graph in another manager. If the managers do not use the same BDD type then a graph is converted. The resulting graph will represent the same Boolean function assuming the domain from the target manager. If ($f == \text{biddyZero}$) then only the complete domain (all variables) is copied.

Side effects

If source and target manager are the same then function does nothing. The variable ordering of the created BDD is trying to follow the original ordering, but if some variables already exist in the target manager then the final ordering is adapted to the target manager. Please note, that indices of variables in the target manager may not be the same as in the source manager (for example, if source manager does not use initial ordering the indices in the target manager will follow the variable's ordering and not variable's original indices)

More info

Macro `Bidly_Copy(MNG2,f)` is defined for use with anonymous manager.

- void `Bidly_Managed_CopyFormula` (`Bidly_Manager` MNG1, `Bidly_Manager` MNG2, `Bidly_String` x)

Function `Bidly_Managed_CopyFormula` uses `Bidly_Managed_Copy` to copy a graph from one manager to another manager which can use the same or different BDD type.

Description

See `Bidly_Managed_Copy`.

Side effects

If source and target manager are the same then function does nothing. The variable ordering of created BDD is adapted to the target manager. The created formula is refreshed but not preserved.

More info

Macro `Bidly_CopyFormula(MNG2,x)` is defined for use with anonymous manager.

- `Bidly_Edge Bidly_Managed_ConstructBDD` (`Bidly_Manager` MNG, int numV, `Bidly_String` varlist, int numN, `Bidly_String` nodelist)

Variables

- `Bidly_Manager` `bidlyAnonymousManager` = NULL
- `BidlyLocalInfo` * `bidlyLocalInfo` = NULL

3.4.1 Detailed Description**Description**

```

PackageName [Bidly]
Synopsis [Bidly provides data structures and algorithms for the
representation and manipulation of Boolean functions with
ROBDDs, 0-sup-BDDs, and TZBDDs. A hash table is used for quick
search of nodes. Complement edges decreases the number of
nodes. An automatic garbage collection with a system age is
implemented. Variable swapping and sifting are implemented.]

FileName [bidlyMain.c]
Revision [${Revision: 545 $}]
Date [${Date: 2019-02-11 14:07:50 +0100 (pon, 11 feb 2019) $}]
Authors [Robert Meolic (robert@meolic.com)]

```

Copyright

Copyright (C) 2006, 2019 UM FERl, Koroska cesta 46, SI-2000 Maribor, Slovenia

Biddy is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Biddy is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

More info

See also: [biddy.h](#), [biddyInt.h](#)

3.4.2 Function Documentation

3.4.2.1 Biddy_About()

```
Biddy_String Biddy_About ( )
```

Description

Side effects

More info

Definition at line 1127 of file biddyMain.c.

3.4.2.2 Biddy_ExitMNG()

```
void Biddy_ExitMNG (
    Biddy_Manager * mng )
```

Description

Deallocates all memory allocated by Biddy_InitMNG, Biddy_FoaVariable, Biddy_FoaNode etc.

Side effects

More info

Macro [Biddy_Exit\(\)](#) will delete anonymous manager.

Definition at line 904 of file biddyMain.c.

3.4.2.3 Biddy_GetElse()

```
Biddy_Edge Biddy_GetElse (
    Biddy_Edge fun )
```

Description

Input mark is not transferred! External use, only.

Side effects

For terminal nodes, function returns the same node.

More info

Macro BiddyE(fun) is defined for internal use.

Definition at line 1303 of file biddyMain.c.

3.4.2.4 Biddy_GetThen()

```
Biddy_Edge Biddy_GetThen (
    Biddy_Edge fun )
```

Description

Input mark is not transferred! External use, only.

Side effects

For terminal nodes, function returns the same node.

More info

Macro BiddyT(fun) is defined for internal use.

Definition at line 1264 of file biddyMain.c.

3.4.2.5 Biddy_GetTopVariable()

```
Biddy_Variable Biddy_GetTopVariable (
    Biddy_Edge fun )
```

Description

External use, only.

Side effects**More info**

Macro `BiddyV(fun)` is defined for internal use.

Definition at line 1341 of file `bidlyMain.c`.

3.4.2.6 Bidly_InitMNG()

```
void Bidly_InitMNG (
    Bidly_Manager * mng,
    int bddtype )
```

Description

`Bidly_InitMNG` creates and initializes a manager. Initialization consists of creating manager structure (MNG), node table (`bidlyNodeTable`), variable table (`bidlyVariableTable`), formula table (`bidlyFormulaTable`), four basic caches (`bidlyOPCache`, `bidlyEACache`, `bidlyRCCache`, and `bidlyReplaceCache`), and cache list (`bidlyCacheList`). `Bidly_InitMNG` also initializes constant edges (`bidlyOne`, `bidlyZero`), memory management and automatic garbage collection.

Side effects

Allocates a lot of memory.

More info

Macro `Bidly_InitAnonymous()` will initialize anonymous manager. Macro `Bidly_Init()` will initialize anonymous manager for ROBDDs.

Definition at line 179 of file `bidlyMain.c`.

3.4.2.7 Bidly_Managed_AddCache()

```
void Bidly_Managed_AddCache (
    Bidly_Manager MNG,
    Bidly_GCFunction gc )
```

Description

If Cache list does not exist, function creates it.

Side effects**More info**

Macro `Bidly_AddCache(gc)` is defined for use with anonymous manager.

Definition at line 4437 of file `bidlyMain.c`.

3.4.2.8 Biddy_Managed_AddElementByName()

```
Biddy_Variable Biddy_Managed_AddElementByName (
    Biddy_Manager MNG,
    Biddy_String x )
```

Description

Biddy_Managed_AddElementByName uses Biddy_Managed_FoaVariable to find or add element. Function returns element edge. If element already exists, function returns the existing element edge. For more details see Biddy_Managed_FoaVariable.

Side effects

See Biddy_Managed_FoaVariable.

More info

Macro Biddy_AddElementByName(x) is defined for use with anonymous manager. Macros Biddy_Managed_AddElement(MNG) and Biddy_AddElement() are defined for creating numbered elements.

Definition at line 2928 of file biddyMain.c.

3.4.2.9 Biddy_Managed_AddFormula()

```
unsigned int Biddy_Managed_AddFormula (
    Biddy_Manager MNG,
    Biddy_String x,
    Biddy_Edge f,
    int c )
```

Description

Given BDD becomes a formula. If (x != NULL) then formula is accessible by its name. If (x != NULL) then index of the formula in the Formulae Table is returned, otherwise function returns 0. Nodes of the given BDD will be preserved for the given number of clearings. If (c == -1) then formula is refreshed but not preserved. If (c == 0) then formula is persistently preserved. You have to use Biddy_DeleteFormula and Biddy_Purge to remove nodes of persistently preserved formulae. There are five macros defined to simplify formulae management: Biddy_Managed_AddTmpFormula(mng,name,bdd) := Biddy_Managed_AddFormula(mng,name,bdd,-1) Biddy_Managed_AddPersistentFormula(mng,name,bdd) := Biddy_Managed_AddFormula(mng,name,bdd,0) Biddy_Managed_KeepFormula(mng,bdd) := Biddy_Managed_AddFormula(mng,NULL,bdd,1) Biddy_Managed_KeepFormulaProlonged(mng,bdd,c) := Biddy_Managed_AddFormula(mng,NULL,bdd,c) Biddy_Managed_KeepFormulaUntilPurge(mng,bdd) := Biddy_Managed_AddFormula(mng,NULL,bdd,0)

Side effects

Function is prolonged or fortified. Formulae with name are ordered by name. If formula with the same name already exists, it will be overwritten - preserved (i.e. not obsolete and not fresh) and persistently preserved formulae will be deleted at the original index and recreated at new index!

More info

Macros [Biddy_AddFormula\(x,f,c\)](#), [Biddy_AddTmpFormula\(x,f\)](#), [Biddy_AddPersistentFormula\(x,f\)](#), [Biddy_KeepFormula\(f\)](#), [Biddy_KeepFormulaProlonged\(f,c\)](#), and [Biddy_KeepFormulaUntilPurge\(f\)](#) are defined for use with anonymous manager.

Definition at line 4511 of file biddyMain.c.

3.4.2.10 Biddy_Managed_AddVariableAbove()

```
Biddy_Edge Biddy_Managed_AddVariableAbove (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

[Biddy_Managed_AddVariableAbove](#) uses [Biddy_Managed_AddVariableByName](#) to add new variable. Then, the order of the new variable is changed to become immediately above the given variable (above = previous = topmore in BDD) Function returns variable edge.

Side effects

More info

Macro [Biddy_AddVariableAbove\(v\)](#) is defined for use with anonymous manager.

Definition at line 3040 of file biddyMain.c.

3.4.2.11 Biddy_Managed_AddVariableBelow()

```
Biddy_Edge Biddy_Managed_AddVariableBelow (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

[Biddy_Managed_AddVariableBelow](#) uses [Biddy_Managed_AddVariableByName](#) to add new variable. Then, the order of the new variable is changed to become immediately below the given variable (below = next = bottommore in BDD) Function returns variable edge.

Side effects

More info

Macro [Biddy_AddVariableBelow\(v\)](#) is defined for use with anonymous manager.

Definition at line 2957 of file biddyMain.c.

3.4.2.12 Biddy_Managed_AddVariableByName()

```
Biddy_Variable Biddy_Managed_AddVariableByName (
    Biddy_Manager MNG,
    Biddy_String x )
```

Description

Biddy_Managed_AddVariableByName uses Biddy_Managed_FoaVariable to find or add variable. Function returns variable edge. If variable already exists, function returns the existing variable edge. For more details see Biddy_Managed_FoaVariable.

Side effects

See Biddy_Managed_FoaVariable.

More info

Macro [Biddy_AddVariableByName\(x\)](#) is defined for use with anonymous manager. Macros [Biddy_Managed_AddVariable\(MNG\)](#) and [Biddy_AddVariable\(\)](#) are defined for creating numbered variables.

Definition at line 2896 of file biddyMain.c.

3.4.2.13 Biddy_Managed_ChangeVariableName()

```
void Biddy_Managed_ChangeVariableName (
    Biddy_Manager MNG,
    Biddy_Variable v,
    Biddy_String x )
```

Description

Side effects

It is not checked that the same variable/element does not already exist. The ordering of the variable is not changed.

More info

Macro [Biddy_ChangeVariableName\(v,x\)](#) is defined for use with anonymous manager.

Definition at line 2860 of file biddyMain.c.

3.4.2.14 Biddy_Managed_Clean()

```
void Biddy_Managed_Clean (
    Biddy_Manager MNG )
```

Description

Discard all nodes which are not preserved or which are not preserved anymore. Obsolete nodes are not immediately removed, they will be removed during the first garbage collection.

Side effects

Field deleted is not considered and thus no fortified node and no prolonged node is discarded.

More info

Macro [Bidly_Clean\(\)](#) is defined for use with anonymous manager.

Definition at line 4283 of file `bidlyMain.c`.

3.4.2.15 Bidly_Managed_ClearVariablesData()

```
void Bidly_Managed_ClearVariablesData (
    Bidly_Manager MNG )
```

Description**Side effects**

Only active (used) variables are considered.

More info

Macro [Bidly_ClearVariablesData\(\)](#) is defined for use with anonymous manager.

Definition at line 2168 of file `bidlyMain.c`.

3.4.2.16 Bidly_Managed_ConstructBDD()

```
Bidly_Edge Bidly_Managed_ConstructBDD (
    Bidly_Manager MNG,
    int numV,
    Bidly_String varlist,
    int numN,
    Bidly_String nodelist )
```

Function***** Synopsis [Function `Bidly_Managed_ConstructBDD`.] Description [`Bidly_Managed_ConstructBDD` constructs BDD from lists of nodes and edges. In both lists, elements are separated with spaces. Elements of node list has the following format: terminal node zero = (type=0,l=-1,r=-1), terminal node one = (type=1,l=-1,r=-1), regular label = (type=2,r=-1), complemented label = (type=3,r=-1), regular node = (type=4), node with complemented successor(s) = (type=5) The first element in nodelist is ignored. An example node list is: 'Bidly 0 Bidly 2 1 -1 1 B 4 2 3 2 0 0 -1 -1 3 i 4 4 9 4 d 4 5 6 5 0 0 -1 -1 6 y 4 7 8 7 0 0 -1 -1 8 1 1 -1 -1 9 d 4 6 10 10 1 1 -1 -1' which is constructed as: '{Bidly} {0 Bidly 2 1 -1} {1 B 4 2 3} {2 0 0 -1 -1} {3 i 4 4 9} {4 d 4 5 6} {5 0 0 -1 -1} {6 y 4 7 8} {7 0 0 -1 -1} {8 1 1 -1 -1} {9 d 4 6 10} {10 1 1 -1 -1}'.] SideEffects [If variable ordering in the file is not compatible with the active ordering then the result will be wrong.] SeeAlso []

Definition at line 6203 of file `bidlyMain.c`.

3.4.2.17 Bidly_Managed_DeleteFormula()

```
Bidly_Boolean Bidly_Managed_DeleteFormula (
    Bidly_Manager MNG,
    Bidly_String x )
```

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. Formulae representing constants and variables will not be deleted.

More info

Macro [Bidly_DeleteFormula\(x\)](#) is defined for use with anonymous manager.

Definition at line 4964 of file bidlyMain.c.

3.4.2.18 Bidly_Managed_DeleteIthFormula()

```
Bidly_Boolean Bidly_Managed_DeleteIthFormula (
    Bidly_Manager MNG,
    unsigned int i )
```

Description

Formula is labelled but not immediately removed. Nodes of the given formula are not immediately removed.

Side effects

Formula is not accessible by its name anymore. The first two formulae ("0" and "1") will not be deleted. Formulae representing variables will not be deleted.

More info

Macro [Bidly_DeleteIthFormula\(x\)](#) is defined for use with anonymous manager.

Definition at line 5027 of file bidlyMain.c.

3.4.2.19 Bidly_Managed_DeselectAll()

```
void Bidly_Managed_DeselectAll (
    Bidly_Manager MNG )
```

Description

Deselect all nodes.

Side effects**More info**

Macro [Biddy_DeselectAll\(\)](#) is defined for use with anonymous manager.

Definition at line 1536 of file biddyMain.c.

3.4.2.20 Biddy_Managed_DeselectNode()

```
void Biddy_Managed_DeselectNode (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects****More info**

Macro [Biddy_DeselectNode\(f\)](#) is defined for use with anonymous manager.

Definition at line 1426 of file biddyMain.c.

3.4.2.21 Biddy_Managed_EvalProbability()

```
double Biddy_Managed_EvalProbability (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Each variable should be associated with a probability q (0-1) via data element in BiddyVariable. For each 1-path a product of variable's probability is calculated (q for 'then' successor and $(1-q)$ for 'else' successor). The result of the function is the sum of all such products.

Side effects

Implemented for OBDD, ZBDD, and TZBDD.

More info

Macro [Bidly_EvalProbability\(f\)](#) is defined for use with anonymous manager.

Definition at line 2388 of file `bidlyMain.c`.

3.4.2.22 Bidly_Managed_FindFormula()

```
Bidly_Boolean Bidly_Managed_FindFormula (
    Bidly_Manager MNG,
    Bidly_String x,
    unsigned int * idx,
    Bidly_Edge * f )
```

Description

Return TRUE/FALSE, index, and the formula. If formula is constant or variable then `idx = 0` and `f != bidlyNull`. If formula is not found then `idx` is a position where the formulae should exist and `f == bidlyNull`.

Side effects**More info**

Macro [Bidly_FindFormula\(x,f\)](#) is defined for use with anonymous manager.

Definition at line 4790 of file `bidlyMain.c`.

3.4.2.23 Bidly_Managed_FoaVariable()

```
Bidly_Variable Bidly_Managed_FoaVariable (
    Bidly_Manager MNG,
    Bidly_String x,
    Bidly_Boolean varelem )
```

Description

If variable/element already exists, function returns the existing one. If `x == NULL` then numbered variable/element is added. Numbered variables/elements have only digits in its name. The current number of numbered variables/elements is stored in `numnum`. If numbered variable/element is requested then function increments `numnum` and creates a new (non-existing) variable/element. Parameter `varelem` is used to determine how to adapt the existing BDD base to keep the current formula valid (use `varelem = TRUE` if formulae represent Boolean functions and `varelem = FALSE` if they represent combination sets). The ordering of the new variable/element is determined in `Bidly_InitMNG`. Function always returns variable.

Side effects

Adding new variable/element may change the meaning of the existing BDDs. Variables and elements are always repaired. Formulae are repaired with regards to the parameter `varelem`. BDDs without external references are not repaired. For OBDDs, OFDDs, TZBDDs, and TZFDDs, it is safe to add new variables/elements if BDDs are used to represent Boolean functions. For ZBDDs and ZFDDs, it is safe to add new variables/elements if BDDs are used to represent combination sets. User should not add numbered variables/elements with some other function. TO DO: Formulae in user's formula tables are not repaired, yet! TO DO: Variables cannot be deleted, yet!

More info

Macro `Bidly_FoaVariable(x)` is defined for use with anonymous manager.

Definition at line 2573 of file `bidlyMain.c`.

3.4.2.24 Bidly_Managed_GC()

```
void Bidly_Managed_GC (
    Bidly_Manager MNG,
    Bidly_Variable targetLT,
    Bidly_Variable targetGEQ,
    Bidly_Boolean purge,
    Bidly_Boolean total )
```

Description

All obsolete nodes are deleted. Parameter `purge` should not be true during automatic garbage collection. If parameter `purge` is true then all formulae without name are deleted. If parameter `purge` is true then all nodes which are not part of non-obsolete non-deleted formulae are removed even if they are fresh or fortified. If parameter `total` is true than all unnecessary nodes are immediately deleted, otherwise they are deleted only when there are enough of them. If (`targetLT != 0`) then node table resizing is disabled. If (`targetLT != 0`) then there should not exist obsolete formulae. If (`targetLT != 0`) then there should not exist obsolete nodes which are not part of any non-obsolete non-deleted formulae. If (`targetLT != 0`) then there should not exist obsolete nodes with variable equal or higher (bottom-more) than `target` and smaller (top-more) than `targetGEQ`.

Side effects

The first element of each chain in a node table should have a special value for its 'prev' element to allow tricky but efficient deleting. Moreover, 'prev' and 'next' should be the first and the second element in the structure `Bidly_Node`, respectively. Garbage collection is reported by `bidlyNodeTable.garbage` only if some bad nodes are purged! Parameters `targetLT` and `targetGEQ` are used during sifting, only, in all other cases 0 is used.

More info

Macro `Bidly_GC(targetLT,targetGEQ,purge,total)` is defined for use with anonymous manager. Macros `Bidly_Managed_AutoGC(MNG)` and `Bidly_AutoGC()` are useful variants with `targetLT = targetGEQ = 0`, `purge = FALSE`, and `total = FALSE`. Macros `Bidly_Managed_ForceGC(MNG)` and `Bidly_ForceGC()` are useful variants with `targetLT = targetGEQ = 0`, `purge = FALSE`, and `total = TRUE`.

Definition at line 3788 of file `bidlyMain.c`.

3.4.2.25 Bidly_Managed_GetBaseSet()

```
Bidly_Edge Bidly_Managed_GetBaseSet (
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro `Biddy_GetBaseSet()` is defined for use with anonymous manager.

Definition at line 1668 of file `biddyMain.c`.

3.4.2.26 Biddy_Managed_GetConstantOne()

```
Biddy_Edge Biddy_Managed_GetConstantOne (
    Biddy_Manager MNG )
```

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 1 coincides with universal set.

Side effects

For ZBDDs and ZFDDs, you should always obtain constant 1 through the call of this function!

More info

Internally, use macro `biddyOne` (also for ZBDDs and ZFDDs!). Macro `Biddy_GetConstantOne()` is defined for use with anonymous manager. Macros `Biddy_Managed_GetUniversalSet(MNG)` and `Biddy_GetUniversalSet()` are defined for manipulation of combination sets.

Definition at line 1642 of file `biddyMain.c`.

3.4.2.27 Biddy_Managed_GetConstantZero()

```
Biddy_Edge Biddy_Managed_GetConstantZero (
    Biddy_Manager MNG )
```

Description

Constants 0 and 1 depend on a manager. For combination sets, constant 0 coincides with empty set.

Side effects**More info**

Internally, use macro `biddyZero`. Macro `Biddy_GetConstantZero()` is defined for use with anonymous manager. Macros `Biddy_Managed_GetEmptySet(MNG)` and `Biddy_GetEmptySet()` are defined for manipulation of combination sets.

Definition at line 1610 of file `biddyMain.c`.

3.4.2.28 Bidly_Managed_GetElementEdge()

```
Bidly_Edge Bidly_Managed_GetElementEdge (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Side effects

More info

Macro [Bidly_GetElementEdge\(v\)](#) is defined for use with anonymous manager.

Definition at line 1947 of file `bidlyMain.c`.

3.4.2.29 Bidly_Managed_GetIthFormula()

```
Bidly_Edge Bidly_Managed_GetIthFormula (
    Bidly_Manager MNG,
    unsigned int i )
```

Description

Return `bidlyNull` if `ith` formulae does not exist.

Side effects

After adding new formula the index of others may change!

More info

Macro [Bidly_GetIthFormula\(i\)](#) is defined for use with anonymous manager.

Definition at line 5087 of file `bidlyMain.c`.

3.4.2.30 Bidly_Managed_GetIthFormulaName()

```
Bidly_String Bidly_Managed_GetIthFormulaName (
    Bidly_Manager MNG,
    unsigned int i )
```

Description

Return `NULL` if `ith` formulae does not exist.

Side effects

After adding new formula the index of others may change!

More info

Macro [Bidly_GetlthFormulaName\(i\)](#) is defined for use with anonymous manager.

Definition at line 5118 of file bidlyMain.c.

3.4.2.31 Bidly_Managed_GetlthVariable()

```
Bidly_Variable Bidly_Managed_GetlthVariable (
    Bidly_Manager MNG,
    Bidly_Variable i )
```

Description

The lowest (topmost) variable has global ordering 1. The highest (bottommost) variable is '1' and has global ordering equal to numUsedVariables.

Side effects

If argument is 0, function returns 0. If argument is larger than the number of variables, function returns 0.

More info

Macro [Bidly_GetlthVariable\(x\)](#) is defined for use with anonymous manager.

Definition at line 1825 of file bidlyMain.c.

3.4.2.32 Bidly_Managed_GetLowestVariable()

```
Bidly_Variable Bidly_Managed_GetLowestVariable (
    Bidly_Manager MNG )
```

Description

The lowest variable is the topmost variable.

Side effects**More info**

Macro [Bidly_GetLowestVariable\(x\)](#) is defined for use with anonymous manager.

Definition at line 1787 of file bidlyMain.c.

3.4.2.33 **Bidly_Managed_GetManagerName()**

```
Bidly_String Bidly_Managed_GetManagerName (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_GetManagerName\(\)](#) is defined for use with anonymous manager.

Definition at line 1176 of file `bidlyMain.c`.

3.4.2.34 **Bidly_Managed_GetManagerType()**

```
int Bidly_Managed_GetManagerType (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_GetManagerType\(\)](#) is defined for use with anonymous manager.

Definition at line 1150 of file `bidlyMain.c`.

3.4.2.35 **Bidly_Managed_GetNextVariable()**

```
Bidly_Variable Bidly_Managed_GetNextVariable (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Side effects

More info

Macro [Bidly_GetNextVariable\(v\)](#) is defined for use with anonymous manager.

Definition at line 1893 of file `bidlyMain.c`.

3.4.2.36 Biddy_Managed_GetPrevVariable()

```
Biddy_Variable Biddy_Managed_GetPrevVariable (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

Side effects

More info

Macro [Biddy_GetPrevVariable\(v\)](#) is defined for use with anonymous manager.

Definition at line 1863 of file biddyMain.c.

3.4.2.37 Biddy_Managed_GetTerminal()

```
Biddy_Edge Biddy_Managed_GetTerminal (
    Biddy_Manager MNG )
```

Description

Terminal node depends on a manager.

Side effects

More info

Internally, use macro `biddyTerminal`. Macro [Biddy_GetTerminal\(\)](#) is defined for use with anonymous manager.

Definition at line 1580 of file biddyMain.c.

3.4.2.38 Biddy_Managed_GetTopVariableChar()

```
char Biddy_Managed_GetTopVariableChar (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Side effects

More info

Macro [Biddy_GetTopVariableChar\(f\)](#) is defined for use with anonymous manager.

Definition at line 2056 of file biddyMain.c.

3.4.2.39 `Bidly_Managed_GetTopVariableEdge()`

```
Bidly_Edge Bidly_Managed_GetTopVariableEdge (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Side effects

TO DO: For ZBDDs, element edge is sometimes preferred over variable edge.

More info

Macro `Bidly_GetTopVariableEdge(f)` is defined for use with anonymous manager.

Definition at line 2000 of file `bidlyMain.c`.

3.4.2.40 `Bidly_Managed_GetTopVariableName()`

```
Bidly_String Bidly_Managed_GetTopVariableName (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Side effects

More info

Macro `Bidly_GetTopVariableName(f)` is defined for use with anonymous manager.

Definition at line 2028 of file `bidlyMain.c`.

3.4.2.41 `Bidly_Managed_GetVariable()`

```
Bidly_Variable Bidly_Managed_GetVariable (
    Bidly_Manager MNG,
    Bidly_String x )
```

Description

Side effects

If variable is not found function returns 0!

More info

Macro [Bidly_GetVariable\(x\)](#) is defined for use with anonymous manager.

Definition at line 1733 of file `bidlyMain.c`.

3.4.2.42 Bidly_Managed_GetVariableData()

```
void* Bidly_Managed_GetVariableData (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Side effects

It is not checked that the given variable is valid.

More info

Macro [Bidly_GetVariableData\(v\)](#) is defined for use with anonymous manager.

Definition at line 2227 of file `bidlyMain.c`.

3.4.2.43 Bidly_Managed_GetVariableEdge()

```
Bidly_Edge Bidly_Managed_GetVariableEdge (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Side effects

More info

Macro [Bidly_GetVariableEdge\(v\)](#) is defined for use with anonymous manager.

Definition at line 1922 of file `bidlyMain.c`.

3.4.2.44 Bidly_Managed_GetVariableName()

```
Bidly_String Bidly_Managed_GetVariableName (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description**Side effects****More info**

Macro `Bidly_GetVariableName(v)` is defined for use with anonymous manager.

Definition at line 1972 of file `bidlyMain.c`.

3.4.2.45 Bidly_Managed_GetVariableValue()

```
Bidly_Edge Bidly_Managed_GetVariableValue (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description**Side effects**

It is not checked that the given variable is valid.

More info

Macro `Bidly_GetVariableValue(v)` is defined for use with anonymous manager.

Definition at line 2141 of file `bidlyMain.c`.

3.4.2.46 Bidly_Managed_IncTag()

```
Bidly_Edge Bidly_Managed_IncTag (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Used for TZBDDs and TZFDDs, only.

Side effects

It is not checked, that the resulting tag is not greater than top variable. Function may return non-fresh node even if `f` is fresh.

More info

Macro `Bidly_IncTag()` is defined for use with anonymous manager.

Definition at line 3187 of file `bidlyMain.c`.

3.4.2.47 Bidly_Managed_IsEqv()

```
Bidly_Boolean Bidly_Managed_IsEqv (
    Bidly_Manager MNG1,
    Bidly_Edge f1,
    Bidly_Manager MNG2,
    Bidly_Edge f2 )
```

Description

It is assumed that f1 and f2 have the same ordering.

Side effects

More info

Macro [Bidly_IsEqv\(f1,MNG2,f2\)](#) is defined for use with anonymous manager.

Definition at line 1365 of file bidlyMain.c.

3.4.2.48 Bidly_Managed_IsHighest()

```
Bidly_Boolean Bidly_Managed_IsHighest (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Side effects

More info

Macro [Bidly_IsHighest\(v\)](#) is defined for internal use. Macro [Bidly_IsHighest\(v\)](#) is defined for use with anonymous manager.

Definition at line 2515 of file bidlyMain.c.

3.4.2.49 Bidly_Managed_IsLowest()

```
Bidly_Boolean Bidly_Managed_IsLowest (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description**Side effects****More info**

Macro `BidlyIsLowest(v)` is defined for internal use. Macro `Bidly_IsLowest(v)` is defined for use with anonymous manager.

Definition at line 2478 of file `bidlyMain.c`.

3.4.2.50 Bidly_Managed_IsOK()

```
Bidly_Boolean Bidly_Managed_IsOK (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

This is needed for implementation of user caches.

Side effects**More info**

Macro `BidlyIsOK(f)` is defined for debugging. It will check more properties and not only the expiry value. Macro `Bidly_IsOK(f)` is defined for use with anonymous manager.

Definition at line 3736 of file `bidlyMain.c`.

3.4.2.51 Bidly_Managed_IsSelected()

```
Bidly_Boolean Bidly_Managed_IsSelected (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description**Side effects****More info**

Macro `Bidly_IsSelected(f)` is defined for use with anonymous manager.

Definition at line 1452 of file `bidlyMain.c`.

3.4.2.52 Biddy_Managed_IsSmaller()

```
Biddy_Boolean Biddy_Managed_IsSmaller (
    Biddy_Manager MNG,
    Biddy_Variable fv,
    Biddy_Variable gv )
```

Description

Side effects

More info

Macro BiddyIsSmaller(fv,gv) is defined for internal use. Macro [Biddy_IsSmaller\(fv,gv\)](#) is defined for use with anonymous manager.

Definition at line 2451 of file biddyMain.c.

3.4.2.53 Biddy_Managed_MaximizeBDD()

```
void Biddy_Managed_MaximizeBDD (
    Biddy_Manager MNG,
    Biddy_String name )
```

Description

Steinhaus–Johnson–Trotter algorithm is used to generate all possible permutations. An optimized version of Bubble Sort is used to setup the final ordering. Variables are reordered globally. All obsolete nodes will be removed.

Side effects

Indeed, this function may take a lot of time! For TZBDD, all unreferenced nodes (not part of registered formulae) will be removed. For TZBDD, this function may change top edge or even a top node of any function/formula - this is a problem, because functions referenced by local variables only may become wrong. Consequently, for TZBDDs, sifting is not safe to start automatically!

More info

Macro [Biddy_MaximizeBDD\(f\)](#) is defined for use with anonymous manager.

Definition at line 5719 of file biddyMain.c.

3.4.2.54 Biddy_Managed_MinimizeBDD()

```
void Biddy_Managed_MinimizeBDD (
    Biddy_Manager MNG,
    Biddy_String name )
```

Description

Steinhaus–Johnson–Trotter algorithm is used to generate all possible permutations. An optimized version of Bubble Sort is used to setup the final ordering. Variables are reordered globally. All obsolete nodes will be removed.

Side effects

Indeed, this function may take a lot of time! For TZBDD, all unreferenced nodes (not part of registered formulae) will be removed. For TZBDD, this function may change top edge or even a top node of any function/formula - this is a problem, because functions referenced by local variables only may become wrong. Consequently, for TZBDDs, sifting is not safe to start automatically!

More info

Macro [Biddy_MinimizeBDD\(f\)](#) is defined for use with anonymous manager.

Definition at line 5620 of file `biddyMain.c`.

3.4.2.55 Biddy_Managed_Purge()

```
void Biddy_Managed_Purge (
    Biddy_Manager MNG )
```

Description

All formulae without name are deleted. All deleted formulae (including prolonged/fortified formulae) are removed. All fresh and obsolete nodes are immediately removed. Moreover, all prolonged and fortified nodes are immediately removed if they are not needed by some of the remaining formula. Call to `Biddy_Purge` does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

Removes all fresh nodes!

More info

Macro [Biddy_Purge\(f\)](#) is defined for use with anonymous manager.

Definition at line 4334 of file `biddyMain.c`.

3.4.2.56 Biddy_Managed_PurgeAndReorder()

```
void Biddy_Managed_PurgeAndReorder (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Boolean converge )
```


Description

All obsolete nodes are immediately removed. Moreover, nodes from deleted prolonged formulae and nodes from deleted fortified formulae are removed if they are not needed by other formulae. If BDD is given ($f \neq \text{NULL}$), reordering on function is used. Otherwise ($f == \text{NULL}$) global reordering is used. Call to `Biddy_PurgeAndReorder` does not count as clearing and thus all preserved formulae remains preserved for the same number of clearings.

Side effects

Removes all fresh nodes.

More info

Macro `Biddy_PurgeAndReorder(f)` is defined for use with anonymous manager.

Definition at line 4382 of file `biddyMain.c`.

3.4.2.57 Biddy_Managed_Refresh()

```
void Biddy_Managed_Refresh (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

This is an external variant of internal macro `BiddyRefresh`. This is needed for implementing user caches.

Side effects**More info**

Macro `Biddy_Refresh(f)` is defined for use with anonymous manager.

Definition at line 4411 of file `biddyMain.c`.

3.4.2.58 Biddy_Managed_ResetVariablesValue()

```
void Biddy_Managed_ResetVariablesValue (
    Biddy_Manager MNG )
```

Description**Side effects**

Only active (used) variables are reinitialized.

More info

Macro [Biddy_ResetVariablesValue\(\)](#) is defined for use with anonymous manager.

Definition at line 2085 of file biddyMain.c.

3.4.2.59 Biddy_Managed_SelectFunction()

```
void Biddy_Managed_SelectFunction (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects**

Terminal node must be selected before starting this function!

More info

Macro [Biddy_SelectFunction\(f\)](#) is defined for use with anonymous manager.

Definition at line 1479 of file biddyMain.c.

3.4.2.60 Biddy_Managed_SelectNode()

```
void Biddy_Managed_SelectNode (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects****More info**

Macro [Biddy_SelectNode\(f\)](#) is defined for use with anonymous manager.

Definition at line 1400 of file biddyMain.c.

3.4.2.61 Biddy_Managed_SetAlphabeticOrdering()

```
void Biddy_Managed_SetAlphabeticOrdering (
    Biddy_Manager MNG )
```

Description

Named variables are ordered according to their names. Numbered variables are ordered according to their numbers and are always smaller (topmore) as any named variable.

Side effects**More info**

Macro [Bidly_SetAlphabeticOrdering\(\)](#) is defined for use with anonymous manager.

Definition at line 5256 of file `bidlyMain.c`.

3.4.2.62 Bidly_Managed_SetManagerParameters()

```
void Bidly_Managed_SetManagerParameters (
    Bidly_Manager MNG,
    float gcr,
    float gcrF,
    float gcrX,
    float rr,
    float rrF,
    float rrX,
    float st,
    float cst )
```

Description

Function expect 6 float values. If the value is < 0 then the parameter is not modified. The parameters are: `bidlyNodeTable.gcratio` (do not delete nodes if the effect is to small), `bidlyNodeTable.gcratioF` (do not delete nodes if the effect is to small), `bidlyNodeTable.gcratioX` (do not delete nodes if the effect is to small), `bidlyNodeTable.resizeratio` (resize Node table if there are to many nodes), `bidlyNodeTable.resizeratioF` (resize Node table if there are to many nodes), `bidlyNodeTable.resizeratioX` (resize Node table if there are to many nodes), `bidlyNodeTable.siftingtreshold` (stop sifting if the size of the system grows to much), `bidlyNodeTable.fsiftingtreshold` (stop sifting if the size of the function grows to much), `bidlyNodeTable.convergesiftingtreshold` (stop one step of converging sifting if the size of the system grows to much), `bidlyNodeTable.fconvergesiftingtreshold` (stop one step of converging sifting if the size of the function grows to much).

Side effects

Initial values are given in `Bidly_InitMNG`.

More info

Macro [Bidly_SetManagerParameters\(\)](#) is defined for use with anonymous manager.

Definition at line 1219 of file `bidlyMain.c`.

3.4.2.63 Biddy_Managed_SetOrdering()

```
void Biddy_Managed_SetOrdering (
    Biddy_Manager MNG,
    Biddy_String ordering )
```

Description

Non-existing variables included in the string are simply ignored. Not all variables need to be given.

Side effects

String should be formatted in the same way as returned by Biddy_GetOrdering. It should not be prefixed with spaces.

More info

Macro [Biddy_SetOrdering\(ordering\)](#) is defined for use with anonymous manager.

Definition at line 5202 of file biddyMain.c.

3.4.2.64 Biddy_Managed_SetVariableData()

```
void Biddy_Managed_SetVariableData (
    Biddy_Manager MNG,
    Biddy_Variable v,
    void * x )
```

Description

Side effects

It is not checked that the given variable is valid.

More info

Macro [Biddy_SetVariableData\(v,x\)](#) is defined for use with anonymous manager.

Definition at line 2201 of file biddyMain.c.

3.4.2.65 Biddy_Managed_SetVariableValue()

```
void Biddy_Managed_SetVariableValue (
    Biddy_Manager MNG,
    Biddy_Variable v,
    Biddy_Edge f )
```

Description**Side effects**

It is not checked that the given variable is valid.

More info

Macro `Biddy_SetVariableValue(v,f)` is defined for use with anonymous manager.

Definition at line 2115 of file `biddyMain.c`.

3.4.2.66 Biddy_Managed_Sifting()

```
Biddy_Boolean Biddy_Managed_Sifting (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Boolean converge )
```

Description

Variables are reordered globally. All obsolete nodes will be removed.

Side effects

For TZBDD, all unreferenced nodes (not part of registered formulae) will be removed. For TZBDD, sifting may change top edge or even a top node of any function/formula - this is a problem, because functions referenced by local variables only may become wrong. Consequently, for TZBDDs, sifting is not safe to start automatically!

More info

Macro `Biddy_Sifting(f)` is defined for use with anonymous manager.

Definition at line 5431 of file `biddyMain.c`.

3.4.2.67 Biddy_Managed_SwapWithHigher()

```
Biddy_Variable Biddy_Managed_SwapWithHigher (
    Biddy_Manager MNG,
    Biddy_Variable v )
```

Description

Higher (greater) variable is the bottommore one! The highest variable is constant variable "1". Global ordering is number of zeros in corresponding line of `orderingTable`. Constant variable '1' has global ordering greater than all others.

Side effects

All obsolete nodes will be removed.

More info

Macro [Bidly_SwapWithHigher\(v\)](#) is defined for use with anonymous manager.

Definition at line 5295 of file `bidlyMain.c`.

3.4.2.68 Bidly_Managed_SwapWithLower()

```
Bidly_Variable Bidly_Managed_SwapWithLower (
    Bidly_Manager MNG,
    Bidly_Variable v )
```

Description

Lower (smaller) variable is the topmore one! The lowest (topmost) element is not fixed. Topmost variable has global ordering 1 (smaller than all except itself). Global ordering is the number of zeros in corresponding line of orderingTable.

Side effects

All obsolete nodes will be removed.

More info

Macro [Bidly_SwapWithLower\(v\)](#) is defined for use with anonymous manager.

Definition at line 5361 of file `bidlyMain.c`.

3.4.2.69 Bidly_Managed_TaggedFoaNode()

```
Bidly_Edge Bidly_Managed_TaggedFoaNode (
    Bidly_Manager MNG,
    Bidly_Variable v,
    Bidly_Edge pf,
    Bidly_Edge pt,
    Bidly_Variable ptag,
    Bidly_Boolean garbageAllowed )
```

Description

If such node already exists, function returns it and does not create the new one. For OBDDs, ZBDDs, OFDDs, and ZFDDs, the returned edge is not tagged (i.e. `tag == 0`). For TZBDDs and TZFDDs, the returned edge is tagged with the given `ptag`. There are two special cases:

1. If (`pf == pt == NULL`) then new variable (for OBDDs, OFDDs, TZBDDs, and TZFDDs) or new element (for ZBDDs and ZFDDs) is created.
2. (If `ptag == 0`) then the reduction rule and the normalization of complemented edges is not used and the node is added exactly as specified (be careful, this may create a wrong node!).

Side effects

This function should not be called directly to add new variables and elements, you must use `Bidly_Managed_↔_FoaVariable`, `Bidly_Managed_AddVariableByName`, or `Bidly_Managed_AddElementByName`. Using `Bidly_↔_Managed_TaggedFoaNode` you can create node with an arbitrary ordering. It is much more safe to use Boolean operators, e.g. `Bidly_Managed_ITE`.

More info

Macro `Bidly_Managed_FoaNode(MNG,v,pf,pt,garbageAllowed)` is defined for use without tags. Macros `Bidly_↔_TaggedFoaNode(v,pf,pt,tag,garbageAllowed)` and `Bidly_FoaNode(v,pf,pt,garbageAllowed)` are defined for use with anonymous manager.

Definition at line 3269 of file `bidlyMain.c`.

3.4.2.70 Bidly_Managed_TransferMark()

```
Bidly_Edge Bidly_Managed_TransferMark (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Boolean mark,
    Bidly_Boolean leftright )
```

Description

Parameter `leftright` should be TRUE for left and FALSE for right. For OBDDC, it is better to use macro `Bidly_Inv↔_Cond`. For OBDDC, parameter `leftright` is ignored.

Side effects

TO DO: swap the meaning of parameter `leftright` (left should be FALSE)

More info

Macro `Bidly_TransferMark()` is defined for use with anonymous manager.

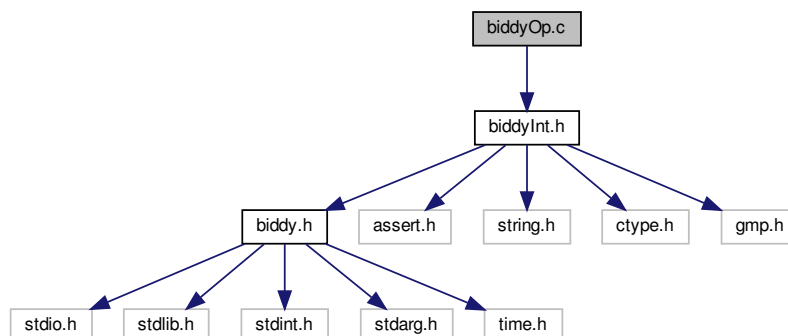
Definition at line 3121 of file `bidlyMain.c`.

3.5 bidlyOp.c File Reference

File `bidlyOp.c` contains functions for operations on various types of Binary Decision Diagrams.

```
#include "bidlyInt.h"
```

Include dependency graph for `bidlyOp.c`:



Functions

- [Bidly_Edge Bidly_Managed_Not](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_Not calculates Boolean function NOT.
- [Bidly_Edge Bidly_Managed_ITE](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g, [Bidly_Edge](#) h)
Function Bidly_Managed_ITE calculates ITE operation of three Boolean functions.
- [Bidly_Edge Bidly_Managed_And](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_And calculates Boolean function AND (conjunction).
- [Bidly_Edge Bidly_Managed_Or](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Or calculates Boolean function OR (disjunction).
- [Bidly_Edge Bidly_Managed_Nand](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Nand calculates Boolean function NAND (Sheffer).
- [Bidly_Edge Bidly_Managed_Nor](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Nor calculates Boolean function NOR (Peirce).
- [Bidly_Edge Bidly_Managed_Xor](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Xor calculates Boolean function XOR.
- [Bidly_Edge Bidly_Managed_Xnor](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Xnor calculates Boolean function XNOR.
- [Bidly_Edge Bidly_Managed_Leq](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Leq calculates Boolean implication.
- [Bidly_Edge Bidly_Managed_Gt](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_Gt calculates the negation of Boolean implication.
- [Bidly_Boolean Bidly_Managed_IsLeq](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g)
Function Bidly_Managed_IsLeq returns TRUE iff function f is included in function g.
- [Bidly_Edge Bidly_Managed_Restrict](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v, [Bidly_↔ Boolean value](#))
Function Bidly_Managed_Restrict calculates a restriction of Boolean function.
- [Bidly_Edge Bidly_Managed_Compose](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g, [Bidly_↔ Variable](#) v)
Function Bidly_Managed_Compose calculates a composition of two Boolean functions.
- [Bidly_Edge Bidly_Managed_E](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v)
Function Bidly_Managed_E calculates an existential quantification of Boolean function.
- [Bidly_Edge Bidly_Managed_A](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v)
Function Bidly_Managed_A calculates an universal quantification of Boolean function.
- [Bidly_Boolean Bidly_Managed_IsVariableDependent](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Variable](#) v)
Function Bidly_Managed_IsVariableDependent returns TRUE iff variable is dependent on others in a function.
- [Bidly_Edge Bidly_Managed_ExistAbstract](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) cube)
Function Bidly_Managed_ExistAbstract existentially abstracts all the variables in cube from f.
- [Bidly_Edge Bidly_Managed_UnivAbstract](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) cube)
Function Bidly_Managed_UnivAbstract universally abstracts all the variables in cube from f.
- [Bidly_Edge Bidly_Managed_AndAbstract](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) g, [Bidly_↔ Edge](#) cube)
Function Bidly_Managed_AndAbstract calculates the AND of two BDDs and simultaneously (existentially) abstracts the variables in cube.
- [Bidly_Edge Bidly_Managed_Constrain](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) c)
Function Bidly_Managed_Constrain calculates Coudert and Madre's constrain function.
- [Bidly_Edge Bidly_Managed_Simplify](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_Edge](#) c)
Function Bidly_Managed_Simplify calculates (a slightly) modified Coudert and Madre's restrict function.
- [Bidly_Edge Bidly_Managed_Support](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_Support calculates a product of all dependent variables (OBDDs and TZBDDs) or the combination set containing a single subset which includes all dependent variables (ZBDDs).

- [Biddy_Edge Biddy_Managed_ReplaceByKeyword](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_String](#) keyword)
Function Biddy_Managed_ReplaceByKeyword calculates Boolean function with one or more variables replaced.
- [Biddy_Edge Biddy_Managed_Change](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Variable](#) v)
Function Biddy_Managed_Change change the form of the given variable (positive literal becomes negative and vice versa).
- [Biddy_Edge Biddy_Managed_VarSubset](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Variable](#) v, [Biddy_↔](#) Boolean value)
Function Biddy_Managed_VarSubset calculates a division of Boolean function with a literal.
- [Biddy_Edge Biddy_Managed_ElementAbstract](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Variable](#) v)
Function Biddy_Managed_ElementAbstract remove element from all combinations in the set.
- [Biddy_Edge Biddy_Managed_Product](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Edge](#) g)
Function Biddy_Managed_Product calculates operation product defined over combination sets.
- [Biddy_Edge Biddy_Managed_SelectiveProduct](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Edge](#) g, [Biddy_Edge](#) cube)
Function Biddy_Managed_SelectiveProduct calculates operation selective product defined over combination sets.
- [Biddy_Edge Biddy_Managed_Supset](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Edge](#) g)
Function Biddy_Managed_Supset calculates Coudert and Madre's operation SupSet.
- [Biddy_Edge Biddy_Managed_Subset](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, [Biddy_Edge](#) g)
Function Biddy_Managed_Subset calculates Coudert and Madre's operation SubSet.
- [Biddy_Edge Biddy_Managed_Permitsym](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f, unsigned int n)
Function Biddy_Managed_Permitsym return a subset of f where only combinations with up to n elements are included.
- [Biddy_Edge Biddy_Managed_Stretch](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) f)
Function Biddy_Managed_Stretch calculates minimal combination set such that all elements in the original set has at least one superset in the new set.
- [Biddy_Edge Biddy_Managed_CreateMinterm](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, long long unsigned int x)
Function Biddy_Managed_CreateMinterm generates one minterm.
- [Biddy_Edge Biddy_Managed_CreateFunction](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, long long unsigned int x)
Function Biddy_Managed_CreateFunction generates one Boolean function.
- [Biddy_Edge Biddy_Managed_RandomFunction](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, double r)
Function Biddy_Managed_RandomFunction generates a random BDD.
- [Biddy_Edge Biddy_Managed_RandomSet](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) unit, double r)
Function Biddy_Managed_RandomSet generates a random BDD.
- [Biddy_Edge Biddy_Managed_ExtractMinterm](#) ([Biddy_Manager](#) MNG, [Biddy_Edge](#) support, [Biddy_Edge](#) f)
Function Biddy_Managed_ExtractMinterm ...

3.5.1 Detailed Description

Description

```

PackageName [Biddy]
Synopsis [Biddy provides data structures and algorithms for the
representation and manipulation of Boolean functions with
ROBDDs, 0-sup-BDDs, and TZBDDs. A hash table is used for quick
search of nodes. Complement edges decreases the number of
nodes. An automatic garbage collection with a system age is
implemented. Variable swapping and sifting are implemented.]

FileName [biddyOp.c]
Revision [${Revision: 545 $}]
Date [${Date: 2019-02-11 14:07:50 +0100 (pon, 11 feb 2019) $}]
Authors [Robert Meolic (robert@meolic.com)]

```

Copyright

Copyright (C) 2006, 2019 UM FERl, Koroska cesta 46, SI-2000 Maribor, Slovenia

Biddy is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Biddy is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

More info

See also: [bidly.h](#), [bidlyInt.h](#)

3.5.2 Function Documentation

3.5.2.1 `Biddy_Managed_A()`

```
Biddy_Edge Biddy_Managed_A (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Variable v )
```

Description

Side effects

Original BDD is not changed. Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD. Be careful: $AxA F \neq \text{NOT}(AxA (\text{NOT } F))$. Counterexample: $Axb (\text{AND } (\text{NOT } a) b c)$.

More info

Macro `Biddy_A(f,v)` is defined for use with anonymous manager.

Definition at line 3826 of file `bidlyOp.c`.

3.5.2.2 `Biddy_Managed_And()`

```
Biddy_Edge Biddy_Managed_And (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

For combination sets, this function coincides with Intersection.

Side Effects

Used by ITE (for OBDD). Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F * H$. For all other BDD types, results are cached as $(f,g,bidder \leftrightarrow \text{Zero})$.

More Info

Macro [Bidder_And\(f,g\)](#) is defined for use with anonymous manager. Macros [Bidder_Managed_Intersect\(MNG,f,g\)](#) and [Bidder_Intersect\(f,g\)](#) are defined for manipulation of combination sets.

Definition at line 812 of file bidderOp.c.

3.5.2.3 Bidder_Managed_AndAbstract()

```
Bidder_Edge Bidder_Managed_AndAbstract (
    Bidder_Manager MNG,
    Bidder_Edge f,
    Bidder_Edge g,
    Bidder_Edge cube )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro [Bidder_AndAbstract\(f,g,cube\)](#) is defined for use with anonymous manager.

Definition at line 4335 of file bidderOp.c.

3.5.2.4 Bidder_Managed_Change()

```
Bidder_Edge Bidder_Managed_Change (
    Bidder_Manager MNG,
    Bidder_Edge f,
    Bidder_Variable v )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters $(f,bidderNull,v)$.

More info

Macro `Biddy_Change()` is defined for use with anonymous manager.

Definition at line 5692 of file `biddyOp.c`.

3.5.2.5 Biddy_Managed_Compose()

```
Biddy_Edge Biddy_Managed_Compose (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g,
    Biddy_Variable v )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDs, recursive calls use optimization: $F(a=G) == \text{NOT}((\text{NOT } F)(a=G))$.

More info

Macro `Biddy_Compose(f,g,v)` is defined for use with anonymous manager.

Definition at line 3383 of file `biddyOp.c`.

3.5.2.6 Biddy_Managed_Constrain()

```
Biddy_Edge Biddy_Managed_Constrain (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge c )
```

Description

Coudert and Madre's constrain function is also called a generalized cofactor of function f with respect to function c . Here is an explanation from <http://gauss.ececs.uc.edu/Courses/c626/lectures/BD↔D/bdd-desc.pdf> "BDD g is a generalized co-factor of f and c if for any truth assignment t , $g(t)$ has the same value as $f(t')$ where t' is the "nearest" truth assignment to t that maps c to 1. By definition, the result of this operation depends on the underlying BDD variable ordering so it cannot be regarded as a logical operation."

Side effects

Original BDD is not changed. Implemented only for OBDD and OBDDC. Cache table is not used.

More info

Macro [Bidly_Constrain\(f,c\)](#) is defined for use with anonymous manager.

Definition at line 4775 of file `bidlyOp.c`.

3.5.2.7 Bidly_Managed_CreateFunction()

```
Bidly_Edge Bidly_Managed_CreateFunction (
    Bidly_Manager MNG,
    Bidly_Edge support,
    long long unsigned int x )
```

Description

The represented Boolean function depends on the variables given with parameter support whilst the parameter `n` determines the generated function.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro [Bidly_CreateFunction\(support,x\)](#) is defined for use with anonymous manager.

Definition at line 7670 of file `bidlyOp.c`.

3.5.2.8 Bidly_Managed_CreateMinterm()

```
Bidly_Edge Bidly_Managed_CreateMinterm (
    Bidly_Manager MNG,
    Bidly_Edge support,
    long long unsigned int x )
```

Description

The represented Boolean function depends on the variables given with parameter support whilst the parameter `n` determines the generated minterm.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro [Bidly_CreateMinterm\(support,x\)](#) is defined for use with anonymous manager.

Definition at line 7590 of file `bidlyOp.c`.

3.5.2.9 Bidly_Managed_E()

```
Bidly_Edge Bidly_Managed_E (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v )
```

Description

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Be careful: $ExA F \neq \text{NOT}(ExA (\text{NOT } F))$. Counterexample: $Exb (\text{AND } (\text{NOT } a) b c)$.

More info

Macro [Bidly_E\(f,v\)](#) is defined for use with anonymous manager.

Definition at line 3618 of file bidlyOp.c.

3.5.2.10 Bidly_Managed_ElementAbstract()

```
Bidly_Edge Bidly_Managed_ElementAbstract (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v )
```

Description

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro [Bidly_ElementAbstract\(f,v\)](#) is defined for use with anonymous manager.

Definition at line 6141 of file bidlyOp.c.

3.5.2.11 Bidly_Managed_ExistAbstract()

```
Bidly_Edge Bidly_Managed_ExistAbstract (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge cube )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD.

More info

Macro `Biddy_ExistAbstract(f,cube)` is defined for use with anonymous manager.

Definition at line 3987 of file `biddyOp.c`.

3.5.2.12 Biddy_Managed_ExtractMinterm()

```
Biddy_Edge Biddy_Managed_ExtractMinterm (
    Biddy_Manager MNG,
    Biddy_Edge support,
    Biddy_Edge f )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Cache table is not used.

More info

Macro `Biddy_ExtractMinterm(f,g)` is defined for use with anonymous manager.

Definition at line 8067 of file `biddyOp.c`.

3.5.2.13 Biddy_Managed_Gt()

```
Biddy_Edge Biddy_Managed_Gt (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

Boolean function $gt(f,g) = \text{and}(f,\text{not}(g)) = \text{not}(\text{leq}(f,g))$. For combination sets, this function coincides with Diff.

Side Effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to $\text{ITE}(F,G,H) = F * G \text{ XOR } F' * H$. For all other BDD types, results are cached as (f,g,g) .

More Info

Macro `Bidly_Gt(f,g)` is defined for use with anonymous manager. Macros `Bidly_Managed_Diff(MNG,f,g)` and `Bidly_Diff(f,g)` are defined for manipulation of combination sets.

Definition at line 2767 of file `bidlyOp.c`.

3.5.2.14 `Bidly_Managed_IsLeq()`

```
Bidly_Boolean Bidly_Managed_IsLeq (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

Side Effects

Prototyped for OBDDs, ZBDDs, and TZBDDs (via calculating full implication, this is less efficient as implementation in CUDD).

More Info

Macro `Bidly_IsLeq(f,g)` is defined for use with anonymous manager.

Definition at line 3091 of file `bidlyOp.c`.

3.5.2.15 `Bidly_Managed_IsVariableDependent()`

```
Bidly_Boolean Bidly_Managed_IsVariableDependent (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v )
```

Description

A variable is dependent on others in a function iff universal quantification of this variable returns constant FALSE.

Side effects

Prototyped for OBDDs (via x_A , calculating full universal quantification is less efficient as direct implementation in CUDD).

More info

Macro `Bidly_IsVariableDependent(f,v)` is defined for use with anonymous manager.

Definition at line 3924 of file `bidlyOp.c`.

3.5.2.16 Biddy_Managed_ITE()

```
Biddy_Edge Biddy_Managed_ITE (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g,
    Biddy_Edge h )
```

Description

Side Effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TzBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F' * H$. For all other BDD types, results are cached as (f,g,h) where $f,g,h \neq 0$, $f \neq g$, $f \neq h$, and $g \neq h$.

More info

Macro [Biddy_ITE\(f,g,h\)](#) is defined for use with anonymous manager.

Definition at line 370 of file `biddyOp.c`.

3.5.2.17 Biddy_Managed_Leq()

```
Biddy_Edge Biddy_Managed_Leq (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

Boolean function $leq(f,g) = \text{or}(\text{not}(f),g) = \text{not}(gt(f,g))$. This function coincides with implication $f \rightarrow g$.

Side Effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TzBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F' * H$. For all other BDD types, results are cached as (f,f,g) .

More Info

Macro [Biddy_Leq\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 2452 of file `biddyOp.c`.

3.5.2.18 Biddy_Managed_Nand()

```
Biddy_Edge Biddy_Managed_Nand (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description**Side Effects**

Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD (via and-not). For $O \leftrightarrow$ BDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F * H$. For OBDD, ZBDD, ZBDDC and TZBDD, results could be cached as (f,g,bidlyNull).

More Info

Macro `Biddy_Nand(f,g)` is defined for use with anonymous manager.

Definition at line 1718 of file bidyOp.c.

3.5.2.19 Biddy_Managed_Nor()

```
Biddy_Edge Biddy_Managed_Nor (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description**Side Effects**

Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD (via or-not). For $O \leftrightarrow$ BDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F * H$. For OBDD, ZBDD, ZBDDC and TZBDD, results could be cached as (bidlyNull,f,g).

More Info

Macro `Biddy_Nor(f,g)` is defined for use with anonymous manager.

Definition at line 1816 of file bidyOp.c.

3.5.2.20 Biddy_Managed_Not()

```
Biddy_Edge Biddy_Managed_Not (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description**Side effects**

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC and OFDDC, it is better to use macro `Biddy_Inv`. For OBDDC, cache table is not needed. For ZBDD and ZBDDC, recursive calls are via `Xor`, its cache table is used. For OBDD and TZBDD, results are cached as (f,bidlyZero,bidlyOne).

More info

Macro [Bidly_Not\(\)](#) is defined for use with anonymous manager.

Definition at line 100 of file `bidlyOp.c`.

3.5.2.21 Bidly_Managed_Or()

```
Bidly_Edge Bidly_Managed_Or (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

For combination sets, this function coincides with Union.

Side Effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F' * H$. For all other BDD types, results are cached as `(bidlyZero,f,g)`.

More Info

Macro [Bidly_Or\(f,g\)](#) is defined for use with anonymous manager. Macros [Bidly_Managed_Union\(MNG,f,g\)](#) and [Bidly_Union\(f,g\)](#) are defined for manipulation of combination sets.

Definition at line 1243 of file `bidlyOp.c`.

3.5.2.22 Bidly_Managed_Permitsym()

```
Bidly_Edge Bidly_Managed_Permitsym (
    Bidly_Manager MNG,
    Bidly_Edge f,
    unsigned int n )
```

Description**Side effects**

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Cache table is not used.

More info

Macro [Bidly_Permitsym\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 7261 of file `bidlyOp.c`.

3.5.2.23 Bidly_Managed_Product()

```
Bidly_Edge Bidly_Managed_Product (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

Product is also called Multiplication.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (f,g,bidlyTerminal).

More info

Macro [Bidly_Product\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 6307 of file bidlyOp.c.

3.5.2.24 Bidly_Managed_RandomFunction()

```
Bidly_Edge Bidly_Managed_RandomFunction (
    Bidly_Manager MNG,
    Bidly_Edge support,
    double r )
```

Description

The represented Boolean function depends on the variables given with parameter support whilst the parameter r determines the ratio between the number of function's minterms and the number of all possible minterms. Parameter support is a product of positive variables - it can be generated with function Bidly_Support.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Parameter r must be a number from [0,1]. Otherwise, function returns bidlyNull.

More info

Macro [Bidly_RandomFunction\(support,r\)](#) is defined for use with anonymous manager.

Definition at line 7767 of file bidlyOp.c.

3.5.2.25 Biddy_Managed_RandomSet()

```
Biddy_Edge Biddy_Managed_RandomSet (
    Biddy_Manager MNG,
    Biddy_Edge unit,
    double r )
```

Description

The represented set is a random combination set determined by the parameter *unit* whilst the parameter *r* determines the ratio between the number of set's subsets and the number of all possible subsets. Parameter *unit* is a set containing a single subset which consists of all elements, i.e. it is a set $\{\{x_1, x_2, \dots, x_n\}\}$ - this is encoded as a product of positive variables and can be generated with function `Biddy_Support`.

Side effects

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Parameter *r* must be a number from [0,1]. Otherwise, function returns `biddyNull`.

More info

Macro `Biddy_RandomSet(unit,r)` is defined for use with anonymous manager.

Definition at line 7908 of file `biddyOp.c`.

3.5.2.26 Biddy_Managed_ReplaceByKeyword()

```
Biddy_Edge Biddy_Managed_ReplaceByKeyword (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_String keyword )
```

Description

Original BDD is not changed. Implemented for OBDD, OBDDC, and TZBDD. Prototyped for ZBDD and ZBDDC (via And-Xor-Not-Restrict). Replacing is controlled by variable's values (which are edges!). For OBDDs and TZBDDs control values must be variables, but for ZBDDs they must be elements! Use `Biddy_ResetVariablesValue` and `Biddy_SetVariableValue` to prepare control values. Parameter *keyword* is used to maintain cache table. If (`keyword == NULL`) then entries in the cache table from previous calculations are deleted.

Side effects

For ZBDD and ZBDDC the function is prototyped and not implemented, yet. Therefore, for ZBDD and ZBDDC, the sets of current and new variables should be disjoint.

More info

Macro `Biddy_ReplaceByKeyword(f,keyword)` is defined for use with anonymous manager. Macros `Biddy_Managed_Replace(MNG,f)` and `Biddy_Replace(f)` are variants with less effective cache table.

Definition at line 5265 of file `biddyOp.c`.

3.5.2.27 Bidly_Managed_Restrict()

```
Bidly_Edge Bidly_Managed_Restrict (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Variable v,
    Bidly_Boolean value )
```

Description

This is not Coudert and Madre's restrict function (use `Bidly_Simplify` if you need that one).

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDs, recursive calls use optimization: $F(a=x) == \text{NOT}(\text{NOT } F(a=x))$.

More info

Macro `Bidly_Restrict(f,v,value)` is defined for use with anonymous manager.

Definition at line 3163 of file `bidlyOp.c`.

3.5.2.28 Bidly_Managed_SelectiveProduct()

```
Bidly_Edge Bidly_Managed_SelectiveProduct (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g,
    Bidly_Edge cube )
```

Description

Selective product is also called Selective multiplication. Combinations are composed only if they agree on variables from cube. This is a non-commutative operation. Variables in Boolean function cube, which are presented in g, must exist in f while, viceversa, this is not required. Moreover, variables which are missing in Boolean function cube and are presented in g must not exist in f (they will be included in the result!) while, viceversa, this is not required. Combinations from f, which are not composed with any combination from g are not included in the result.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (f,g,cube). Because the usage of RC Cache coincide with functions `Bidly_Restrict` and `Bidly_Compose` they should not be used together with `Bidly_SelectiveProduct` in the same manager (the allowed exception is `Restrict` to zero).

More info

Macro `Bidly_SelectiveProduct(f,g,cube)` is defined for use with anonymous manager.

Definition at line 6614 of file `bidlyOp.c`.

3.5.2.29 Biddy_Managed_Simplify()

```
Biddy_Edge Biddy_Managed_Simplify (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge c )
```

Description

Coudert and Madre's restrict function tries to simplify function f by restricting it to the domain covered by function c . No checks are done to see if the result is actually smaller than the input. Here is an explanation from <http://gauss.ececs.uc.edu/Courses/c626/lectures/BDD/bdd-desc.pdf> "Consider the truth tables corresponding to two BDDs f and c over the union of variable sets of both f and c . Build a new BDD g with variable set no larger than the union of the variable sets of f and c and with a truth table such that on rows which c maps to 1 g maps to the same value that f maps to, and on other rows g maps to any value, independent of f . It should be clear that $(f \text{ AND } c)$ and $(g \text{ AND } c)$ are identical so g can replace f in a collection of BDDs without changing its solution space."

Side effects

Original BDD is not changed. Implemented only for OBDD and OBDDC. Cache table is not used.

More info

Macro [Biddy_Simplify\(f,c\)](#) is defined for use with anonymous manager.

Definition at line 4925 of file bidddyOp.c.

3.5.2.30 Biddy_Managed_Stretch()

```
Biddy_Edge Biddy_Managed_Stretch (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (f,bidddyNull,bidddyNull).

More info

Macro [Biddy_Stretch\(f,g\)](#) is defined for use with anonymous manager.

Definition at line 7432 of file bidddyOp.c.

3.5.2.31 `Bidly_Managed_Subset()`

```
Bidly_Edge Bidly_Managed_Subset (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description

The set `SubSet(F,G)` is the set of products of `F` that is contained in at least one product of `G`. For combination sets, function `Subset` coincides with a function which is called a Permission operation. It extracts the product terms from `F` such that the item combination is a subset of at least one item combination in `G`.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters `(f,g,bidlyNull)`.

More info

Macro `Bidly_Subset(f,g)` is defined for use with anonymous manager.

Definition at line 7076 of file `bidlyOp.c`.

3.5.2.32 `Bidly_Managed_Support()`

```
Bidly_Edge Bidly_Managed_Support (
    Bidly_Manager MNG,
    Bidly_Edge f )
```

Description

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDs, dependent variables are exactly those variables existing in the graph. For ZBDDs and TZBDDs, this is not true.

Side effects

For ZBDDs and ZFDDs, variables above the top variable (which are always all dependent) are also included.

More info

Macro `Bidly_Support(f)` is defined for use with anonymous manager.

Definition at line 5065 of file `bidlyOp.c`.

3.5.2.33 Biddy_Managed_Supset()

```
Biddy_Edge Biddy_Managed_Supset (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge g )
```

Description

The set $\text{SupSet}(F,G)$ is the set of products of F that contain at least one product of G . For combination sets, function `Supset` coincides with a function which is called a Restriction operation. It extracts the product terms from F such that the item combination is a superset of at least one item combination in G .

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. RC Cache is used with parameters (biddyNull,f,g).

More info

Macro `Biddy_Supset(f,g)` is defined for use with anonymous manager.

Definition at line 6855 of file `biddyOp.c`.

3.5.2.34 Biddy_Managed_UnivAbstract()

```
Biddy_Edge Biddy_Managed_UnivAbstract (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Edge cube )
```

Description

Side effects

Original BDD is not changed. Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD.

More info

Macro `Biddy_UnivAbstract(f,cube)` is defined for use with anonymous manager.

Definition at line 4240 of file `biddyOp.c`.

3.5.2.35 Biddy_Managed_VarSubset()

```
Biddy_Edge Biddy_Managed_VarSubset (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Variable v,
    Biddy_Boolean value )
```

Description

If (value == FALSE) then calculates $(f|_{\{v=0\}}) * (\text{NOT } v)$. If (value == TRUE) then calculates $(f|_{\{v=1\}}) * v$. This is NOT a Couder and Madre's SubSet operation (which is also called a Permission operation and is a counterpart of a SupSet operation). For combination sets, function Subset coincides with functions Subset0 and Subset1. Moreover, function Offset (also called Modulo or Remainder) is the same as function Subset0, while function Onset (also called Division or Quotient) can be calculated as $\text{Change}(\text{Subset1}(f,v),v)$.

Side effects

Original BDD is not changed. Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. Cache table for AND is used.

More info

Macro `Bidly_VarSubset(f,v,value)` is defined for use with anonymous manager. Macros `Bidly_Managed_Subset0(MNG,f,v)`, `Bidly_Subset0(f,v)`, `Bidly_Managed_Subset1(MNG,f,v)`, and `Bidly_Subset1(f,v)` are defined for manipulation of combination sets. Macros `Bidly_Managed_Quotient(MNG,f,v)`, `Bidly_Quotient(f,v)`, `Bidly_Managed_Remainder(MNG,f,v)`, and `Bidly_Remainder(f,v)` are defined for manipulation of combination sets. Using the provided macros, `Bidly_Managed_Quotient` and `Bidly_Quotient` are not implemented optimally.

Definition at line 5891 of file `bidlyOp.c`.

3.5.2.36 Bidly_Managed_Xnor()

```
Bidly_Edge Bidly_Managed_Xnor (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description**Side Effects**

Implemented for OBDDC. Prototyped for OBDD. Prototyped for ZBDD, ZBDDC and TZBDD (via xor-not). For OBDDC, results are cached as parameters to $\text{ITE}(F,G,H) = F * G \text{ XOR } F' * H$. For OBDD, ZBDD, ZBDDC and TZBDD, results could be cached as $(f, \text{bidlyNull}, g)$.

More Info

Macro `Bidly_Xnor(f,g)` is defined for use with anonymous manager.

Definition at line 2354 of file `bidlyOp.c`.

3.5.2.37 Bidly_Managed_Xor()

```
Bidly_Edge Bidly_Managed_Xor (
    Bidly_Manager MNG,
    Bidly_Edge f,
    Bidly_Edge g )
```

Description**Side Effects**

Used by ITE (for OBDDC). Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. For OBDDC, results are cached as parameters to $ITE(F,G,H) = F * G \text{ XOR } F * H$. For all other BDD types, results are cached as $(f, \text{bidlyZero}, g)$.

More Info

Macro [Bidly_Xor\(f,g\)](#) is defined for use with anonymous manager.

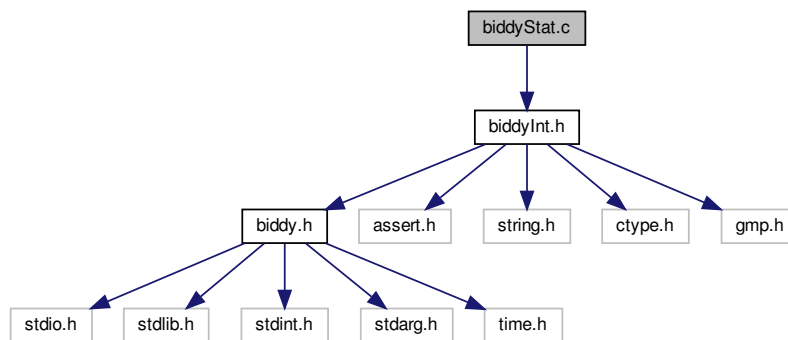
Definition at line 1913 of file `bidlyOp.c`.

3.6 bidlyStat.c File Reference

File [bidlyStat.c](#) contains statistical functions.

```
#include "bidlyInt.h"
```

Include dependency graph for `bidlyStat.c`:

**Functions**

- unsigned int [Bidly_Managed_CountNodes](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_CountNodes.
- unsigned int [Bidly_MaxLevel](#) ([Bidly_Edge](#) f)
Function Bidly_MaxLevel.
- float [Bidly_AvgLevel](#) ([Bidly_Edge](#) f)
Function Bidly_AvgLevel.
- [Bidly_Variable Bidly_Managed_VariableTableNum](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_VariableTableNum returns number of used variables.
- unsigned int [Bidly_Managed_NodeTableSize](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_NodeTableSize returns the size of node table.
- unsigned int [Bidly_Managed_NodeTableBlockNumber](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_NodeTableBlockNumber.
- unsigned int [Bidly_Managed_NodeTableGenerated](#) ([Bidly_Manager](#) MNG)

- Function Bidly_Managed_NodeTableGenerated.*

 - unsigned int [Bidly_Managed_NodeTableMax](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableMax returns maximal (peek) number of nodes in node table.
- unsigned int [Bidly_Managed_NodeTableNum](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableNum returns number of all nodes currently in node table.
- unsigned int [Bidly_Managed_NodeTableNumVar](#) ([Bidly_Manager](#) MNG, [Bidly_Variable](#) v)

Function Bidly_Managed_NodeTableNumVar returns number of nodes with a given variable currently in node table.
- unsigned int [Bidly_Managed_NodeTableResizeNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableResizeNumber.
- unsigned long long int [Bidly_Managed_NodeTableFoaNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableFoaNumber.
- unsigned long long int [Bidly_Managed_NodeTableFindNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableFindNumber.
- unsigned long long int [Bidly_Managed_NodeTableCompareNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableCompareNumber.
- unsigned long long int [Bidly_Managed_NodeTableAddNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableAddNumber.
- unsigned int [Bidly_Managed_NodeTableGCNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableGCNumber.
- unsigned int [Bidly_Managed_NodeTableGCTime](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableGCTime.
- unsigned long long int [Bidly_Managed_NodeTableGCObsoleteNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableGCObsoleteNumber.
- unsigned int [Bidly_Managed_NodeTableSwapNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableSwapNumber.
- unsigned int [Bidly_Managed_NodeTableSiftingNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableSiftingNumber.
- unsigned int [Bidly_Managed_NodeTableDRTime](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableDRTime.
- unsigned int [Bidly_Managed_NodeTableITENumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableITENumber.
- unsigned long long int [Bidly_Managed_NodeTableITERecursiveNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableITERecursiveNumber.
- unsigned int [Bidly_Managed_NodeTableANDORNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableANDORNumber.
- unsigned long long int [Bidly_Managed_NodeTableANDORRecursiveNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableANDORRecursiveNumber.
- unsigned int [Bidly_Managed_NodeTableXORNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableXORNumber.
- unsigned long long int [Bidly_Managed_NodeTableXORRecursiveNumber](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_NodeTableXORRecursiveNumber.
- unsigned int [Bidly_Managed_FormulaTableNum](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_FormulaTableNum returns number of known formulae.
- unsigned int [Bidly_Managed_ListUsed](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_ListUsed.
- unsigned int [Bidly_Managed_ListMaxLength](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_ListMaxLength.
- float [Bidly_Managed_ListAvgLength](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_ListAvgLength.
- unsigned long long int [Bidly_Managed_OPCCacheSearch](#) ([Bidly_Manager](#) MNG)

Function Bidly_Managed_OPCCacheSearch.

- unsigned long long int [Bidly_Managed_OPCacheFind](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_OPCacheFind.
- unsigned long long int [Bidly_Managed_OPCacheInsert](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_OPCacheInsert.
- unsigned long long int [Bidly_Managed_OPCacheOverwrite](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_OPCacheOverwrite.
- unsigned int [Bidly_Managed_CountNodesPlain](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_CountNodesPlain.
- unsigned int [Bidly_Managed_DependentVariableNumber](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, [Bidly_↔](#) Boolean select)
Function Bidly_Managed_DependentVariableNumber.
- unsigned int [Bidly_Managed_CountComplementedEdges](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_CountComplementedEdges count the number of complemented edges.
- unsigned long long int [Bidly_Managed_CountPaths](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_CountPaths count the number of 1-paths.
- double [Bidly_Managed_CountMinterms](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, unsigned int nvars)
Function Bidly_Managed_CountMinterms.
- double [Bidly_Managed_DensityOfFunction](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, unsigned int nvars)
Function Bidly_Managed_DensityOfFunction calculates the ratio of the number of on-set minterms to the number of all minterms.
- double [Bidly_Managed_DensityOfBDD](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f, unsigned int nvars)
Function Bidly_Managed_DensityOfBDD calculates the ratio of the number of on-set minterms to the number of nodes.
- unsigned int [Bidly_Managed_MinNodes](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_MinNodes reports number of nodes in the optimal ordering.
- unsigned int [Bidly_Managed_MaxNodes](#) ([Bidly_Manager](#) MNG, [Bidly_Edge](#) f)
Function Bidly_Managed_MaxNodes reports number of nodes in the worst ordering.
- unsigned long long int [Bidly_Managed_ReadMemoryInUse](#) ([Bidly_Manager](#) MNG)
Function Bidly_Managed_ReadMemoryInUse reports memory consumption of main data structures in bytes (nodes, node table, variable table, ordering table, formula table, ITE cache, EA cache, RC cache, REPLACEcache).
- void [Bidly_Managed_PrintInfo](#) ([Bidly_Manager](#) MNG, FILE *f)
Function Bidly_Managed_PrintInfo prepares a file with stats.

3.6.1 Detailed Description

Description

```

PackageName [Bidly]
Synopsis [Bidly provides data structures and algorithms for the
representation and manipulation of Boolean functions with
ROBDDs, 0-sup-BDDs, and TZBDDs. A hash table is used for quick
search of nodes. Complement edges decreases the number of
nodes. An automatic garbage collection with a system age is
implemented. Variable swapping and sifting are implemented.]

FileName [bidlyStat.c]
Revision [${Revision: 545 $}]
Date [${Date: 2019-02-11 14:07:50 +0100 (pon, 11 feb 2019) $}]
Authors [Robert Meolic (robert@meolic.com),
Ales Casar (ales@homemade.net)]

```

Copyright

Copyright (C) 2006, 2019 UM FERl, Koroska cesta 46, SI-2000 Maribor, Slovenia

Biddy is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Biddy is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

More info

See also: [biddy.h](#), [biddyInt.h](#)

3.6.2 Function Documentation

3.6.2.1 `Biddy_AvgLevel()`

```
float Biddy_AvgLevel (
    Biddy_Edge f )
```

Description

Side effects

The result may not be compatible with your definition of Average Level for DAG. The result is especially problematic if there exist nodes with two equal descendants (e.g for ZBDDs and TZBDDs).

More info

Macro `Biddy_Managed_AvgLevel(f)` is defined for user convenience.

Definition at line 198 of file `biddyStat.c`.

3.6.2.2 `Biddy_Managed_CountComplementedEdges()`

```
unsigned int Biddy_Managed_CountComplementedEdges (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Count number of complemented edges in a given BDD.

Side effects

Terminal 0 is represented by complemented edge to terminal 1 with all BDD types but this edge is counted as a complemented one only if complemented edges are explicitly used.

More info

Macro `Biddy_Managed_CountComplementedEdges(f)` is defined for use with anonymous manager.

Definition at line 1407 of file `biddyStat.c`.

3.6.2.3 Biddy_Managed_CountMinterms()

```
double Biddy_Managed_CountMinterms (
    Biddy_Manager MNG,
    Biddy_Edge f,
    unsigned int nvars )
```

Description

For combination sets, this function coincides with combinations counting. Parameter `nvars` is a user-defined number of dependent variables. If (`nvars == 0`) then all noticeable variables are considered.

Side effects

We are using GNU Multiple Precision Arithmetic Library (GMP). For ZBDDs, this function coincides with the 1-path count. For ZBDDs, result does not depend on the number of dependent variables. For OBDD, noticeable variables are all variables existing in the graph. For TZBDD, noticeable variables are all variables equal or below a top variable (considering the tag). For OBDDs and TZBDDs, if (`nvars == 0`) the result may not be consistent with `Biddy_PrintfMinterms` because this function considers noticeable variables, while `Biddy_PrintfMinterms` considers all created variables.

More info

Macro `Biddy_CountMinterms(f,nvars)` is defined for use with anonymous manager. Macros `Biddy_Managed_CountCombination(MNG,f,nvars)` and `Biddy_CountCombinations(f,nvars)` are defined for use with combination sets.

Definition at line 1539 of file `biddyStat.c`.

3.6.2.4 Biddy_Managed_CountNodes()

```
unsigned int Biddy_Managed_CountNodes (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

Count number of nodes in a BDD.

Side effects

This function must be managed because node selection is used.

More info

Macro `Biddy_CountNodes(f)` is defined for use with anonymous manager.

Definition at line 85 of file `biddyStat.c`.

3.6.2.5 Biddy_Managed_CountNodesPlain()

```
unsigned int Biddy_Managed_CountNodesPlain (  
    Biddy_Manager MNG,  
    Biddy_Edge f )
```

Description

Count number of nodes in a corresponding BDD without complement edges.

Side effects**More info**

Macro `Biddy_Managed_CountNodesPlain(f)` is defined for use with anonymous manager.

Definition at line 1195 of file `biddyStat.c`.

3.6.2.6 Biddy_Managed_CountPaths()

```
unsigned long long int Biddy_Managed_CountPaths (  
    Biddy_Manager MNG,  
    Biddy_Edge f )
```

Description**Side effects**

Implemented for OBDD, OBDDC, ZBDD, ZBDDC, and TZBDD. TO DO: implement this using GNU Multiple Precision Arithmetic Library (GMP).

More info

Macro `Biddy_CountPaths(f)` is defined for use with anonymous manager.

Definition at line 1463 of file `biddyStat.c`.

3.6.2.7 Biddy_Managed_DensityOfBDD()

```
double Biddy_Managed_DensityOfBDD (
    Biddy_Manager MNG,
    Biddy_Edge f,
    unsigned int nvars )
```

Description

If `nvars == 0` then number of dependent variables is used.

Side effects

More info

Macro `Biddy_DensityOfBDD(f,nvars)` is defined for use with anonymous manager.

Definition at line 1756 of file `biddyStat.c`.

3.6.2.8 Biddy_Managed_DensityOfFunction()

```
double Biddy_Managed_DensityOfFunction (
    Biddy_Manager MNG,
    Biddy_Edge f,
    unsigned int nvars )
```

Description

If `nvars == 0` then number of dependent variables is used.

Side effects

More info

Macro `Biddy_DensityOfFunction(f,nvars)` is defined for use with anonymous manager.

Definition at line 1683 of file `biddyStat.c`.

3.6.2.9 Biddy_Managed_DependentVariableNumber()

```
unsigned int Biddy_Managed_DependentVariableNumber (
    Biddy_Manager MNG,
    Biddy_Edge f,
    Biddy_Boolean select )
```

Description

Count number of dependent variables. For OBDDs, the number of dependent variables is the same as the number of variables in the graph. For ZBDDs and TZBDDs, this is not true. If (`select == TRUE`) then dependent variables remain selected otherwise the function will unselect them.

Side effects

For ZBDDs, variables above the top variable (which are always all dependent) are also counted and selected!

More info

Macro `Biddy_DependentVariableNumber(f)` is defined for use with anonymous manager.

Definition at line 1283 of file `biddyStat.c`.

3.6.2.10 `Biddy_Managed_FormulaTableNum()`

```
unsigned int Biddy_Managed_FormulaTableNum (  
    Biddy_Manager MNG )
```

Description**Side effects**

Formulae '0' and '1' are included.

More info

Macro `Biddy_FormulaTableNum()` is defined for use with anonymous manager.

Definition at line 917 of file `biddyStat.c`.

3.6.2.11 `Biddy_Managed_ListAvgLength()`

```
float Biddy_Managed_ListAvgLength (  
    Biddy_Manager MNG )
```

Description**Side effects****More info**

Macro `Biddy_ListAvgLength()` is defined for use with anonymous manager.

Definition at line 1039 of file `biddyStat.c`.

3.6.2.12 `Biddy_Managed_ListMaxLength()`

```
unsigned int Biddy_Managed_ListMaxLength (  
    Biddy_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_ListMaxLength\(\)](#) is defined for use with anonymous manager.

Definition at line 975 of file `biddyStat.c`.

3.6.2.13 Bidly_Managed_ListUsed()

```
unsigned int Bidly_Managed_ListUsed (  
    Bidly_Manager MNG )
```

Description**Side effects****More info**

Macro [Bidly_ListUsed\(\)](#) is defined for use with anonymous manager.

Definition at line 942 of file `biddyStat.c`.

3.6.2.14 Bidly_Managed_MaxNodes()

```
unsigned int Bidly_Managed_MaxNodes (  
    Bidly_Manager MNG,  
    Bidly_Edge f )
```

Description

BDD is copied into new empty manager and then Steinhaus–Johnson–Trotter algorithm is used to check the node number for all possible orderings.

Side effects

Function will finish in a good time only for small number of variables.

More info

Macro [Bidly_MaxNodes\(\)](#) is defined for use with anonymous manager.

Definition at line 1995 of file `biddyStat.c`.

3.6.2.15 Biddy_Managed_MinNodes()

```
unsigned int Biddy_Managed_MinNodes (
    Biddy_Manager MNG,
    Biddy_Edge f )
```

Description

BDD is copied into new empty manager and then Steinhaus–Johnson–Trotter algorithm is used to check the node number for all possible orderings.

Side effects

Function will finish in a good time only for small number of variables.

More info

Macro [Biddy_MinNodes\(\)](#) is defined for use with anonymous manager.

Definition at line 1852 of file biddyStat.c.

3.6.2.16 Biddy_Managed_NodeTableAddNumber()

```
unsigned long long int Biddy_Managed_NodeTableAddNumber (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableAddNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 538 of file biddyStat.c.

3.6.2.17 Biddy_Managed_NodeTableANDORNumber()

```
unsigned int Biddy_Managed_NodeTableANDORNumber (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableANDORNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 798 of file biddyStat.c.

3.6.2.18 Bidly_Managed_NodeTableANDORRecursiveNumber()

```
unsigned long long int Bidly_Managed_NodeTableANDORRecursiveNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

Recursive AND/OR calls are counted only if Bidly is compiled using directive BIDDYEXTENDEDSTATS_YES.

More info

Macro [Bidly_NodeTableANDORRecursiveNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 826 of file biddyStat.c.

3.6.2.19 Bidly_Managed_NodeTableBlockNumber()

```
unsigned int Bidly_Managed_NodeTableBlockNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableBlockNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 287 of file biddyStat.c.

3.6.2.20 Bidly_Managed_NodeTableCompareNumber()

```
unsigned long long int Bidly_Managed_NodeTableCompareNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableCompareNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 506 of file biddyStat.c.

3.6.2.21 **Biddy_Managed_NodeTableDRTime()**

```
unsigned int Biddy_Managed_NodeTableDRTime (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableDRTime\(\)](#) is defined for use with anonymous manager.

Definition at line 713 of file biddyStat.c.

3.6.2.22 **Biddy_Managed_NodeTableFindNumber()**

```
unsigned long long int Biddy_Managed_NodeTableFindNumber (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableFindNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 474 of file biddyStat.c.

3.6.2.23 **Biddy_Managed_NodeTableFoaNumber()**

```
unsigned long long int Biddy_Managed_NodeTableFoaNumber (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableFoaNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 442 of file biddyStat.c.

3.6.2.24 Biddy_Managed_NodeTableGCNumber()

```
unsigned int Biddy_Managed_NodeTableGCNumber (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableGCNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 569 of file `biddyStat.c`.

3.6.2.25 Biddy_Managed_NodeTableGCObsoleteNumber()

```
unsigned long long int Biddy_Managed_NodeTableGCObsoleteNumber (
    Biddy_Manager MNG )
```

Description

Return the number of nodes deleted by GC.

Side effects

Obsolete nodes deleted by GC are counted only if Biddy is compiled using directive `BIDDYEXTENDEDSTATS_↔` YES.

More info

Macro [Biddy_NodeTableGCObsoleteNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 623 of file `biddyStat.c`.

3.6.2.26 Biddy_Managed_NodeTableGCTime()

```
unsigned int Biddy_Managed_NodeTableGCTime (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableGCTime\(\)](#) is defined for use with anonymous manager.

Definition at line 594 of file `biddyStat.c`.

3.6.2.27 Bidly_Managed_NodeTableGenerated()

```
unsigned int Bidly_Managed_NodeTableGenerated (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableGenerated\(\)](#) is defined for use with anonymous manager.

Definition at line 312 of file biddyStat.c.

3.6.2.28 Bidly_Managed_NodeTableITENumber()

```
unsigned int Bidly_Managed_NodeTableITENumber (
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableITENumber\(\)](#) is defined for use with anonymous manager.

Definition at line 738 of file biddyStat.c.

3.6.2.29 Bidly_Managed_NodeTableITERRecursiveNumber()

```
unsigned long long int Bidly_Managed_NodeTableITERRecursiveNumber (
    Bidly_Manager MNG )
```

Description

Side effects

Recursive ITE calls are counted only if Bidly is compiled using directive BIDDYEXTENDEDSTATS_YES.

More info

Macro [Bidly_NodeTableITERRecursiveNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 766 of file biddyStat.c.

3.6.2.30 Biddy_Managed_NodeTableMax()

```
unsigned int Biddy_Managed_NodeTableMax (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableMax\(\)](#) is defined for use with anonymous manager.

Definition at line 338 of file `biddyStat.c`.

3.6.2.31 Biddy_Managed_NodeTableNum()

```
unsigned int Biddy_Managed_NodeTableNum (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableNum\(\)](#) is defined for use with anonymous manager.

Definition at line 364 of file `biddyStat.c`.

3.6.2.32 Biddy_Managed_NodeTableNumVar()

```
unsigned int Biddy_Managed_NodeTableNumVar (  
    Biddy_Manager MNG,  
    Biddy_Variable v )
```

Description

Side effects

More info

Macro [Biddy_NodeTableNumVar\(v\)](#) is defined for use with anonymous manager.

Definition at line 390 of file `biddyStat.c`.

3.6.2.33 Biddy_Managed_NodeTableResizeNumber()

```
unsigned int Biddy_Managed_NodeTableResizeNumber (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableResizeNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 416 of file biddyStat.c.

3.6.2.34 Biddy_Managed_NodeTableSiftingNumber()

```
unsigned int Biddy_Managed_NodeTableSiftingNumber (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableSiftingNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 688 of file biddyStat.c.

3.6.2.35 Biddy_Managed_NodeTableSize()

```
unsigned int Biddy_Managed_NodeTableSize (  
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_NodeTableSize\(\)](#) is defined for use with anonymous manager.

Definition at line 257 of file biddyStat.c.

3.6.2.36 Bidly_Managed_NodeTableSwapNumber()

```
unsigned int Bidly_Managed_NodeTableSwapNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableSwapNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 662 of file bidlyStat.c.

3.6.2.37 Bidly_Managed_NodeTableXORNumber()

```
unsigned int Bidly_Managed_NodeTableXORNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_NodeTableXORNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 857 of file bidlyStat.c.

3.6.2.38 Bidly_Managed_NodeTableXORRecursiveNumber()

```
unsigned long long int Bidly_Managed_NodeTableXORRecursiveNumber (  
    Bidly_Manager MNG )
```

Description

Side effects

Recursive XOR calls are counted only if Bidly is compiled using directive BIDDYEXTENDEDSTATS_YES.

More info

Macro [Bidly_NodeTableXORRecursiveNumber\(\)](#) is defined for use with anonymous manager.

Definition at line 885 of file bidlyStat.c.

3.6.2.39 Bidly_Managed_OPCacheFind()

```
unsigned long long int Bidly_Managed_OPCacheFind (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_OPCacheFind\(\)](#) is defined for use with anonymous manager.

Definition at line 1106 of file biddyStat.c.

3.6.2.40 Bidly_Managed_OPCacheInsert()

```
unsigned long long int Bidly_Managed_OPCacheInsert (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_OPCacheInsert\(\)](#) is defined for use with anonymous manager.

Definition at line 1131 of file biddyStat.c.

3.6.2.41 Bidly_Managed_OPCacheOverwrite()

```
unsigned long long int Bidly_Managed_OPCacheOverwrite (  
    Bidly_Manager MNG )
```

Description

Side effects

More info

Macro [Bidly_OPCacheOverwrite\(\)](#) is defined for use with anonymous manager.

Definition at line 1162 of file biddyStat.c.

3.6.2.42 Biddy_Managed_OPCCacheSearch()

```
unsigned long long int Biddy_Managed_OPCCacheSearch (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_OPCCacheSearch\(\)](#) is defined for use with anonymous manager.

Definition at line 1081 of file biddyStat.c.

3.6.2.43 Biddy_Managed_PrintInfo()

```
void Biddy_Managed_PrintInfo (
    Biddy_Manager MNG,
    FILE * f )
```

Description

Side effects

More info

Macro [Biddy_PrintInfo\(f\)](#) is defined for use with anonymous manager.

Definition at line 2234 of file biddyStat.c.

3.6.2.44 Biddy_Managed_ReadMemoryInUse()

```
unsigned long long int Biddy_Managed_ReadMemoryInUse (
    Biddy_Manager MNG )
```

Description

Side effects

More info

Macro [Biddy_ReadMemoryInUse\(\)](#) is defined for use with anonymous manager.

Definition at line 2137 of file biddyStat.c.

3.6.2.45 Biddy_Managed_VariableTableNum()

```
Biddy_Variable Biddy_Managed_VariableTableNum (
    Biddy_Manager MNG )
```

Description

Side effects

Variable '1' is included.

More info

Macro [Biddy_VariableTableNum\(\)](#) is defined for use with anonymous manager.

Definition at line 232 of file biddyStat.c.

3.6.2.46 Biddy_MaxLevel()

```
unsigned int Biddy_MaxLevel (
    Biddy_Edge f )
```

Description

Side effects

More info

Macro [Biddy_Managed_MaxLevel\(f\)](#) is defined for user convenience.

Definition at line 164 of file biddyStat.c.

Index

biddy.h, 15

- Bidly_About, 68
- Bidly_AddCache, 30
- Bidly_AddElementByName, 30
- Bidly_AddFormula, 31
- Bidly_AddVariableAbove, 31
- Bidly_AddVariableBelow, 31
- Bidly_AddVariableByName, 31
- Bidly_And, 32
- Bidly_AndAbstract, 32
- Bidly_AvgLevel, 68
- Bidly_Change, 32
- Bidly_ChangeVariableName, 32
- Bidly_Clean, 33
- Bidly_ClearMark, 33
- Bidly_ClearTag, 33
- Bidly_ClearVariablesData, 33
- Bidly_Complement, 33
- Bidly_Compose, 34
- Bidly_Constrain, 34
- Bidly_ConstructBDD, 34
- Bidly_Copy, 34
- Bidly_CopyFormulaFrom, 35
- Bidly_CountComplementedEdges, 35
- Bidly_CountMinterms, 35
- Bidly_CountNodes, 35
- Bidly_CountNodesPlain, 35
- Bidly_CountPaths, 36
- Bidly_CreateFunction, 36
- Bidly_CreateMinterm, 36
- Bidly_DeleteFormula, 36
- Bidly_DeletelthFormula, 36
- Bidly_DensityOfBDD, 37
- Bidly_DensityOfFunction, 37
- Bidly_DependentVariableNumber, 37
- Bidly_DeselectAll, 37
- Bidly_DeselectNode, 37
- Bidly_ElementAbstract, 38
- Bidly_Eval, 38
- Bidly_Eval0, 38
- Bidly_Eval1x, 38
- Bidly_Eval2, 39
- Bidly_EvalProbability, 39
- Bidly_ExistAbstract, 39
- Bidly_Exit, 39
- Bidly_ExitMNG, 69
- Bidly_ExtractMinterm, 39
- Bidly_FindFormula, 40
- Bidly_FoaVariable, 40
- Bidly_FormulaTableNum, 40
- Bidly_GC, 40
- Bidly_GetBaseSet, 40
- Bidly_GetConstantOne, 41
- Bidly_GetConstantZero, 41
- Bidly_GetElementEdge, 41
- Bidly_GetElse, 69
- Bidly_GetlthFormula, 41
- Bidly_GetlthFormulaName, 41
- Bidly_GetlthVariable, 42
- Bidly_GetLowestVariable, 42
- Bidly_GetManagerName, 42
- Bidly_GetManagerType, 42
- Bidly_GetMark, 42
- Bidly_GetNextVariable, 43
- Bidly_GetOrdering, 43
- Bidly_GetPrevVariable, 43
- Bidly_GetTag, 43
- Bidly_GetTerminal, 43
- Bidly_GetThen, 70
- Bidly_GetTopVariable, 70
- Bidly_GetTopVariableChar, 44
- Bidly_GetTopVariableEdge, 44
- Bidly_GetTopVariableName, 44
- Bidly_GetVariable, 44
- Bidly_GetVariableData, 44
- Bidly_GetVariableEdge, 45
- Bidly_GetVariableName, 45
- Bidly_GetVariableValue, 45
- Bidly_Gt, 45
- Bidly_ITE, 49
- Bidly_IncTag, 45
- Bidly_Init, 46
- Bidly_InitMNG, 70
- Bidly_Inv, 46
- Bidly_InvCond, 46
- Bidly_InvertMark, 46
- Bidly_IsEqv, 46
- Bidly_IsEqvPointer, 47
- Bidly_IsHighest, 47
- Bidly_IsLeq, 47
- Bidly_IsLowest, 47
- Bidly_IsNull, 47
- Bidly_IsOK, 48
- Bidly_IsSelected, 48
- Bidly_IsSmaller, 48
- Bidly_IsTerminal, 48
- Bidly_IsVariableDependent, 48
- Bidly_Leq, 49
- Bidly_ListAvgLength, 49
- Bidly_ListMaxLength, 49
- Bidly_ListUsed, 49
- Bidly_Managed_AddCache, 71
- Bidly_Managed_AddElementByName, 72
- Bidly_Managed_AddFormula, 72
- Bidly_Managed_AddVariableAbove, 73
- Bidly_Managed_AddVariableBelow, 73
- Bidly_Managed_AddVariableByName, 74
- Bidly_Managed_And, 74
- Bidly_Managed_AndAbstract, 75
- Bidly_Managed_AvgLevel, 50

- Bidly_Managed_Change, 75
- Bidly_Managed_ChangeVariableName, 75
- Bidly_Managed_Clean, 76
- Bidly_Managed_ClearVariablesData, 76
- Bidly_Managed_Compose, 77
- Bidly_Managed_Constrain, 77
- Bidly_Managed_ConstructBDD, 78
- Bidly_Managed_CountComplementedEdges, 78
- Bidly_Managed_CountMinterms, 78
- Bidly_Managed_CountNodes, 79
- Bidly_Managed_CountNodesPlain, 79
- Bidly_Managed_CountPaths, 80
- Bidly_Managed_CreateFunction, 80
- Bidly_Managed_CreateMinterm, 81
- Bidly_Managed_DeleteFormula, 81
- Bidly_Managed_DeletelthFormula, 82
- Bidly_Managed_DensityOfBDD, 82
- Bidly_Managed_DensityOfFunction, 82
- Bidly_Managed_DependentVariableNumber, 83
- Bidly_Managed_DeselectAll, 83
- Bidly_Managed_DeselectNode, 84
- Bidly_Managed_ElementAbstract, 84
- Bidly_Managed_Eval0, 85
- Bidly_Managed_Eval1x, 85
- Bidly_Managed_Eval2, 86
- Bidly_Managed_EvalProbability, 86
- Bidly_Managed_ExistAbstract, 87
- Bidly_Managed_ExtractMinterm, 87
- Bidly_Managed_FindFormula, 87
- Bidly_Managed_FoaVariable, 88
- Bidly_Managed_FormulaTableNum, 88
- Bidly_Managed_GC, 89
- Bidly_Managed_GetBaseSet, 89
- Bidly_Managed_GetConstantOne, 90
- Bidly_Managed_GetConstantZero, 90
- Bidly_Managed_GetElementEdge, 91
- Bidly_Managed_GetElse, 50
- Bidly_Managed_GetlthFormula, 91
- Bidly_Managed_GetlthFormulaName, 91
- Bidly_Managed_GetlthVariable, 92
- Bidly_Managed_GetLowestVariable, 92
- Bidly_Managed_GetManagerName, 93
- Bidly_Managed_GetManagerType, 93
- Bidly_Managed_GetNextVariable, 93
- Bidly_Managed_GetPrevVariable, 94
- Bidly_Managed_GetTerminal, 94
- Bidly_Managed_GetThen, 50
- Bidly_Managed_GetTopVariable, 50
- Bidly_Managed_GetTopVariableChar, 94
- Bidly_Managed_GetTopVariableEdge, 95
- Bidly_Managed_GetTopVariableName, 95
- Bidly_Managed_GetVariable, 95
- Bidly_Managed_GetVariableData, 96
- Bidly_Managed_GetVariableEdge, 96
- Bidly_Managed_GetVariableName, 96
- Bidly_Managed_GetVariableValue, 97
- Bidly_Managed_Gt, 97
- Bidly_Managed_ITE, 101
- Bidly_Managed_IncTag, 98
- Bidly_Managed_IsEqv, 98
- Bidly_Managed_IsHighest, 98
- Bidly_Managed_IsLeq, 99
- Bidly_Managed_IsLowest, 99
- Bidly_Managed_IsOK, 100
- Bidly_Managed_IsSelected, 100
- Bidly_Managed_IsSmaller, 100
- Bidly_Managed_IsVariableDependent, 101
- Bidly_Managed_Leq, 102
- Bidly_Managed_ListAvgLength, 102
- Bidly_Managed_ListMaxLength, 102
- Bidly_Managed_ListUsed, 103
- Bidly_Managed_MaxLevel, 50
- Bidly_Managed_MaxNodes, 104
- Bidly_Managed_MaximizeBDD, 103
- Bidly_Managed_MinNodes, 105
- Bidly_Managed_MinimizeBDD, 104
- Bidly_Managed_Nand, 105
- Bidly_Managed_NodeTableANDORNumber, 106
- Bidly_Managed_NodeTableANDORRecursive↔
Number, 106
- Bidly_Managed_NodeTableAddNumber, 105
- Bidly_Managed_NodeTableBlockNumber, 106
- Bidly_Managed_NodeTableCompareNumber, 107
- Bidly_Managed_NodeTableDRTime, 107
- Bidly_Managed_NodeTableFindNumber, 107
- Bidly_Managed_NodeTableFoaNumber, 108
- Bidly_Managed_NodeTableGCNumber, 108
- Bidly_Managed_NodeTableGCObsoleteNumber,
108
- Bidly_Managed_NodeTableGCTime, 109
- Bidly_Managed_NodeTableGenerated, 109
- Bidly_Managed_NodeTableITENumber, 109
- Bidly_Managed_NodeTableITERecursiveNumber,
110
- Bidly_Managed_NodeTableMax, 110
- Bidly_Managed_NodeTableNum, 110
- Bidly_Managed_NodeTableNumVar, 111
- Bidly_Managed_NodeTableResizeNumber, 111
- Bidly_Managed_NodeTableSiftingNumber, 111
- Bidly_Managed_NodeTableSize, 112
- Bidly_Managed_NodeTableSwapNumber, 112
- Bidly_Managed_NodeTableXORNumber, 112
- Bidly_Managed_NodeTableXORRecursive↔
Number, 113
- Bidly_Managed_Nor, 113
- Bidly_Managed_Not, 114
- Bidly_Managed_OPcacheFind, 114
- Bidly_Managed_OPcacheInsert, 114
- Bidly_Managed_OPcacheOverwrite, 115
- Bidly_Managed_OPcacheSearch, 115
- Bidly_Managed_Or, 115
- Bidly_Managed_Permitsym, 116
- Bidly_Managed_PrintInfo, 117
- Bidly_Managed_PrintfBDD, 116
- Bidly_Managed_PrintfSOP, 117
- Bidly_Managed_PrintfTable, 117

Biddy_Managed_Product, 118
Biddy_Managed_Purge, 118
Biddy_Managed_PurgeAndReorder, 119
Biddy_Managed_RandomFunction, 119
Biddy_Managed_RandomSet, 120
Biddy_Managed_ReadBddview, 120
Biddy_Managed_ReadMemoryInUse, 121
Biddy_Managed_ReadVerilogFile, 121
Biddy_Managed_Refresh, 121
Biddy_Managed_ReplaceByKeyword, 122
Biddy_Managed_ResetVariablesValue, 122
Biddy_Managed_Restrict, 123
Biddy_Managed_SelectFunction, 123
Biddy_Managed_SelectNode, 124
Biddy_Managed_SelectiveProduct, 123
Biddy_Managed_SetAlphabeticOrdering, 124
Biddy_Managed_SetManagerParameters, 125
Biddy_Managed_SetOrdering, 125
Biddy_Managed_SetVariableData, 126
Biddy_Managed_SetVariableValue, 126
Biddy_Managed_Sifting, 127
Biddy_Managed_Simplify, 127
Biddy_Managed_Stretch, 128
Biddy_Managed_Subset, 128
Biddy_Managed_Support, 129
Biddy_Managed_Supset, 129
Biddy_Managed_SwapWithHigher, 130
Biddy_Managed_SwapWithLower, 130
Biddy_Managed_TaggedFoaNode, 130
Biddy_Managed_TransferMark, 131
Biddy_Managed_UnivAbstract, 132
Biddy_Managed_VarSubset, 132
Biddy_Managed_VariableTableNum, 132
Biddy_Managed_WriteBDD, 133
Biddy_Managed_WriteBddview, 133
Biddy_Managed_WriteDot, 134
Biddy_Managed_WriteSOP, 134
Biddy_Managed_WriteTable, 135
Biddy_Managed_Xnor, 135
Biddy_Managed_Xor, 136
Biddy_Managed_A, 71
Biddy_Managed_E, 84
Biddy_MaxLevel, 136
Biddy_MaxNodes, 51
Biddy_MaximizeBDD, 51
Biddy_MinNodes, 51
Biddy_MinimizeBDD, 51
Biddy_Nand, 51
Biddy_NodeTableANDORNumber, 52
Biddy_NodeTableANDORRecursiveNumber, 52
Biddy_NodeTableAddNumber, 52
Biddy_NodeTableBlockNumber, 52
Biddy_NodeTableCompareNumber, 52
Biddy_NodeTableDRTIME, 53
Biddy_NodeTableFindNumber, 53
Biddy_NodeTableFoaNumber, 53
Biddy_NodeTableGCNumber, 53
Biddy_NodeTableGCObsoleteNumber, 53
Biddy_NodeTableGCTime, 54
Biddy_NodeTableGenerated, 54
Biddy_NodeTableITENumber, 54
Biddy_NodeTableITERecursiveNumber, 54
Biddy_NodeTableMax, 54
Biddy_NodeTableNum, 55
Biddy_NodeTableNumVar, 55
Biddy_NodeTableResizeNumber, 55
Biddy_NodeTableSiftingNumber, 55
Biddy_NodeTableSize, 55
Biddy_NodeTableSwapNumber, 56
Biddy_NodeTableXORNumber, 56
Biddy_NodeTableXORRecursiveNumber, 56
Biddy_Nor, 56
Biddy_Not, 56
Biddy_OPCacheFind, 57
Biddy_OPCacheInsert, 57
Biddy_OPCacheOverwrite, 57
Biddy_OPCacheSearch, 57
Biddy_Or, 57
Biddy_Permitsym, 58
Biddy_PrintInfo, 59
Biddy_PrintfBDD, 58
Biddy_PrintfMinterms, 58
Biddy_PrintfSOP, 58
Biddy_PrintfTable, 58
Biddy_Product, 59
Biddy_Purge, 59
Biddy_PurgeAndReorder, 59
Biddy_RandomFunction, 59
Biddy_RandomSet, 60
Biddy_ReadBddview, 60
Biddy_ReadMemoryInUse, 60
Biddy_ReadVerilogFile, 60
Biddy_Refresh, 60
Biddy_Regular, 61
Biddy_ReplaceByKeyword, 61
Biddy_ResetVariablesValue, 61
Biddy_Restrict, 61
Biddy_SelectFunction, 61
Biddy_SelectNode, 62
Biddy_SelectiveProduct, 62
Biddy_SetAlphabeticOrdering, 62
Biddy_SetManagerParameters, 62
Biddy_SetMark, 62
Biddy_SetOrdering, 63
Biddy_SetTag, 63
Biddy_SetVariableData, 63
Biddy_SetVariableValue, 63
Biddy_Sifting, 63
Biddy_Simplify, 64
Biddy_Stretch, 64
Biddy_Subset, 64
Biddy_Support, 64
Biddy_Supset, 64
Biddy_SwapWithHigher, 65
Biddy_SwapWithLower, 65
Biddy_TaggedFoaNode, 65

- Bidly_TransferMark, 65
- Bidly_UnivAbstract, 66
- Bidly_Untagged, 66
- Bidly_VarSubset, 66
- Bidly_VariableTableNum, 66
- Bidly_WriteBDD, 66
- Bidly_WriteBddview, 67
- Bidly_WriteDot, 67
- Bidly_WriteSOP, 67
- Bidly_WriteTable, 67
- Bidly_Xnor, 68
- Bidly_Xor, 68
- Bidly_A, 30
- Bidly_E, 38
- Bidly_About
 - bidly.h, 68
 - bidlyMain.c, 153
- Bidly_AddCache
 - bidly.h, 30
- Bidly_AddElementByName
 - bidly.h, 30
- Bidly_AddFormula
 - bidly.h, 31
- Bidly_AddVariableAbove
 - bidly.h, 31
- Bidly_AddVariableBelow
 - bidly.h, 31
- Bidly_AddVariableByName
 - bidly.h, 31
- Bidly_And
 - bidly.h, 32
- Bidly_AndAbstract
 - bidly.h, 32
- Bidly_AvgLevel
 - bidly.h, 68
 - bidlyStat.c, 205
- Bidly_Boolean, 13
- Bidly_Cache, 13
- Bidly_Change
 - bidly.h, 32
- Bidly_ChangeVariableName
 - bidly.h, 32
- Bidly_Clean
 - bidly.h, 33
- Bidly_ClearMark
 - bidly.h, 33
- Bidly_ClearTag
 - bidly.h, 33
- Bidly_ClearVariablesData
 - bidly.h, 33
- Bidly_Complement
 - bidly.h, 33
- Bidly_Compose
 - bidly.h, 34
- Bidly_Constrain
 - bidly.h, 34
- Bidly_ConstructBDD
 - bidly.h, 34
- Bidly_Copy
 - bidly.h, 34
- Bidly_CopyFormulaFrom
 - bidly.h, 35
- Bidly_CountComplementedEdges
 - bidly.h, 35
- Bidly_CountMinterms
 - bidly.h, 35
- Bidly_CountNodes
 - bidly.h, 35
- Bidly_CountNodesPlain
 - bidly.h, 35
- Bidly_CountPaths
 - bidly.h, 36
- Bidly_CreateFunction
 - bidly.h, 36
- Bidly_CreateMinterm
 - bidly.h, 36
- Bidly_DeleteFormula
 - bidly.h, 36
- Bidly_DeletelthFormula
 - bidly.h, 36
- Bidly_DensityOfBDD
 - bidly.h, 37
- Bidly_DensityOfFunction
 - bidly.h, 37
- Bidly_DependentVariableNumber
 - bidly.h, 37
- Bidly_DeselectAll
 - bidly.h, 37
- Bidly_DeselectNode
 - bidly.h, 37
- Bidly_Edge, 13
- Bidly_ElementAbstract
 - bidly.h, 38
- Bidly_Eval
 - bidly.h, 38
- Bidly_Eval0
 - bidly.h, 38
- Bidly_Eval1x
 - bidly.h, 38
- Bidly_Eval2
 - bidly.h, 39
- Bidly_EvalProbability
 - bidly.h, 39
- Bidly_ExistAbstract
 - bidly.h, 39
- Bidly_Exit
 - bidly.h, 39
- Bidly_ExitMNG
 - bidly.h, 69
 - bidlyMain.c, 153
- Bidly_ExtractMinterm
 - bidly.h, 39
- Bidly_FindFormula
 - bidly.h, 40
- Bidly_FoaVariable
 - bidly.h, 40

Bidly_FormulaTableNum
bidly.h, 40

Bidly_GCFunction, 14

Bidly_GC
bidly.h, 40

Bidly_GetBaseSet
bidly.h, 40

Bidly_GetConstantOne
bidly.h, 41

Bidly_GetConstantZero
bidly.h, 41

Bidly_GetElementEdge
bidly.h, 41

Bidly_GetElse
bidly.h, 69
bidlyMain.c, 153

Bidly_GetlthFormula
bidly.h, 41

Bidly_GetlthFormulaName
bidly.h, 41

Bidly_GetlthVariable
bidly.h, 42

Bidly_GetLowestVariable
bidly.h, 42

Bidly_GetManagerName
bidly.h, 42

Bidly_GetManagerType
bidly.h, 42

Bidly_GetMark
bidly.h, 42

Bidly_GetNextVariable
bidly.h, 43

Bidly_GetOrdering
bidly.h, 43

Bidly_GetPrevVariable
bidly.h, 43

Bidly_GetTag
bidly.h, 43

Bidly_GetTerminal
bidly.h, 43

Bidly_GetThen
bidly.h, 70
bidlyMain.c, 154

Bidly_GetTopVariable
bidly.h, 70
bidlyMain.c, 154

Bidly_GetTopVariableChar
bidly.h, 44

Bidly_GetTopVariableEdge
bidly.h, 44

Bidly_GetTopVariableName
bidly.h, 44

Bidly_GetVariable
bidly.h, 44

Bidly_GetVariableData
bidly.h, 44

Bidly_GetVariableEdge
bidly.h, 45

Bidly_GetVariableName
bidly.h, 45

Bidly_GetVariableValue
bidly.h, 45

Bidly_Gt
bidly.h, 45

Bidly_ITE
bidly.h, 49

Bidly_IncTag
bidly.h, 45

Bidly_Init
bidly.h, 46

Bidly_InitMNG
bidly.h, 70
bidlyMain.c, 155

Bidly_Inv
bidly.h, 46

Bidly_InvCond
bidly.h, 46

Bidly_InvertMark
bidly.h, 46

Bidly_IsEqv
bidly.h, 46

Bidly_IsEqvPointer
bidly.h, 47

Bidly_IsHighest
bidly.h, 47

Bidly_IsLeq
bidly.h, 47

Bidly_IsLowest
bidly.h, 47

Bidly_IsNull
bidly.h, 47

Bidly_IsOK
bidly.h, 48

Bidly_IsSelected
bidly.h, 48

Bidly_IsSmaller
bidly.h, 48

Bidly_IsTerminal
bidly.h, 48

Bidly_IsVariableDependent
bidly.h, 48

Bidly_Leq
bidly.h, 49

Bidly_ListAvgLength
bidly.h, 49

Bidly_ListMaxLength
bidly.h, 49

Bidly_ListUsed
bidly.h, 49

Bidly_LookupFunction, 14

Bidly_Managed_AddCache
bidly.h, 71
bidlyMain.c, 155

Bidly_Managed_AddElementByName
bidly.h, 72
bidlyMain.c, 155

- Bidly_Managed_AddFormula
 - bidly.h, 72
 - bidlyMain.c, 156
- Bidly_Managed_AddVariableAbove
 - bidly.h, 73
 - bidlyMain.c, 157
- Bidly_Managed_AddVariableBelow
 - bidly.h, 73
 - bidlyMain.c, 157
- Bidly_Managed_AddVariableByName
 - bidly.h, 74
 - bidlyMain.c, 157
- Bidly_Managed_And
 - bidly.h, 74
 - bidlyOp.c, 185
- Bidly_Managed_AndAbstract
 - bidly.h, 75
 - bidlyOp.c, 186
- Bidly_Managed_AvgLevel
 - bidly.h, 50
- Bidly_Managed_Change
 - bidly.h, 75
 - bidlyOp.c, 186
- Bidly_Managed_ChangeVariableName
 - bidly.h, 75
 - bidlyMain.c, 158
- Bidly_Managed_Clean
 - bidly.h, 76
 - bidlyMain.c, 158
- Bidly_Managed_ClearVariablesData
 - bidly.h, 76
 - bidlyMain.c, 159
- Bidly_Managed_Compose
 - bidly.h, 77
 - bidlyOp.c, 187
- Bidly_Managed_Constrain
 - bidly.h, 77
 - bidlyOp.c, 187
- Bidly_Managed_ConstructBDD
 - bidly.h, 78
 - bidlyMain.c, 159
- Bidly_Managed_CountComplementedEdges
 - bidly.h, 78
 - bidlyStat.c, 205
- Bidly_Managed_CountMinterms
 - bidly.h, 78
 - bidlyStat.c, 206
- Bidly_Managed_CountNodes
 - bidly.h, 79
 - bidlyStat.c, 206
- Bidly_Managed_CountNodesPlain
 - bidly.h, 79
 - bidlyStat.c, 207
- Bidly_Managed_CountPaths
 - bidly.h, 80
 - bidlyStat.c, 207
- Bidly_Managed_CreateFunction
 - bidly.h, 80
- bidlyOp.c, 188
- Bidly_Managed_CreateMinterm
 - bidly.h, 81
 - bidlyOp.c, 188
- Bidly_Managed_DeleteFormula
 - bidly.h, 81
 - bidlyMain.c, 159
- Bidly_Managed_DeletelthFormula
 - bidly.h, 82
 - bidlyMain.c, 160
- Bidly_Managed_DensityOfBDD
 - bidly.h, 82
 - bidlyStat.c, 207
- Bidly_Managed_DensityOfFunction
 - bidly.h, 82
 - bidlyStat.c, 208
- Bidly_Managed_DependentVariableNumber
 - bidly.h, 83
 - bidlyStat.c, 208
- Bidly_Managed_DeselectAll
 - bidly.h, 83
 - bidlyMain.c, 160
- Bidly_Managed_DeselectNode
 - bidly.h, 84
 - bidlyMain.c, 161
- Bidly_Managed_ElementAbstract
 - bidly.h, 84
 - bidlyOp.c, 189
- Bidly_Managed_Eval0
 - bidly.h, 85
 - bidlyInOut.c, 139
- Bidly_Managed_Eval1x
 - bidly.h, 85
 - bidlyInOut.c, 139
- Bidly_Managed_Eval2
 - bidly.h, 86
 - bidlyInOut.c, 139
- Bidly_Managed_EvalProbability
 - bidly.h, 86
 - bidlyMain.c, 161
- Bidly_Managed_ExistAbstract
 - bidly.h, 87
 - bidlyOp.c, 189
- Bidly_Managed_ExtractMinterm
 - bidly.h, 87
 - bidlyOp.c, 190
- Bidly_Managed_FindFormula
 - bidly.h, 87
 - bidlyMain.c, 162
- Bidly_Managed_FoaVariable
 - bidly.h, 88
 - bidlyMain.c, 162
- Bidly_Managed_FormulaTableNum
 - bidly.h, 88
 - bidlyStat.c, 209
- Bidly_Managed_GC
 - bidly.h, 89
 - bidlyMain.c, 163

Bidly_Managed_GetBaseSet
bidly.h, 89
bidlyMain.c, 163

Bidly_Managed_GetConstantOne
bidly.h, 90
bidlyMain.c, 164

Bidly_Managed_GetConstantZero
bidly.h, 90
bidlyMain.c, 164

Bidly_Managed_GetElementEdge
bidly.h, 91
bidlyMain.c, 164

Bidly_Managed_GetElse
bidly.h, 50

Bidly_Managed_GetlthFormula
bidly.h, 91
bidlyMain.c, 165

Bidly_Managed_GetlthFormulaName
bidly.h, 91
bidlyMain.c, 165

Bidly_Managed_GetlthVariable
bidly.h, 92
bidlyMain.c, 166

Bidly_Managed_GetLowestVariable
bidly.h, 92
bidlyMain.c, 166

Bidly_Managed_GetManagerName
bidly.h, 93
bidlyMain.c, 166

Bidly_Managed_GetManagerType
bidly.h, 93
bidlyMain.c, 167

Bidly_Managed_GetNextVariable
bidly.h, 93
bidlyMain.c, 167

Bidly_Managed_GetPrevVariable
bidly.h, 94
bidlyMain.c, 167

Bidly_Managed_GetTerminal
bidly.h, 94
bidlyMain.c, 168

Bidly_Managed_GetThen
bidly.h, 50

Bidly_Managed_GetTopVariable
bidly.h, 50

Bidly_Managed_GetTopVariableChar
bidly.h, 94
bidlyMain.c, 168

Bidly_Managed_GetTopVariableEdge
bidly.h, 95
bidlyMain.c, 168

Bidly_Managed_GetTopVariableName
bidly.h, 95
bidlyMain.c, 169

Bidly_Managed_GetVariable
bidly.h, 95
bidlyMain.c, 169

Bidly_Managed_GetVariableData
bidly.h, 96
bidlyMain.c, 170

Bidly_Managed_GetVariableEdge
bidly.h, 96
bidlyMain.c, 170

Bidly_Managed_GetVariableName
bidly.h, 96
bidlyMain.c, 170

Bidly_Managed_GetVariableValue
bidly.h, 97
bidlyMain.c, 171

Bidly_Managed_Gt
bidly.h, 97
bidlyOp.c, 190

Bidly_Managed_ITE
bidly.h, 101
bidlyOp.c, 191

Bidly_Managed_IncTag
bidly.h, 98
bidlyMain.c, 171

Bidly_Managed_IsEqv
bidly.h, 98
bidlyMain.c, 171

Bidly_Managed_IsHighest
bidly.h, 98
bidlyMain.c, 172

Bidly_Managed_IsLeq
bidly.h, 99
bidlyOp.c, 191

Bidly_Managed_IsLowest
bidly.h, 99
bidlyMain.c, 172

Bidly_Managed_IsOK
bidly.h, 100
bidlyMain.c, 173

Bidly_Managed_IsSelected
bidly.h, 100
bidlyMain.c, 173

Bidly_Managed_IsSmaller
bidly.h, 100
bidlyMain.c, 173

Bidly_Managed_IsVariableDependent
bidly.h, 101
bidlyOp.c, 191

Bidly_Managed_Leq
bidly.h, 102
bidlyOp.c, 192

Bidly_Managed_ListAvgLength
bidly.h, 102
bidlyStat.c, 209

Bidly_Managed_ListMaxLength
bidly.h, 102
bidlyStat.c, 209

Bidly_Managed_ListUsed
bidly.h, 103
bidlyStat.c, 210

Bidly_Managed_MaxLevel
bidly.h, 50

Biddy_Managed_MaxNodes
 biddy.h, 104
 biddyStat.c, 210
 Biddy_Managed_MaximizeBDD
 biddy.h, 103
 biddyMain.c, 174
 Biddy_Managed_MinNodes
 biddy.h, 105
 biddyStat.c, 210
 Biddy_Managed_MinimizeBDD
 biddy.h, 104
 biddyMain.c, 174
 Biddy_Managed_Nand
 biddy.h, 105
 biddyOp.c, 192
 Biddy_Managed_NodeTableANDORNumber
 biddy.h, 106
 biddyStat.c, 211
 Biddy_Managed_NodeTableANDORRecursiveNumber
 biddy.h, 106
 biddyStat.c, 211
 Biddy_Managed_NodeTableAddNumber
 biddy.h, 105
 biddyStat.c, 211
 Biddy_Managed_NodeTableBlockNumber
 biddy.h, 106
 biddyStat.c, 212
 Biddy_Managed_NodeTableCompareNumber
 biddy.h, 107
 biddyStat.c, 212
 Biddy_Managed_NodeTableDRTime
 biddy.h, 107
 biddyStat.c, 212
 Biddy_Managed_NodeTableFindNumber
 biddy.h, 107
 biddyStat.c, 213
 Biddy_Managed_NodeTableFoaNumber
 biddy.h, 108
 biddyStat.c, 213
 Biddy_Managed_NodeTableGCNumber
 biddy.h, 108
 biddyStat.c, 213
 Biddy_Managed_NodeTableGCObsoleteNumber
 biddy.h, 108
 biddyStat.c, 214
 Biddy_Managed_NodeTableGCTime
 biddy.h, 109
 biddyStat.c, 214
 Biddy_Managed_NodeTableGenerated
 biddy.h, 109
 biddyStat.c, 214
 Biddy_Managed_NodeTableITENumber
 biddy.h, 109
 biddyStat.c, 215
 Biddy_Managed_NodeTableITERRecursiveNumber
 biddy.h, 110
 biddyStat.c, 215
 Biddy_Managed_NodeTableMax
 biddy.h, 110
 biddyStat.c, 215
 Biddy_Managed_NodeTableNum
 biddy.h, 110
 biddyStat.c, 216
 Biddy_Managed_NodeTableNumVar
 biddy.h, 111
 biddyStat.c, 216
 Biddy_Managed_NodeTableResizeNumber
 biddy.h, 111
 biddyStat.c, 216
 Biddy_Managed_NodeTableSiftingNumber
 biddy.h, 111
 biddyStat.c, 217
 Biddy_Managed_NodeTableSize
 biddy.h, 112
 biddyStat.c, 217
 Biddy_Managed_NodeTableSwapNumber
 biddy.h, 112
 biddyStat.c, 217
 Biddy_Managed_NodeTableXORNumber
 biddy.h, 112
 biddyStat.c, 218
 Biddy_Managed_NodeTableXORRecursiveNumber
 biddy.h, 113
 biddyStat.c, 218
 Biddy_Managed_Nor
 biddy.h, 113
 biddyOp.c, 193
 Biddy_Managed_Not
 biddy.h, 114
 biddyOp.c, 193
 Biddy_Managed_OPCCacheFind
 biddy.h, 114
 biddyStat.c, 218
 Biddy_Managed_OPCCacheInsert
 biddy.h, 114
 biddyStat.c, 219
 Biddy_Managed_OPCCacheOverwrite
 biddy.h, 115
 biddyStat.c, 219
 Biddy_Managed_OPCCacheSearch
 biddy.h, 115
 biddyStat.c, 219
 Biddy_Managed_Or
 biddy.h, 115
 biddyOp.c, 194
 Biddy_Managed_Permitsym
 biddy.h, 116
 biddyOp.c, 194
 Biddy_Managed_PrintInfo
 biddy.h, 117
 biddyStat.c, 220
 Biddy_Managed_PrintfBDD
 biddy.h, 116
 biddyInOut.c, 140
 Biddy_Managed_PrintfSOP
 biddy.h, 117

- bidlyInOut.c, [140](#)
- Bidly_Managed_PrintfTable
 - bidly.h, [117](#)
 - bidlyInOut.c, [141](#)
- Bidly_Managed_Product
 - bidly.h, [118](#)
 - bidlyOp.c, [194](#)
- Bidly_Managed_Purge
 - bidly.h, [118](#)
 - bidlyMain.c, [175](#)
- Bidly_Managed_PurgeAndReorder
 - bidly.h, [119](#)
 - bidlyMain.c, [175](#)
- Bidly_Managed_RandomFunction
 - bidly.h, [119](#)
 - bidlyOp.c, [195](#)
- Bidly_Managed_RandomSet
 - bidly.h, [120](#)
 - bidlyOp.c, [195](#)
- Bidly_Managed_ReadBddview
 - bidly.h, [120](#)
 - bidlyInOut.c, [141](#)
- Bidly_Managed_ReadMemoryInUse
 - bidly.h, [121](#)
 - bidlyStat.c, [220](#)
- Bidly_Managed_ReadVerilogFile
 - bidly.h, [121](#)
 - bidlyInOut.c, [142](#)
- Bidly_Managed_Refresh
 - bidly.h, [121](#)
 - bidlyMain.c, [176](#)
- Bidly_Managed_ReplaceByKeyword
 - bidly.h, [122](#)
 - bidlyOp.c, [196](#)
- Bidly_Managed_ResetVariablesValue
 - bidly.h, [122](#)
 - bidlyMain.c, [176](#)
- Bidly_Managed_Restrict
 - bidly.h, [123](#)
 - bidlyOp.c, [196](#)
- Bidly_Managed_SelectFunction
 - bidly.h, [123](#)
 - bidlyMain.c, [177](#)
- Bidly_Managed_SelectNode
 - bidly.h, [124](#)
 - bidlyMain.c, [177](#)
- Bidly_Managed_SelectiveProduct
 - bidly.h, [123](#)
 - bidlyOp.c, [197](#)
- Bidly_Managed_SetAlphabeticOrdering
 - bidly.h, [124](#)
 - bidlyMain.c, [177](#)
- Bidly_Managed_SetManagerParameters
 - bidly.h, [125](#)
 - bidlyMain.c, [178](#)
- Bidly_Managed_SetOrdering
 - bidly.h, [125](#)
 - bidlyMain.c, [178](#)
- Bidly_Managed_SetVariableData
 - bidly.h, [126](#)
 - bidlyMain.c, [179](#)
- Bidly_Managed_SetVariableValue
 - bidly.h, [126](#)
 - bidlyMain.c, [179](#)
- Bidly_Managed_Sifting
 - bidly.h, [127](#)
 - bidlyMain.c, [180](#)
- Bidly_Managed_Simplify
 - bidly.h, [127](#)
 - bidlyOp.c, [197](#)
- Bidly_Managed_Stretch
 - bidly.h, [128](#)
 - bidlyOp.c, [198](#)
- Bidly_Managed_Subset
 - bidly.h, [128](#)
 - bidlyOp.c, [198](#)
- Bidly_Managed_Support
 - bidly.h, [129](#)
 - bidlyOp.c, [199](#)
- Bidly_Managed_Supset
 - bidly.h, [129](#)
 - bidlyOp.c, [199](#)
- Bidly_Managed_SwapWithHigher
 - bidly.h, [130](#)
 - bidlyMain.c, [180](#)
- Bidly_Managed_SwapWithLower
 - bidly.h, [130](#)
 - bidlyMain.c, [181](#)
- Bidly_Managed_TaggedFoaNode
 - bidly.h, [130](#)
 - bidlyMain.c, [181](#)
- Bidly_Managed_TransferMark
 - bidly.h, [131](#)
 - bidlyMain.c, [182](#)
- Bidly_Managed_UnivAbstract
 - bidly.h, [132](#)
 - bidlyOp.c, [200](#)
- Bidly_Managed_VarSubset
 - bidly.h, [132](#)
 - bidlyOp.c, [200](#)
- Bidly_Managed_VariableTableNum
 - bidly.h, [132](#)
 - bidlyStat.c, [220](#)
- Bidly_Managed_WriteBDD
 - bidly.h, [133](#)
 - bidlyInOut.c, [142](#)
- Bidly_Managed_WriteBddview
 - bidly.h, [133](#)
 - bidlyInOut.c, [142](#)
- Bidly_Managed_WriteDot
 - bidly.h, [134](#)
 - bidlyInOut.c, [143](#)
- Bidly_Managed_WriteSOP
 - bidly.h, [134](#)
 - bidlyInOut.c, [143](#)
- Bidly_Managed_WriteTable

- bidly.h, 135
- bidlyInOut.c, 144
- Bidly_Managed_Xnor
 - bidly.h, 135
 - bidlyOp.c, 201
- Bidly_Managed_Xor
 - bidly.h, 136
 - bidlyOp.c, 201
- Bidly_Managed_A
 - bidly.h, 71
 - bidlyOp.c, 185
- Bidly_Managed_E
 - bidly.h, 84
 - bidlyOp.c, 188
- Bidly_Manager, 14
- Bidly_MaxLevel
 - bidly.h, 136
 - bidlyStat.c, 221
- Bidly_MaxNodes
 - bidly.h, 51
- Bidly_MaximizeBDD
 - bidly.h, 51
- Bidly_MinNodes
 - bidly.h, 51
- Bidly_MinimizeBDD
 - bidly.h, 51
- Bidly_Nand
 - bidly.h, 51
- Bidly_NodeTableANDORNumber
 - bidly.h, 52
- Bidly_NodeTableANDORRecursiveNumber
 - bidly.h, 52
- Bidly_NodeTableAddNumber
 - bidly.h, 52
- Bidly_NodeTableBlockNumber
 - bidly.h, 52
- Bidly_NodeTableCompareNumber
 - bidly.h, 52
- Bidly_NodeTableDRTIME
 - bidly.h, 53
- Bidly_NodeTableFindNumber
 - bidly.h, 53
- Bidly_NodeTableFoaNumber
 - bidly.h, 53
- Bidly_NodeTableGCNumber
 - bidly.h, 53
- Bidly_NodeTableGCObsoleteNumber
 - bidly.h, 53
- Bidly_NodeTableGCTime
 - bidly.h, 54
- Bidly_NodeTableGenerated
 - bidly.h, 54
- Bidly_NodeTableITENumber
 - bidly.h, 54
- Bidly_NodeTableITERRecursiveNumber
 - bidly.h, 54
- Bidly_NodeTableMax
 - bidly.h, 54
- Bidly_NodeTableNum
 - bidly.h, 55
- Bidly_NodeTableNumVar
 - bidly.h, 55
- Bidly_NodeTableResizeNumber
 - bidly.h, 55
- Bidly_NodeTableSiftingNumber
 - bidly.h, 55
- Bidly_NodeTableSize
 - bidly.h, 55
- Bidly_NodeTableSwapNumber
 - bidly.h, 56
- Bidly_NodeTableXORNumber
 - bidly.h, 56
- Bidly_NodeTableXORRecursiveNumber
 - bidly.h, 56
- Bidly_Nor
 - bidly.h, 56
- Bidly_Not
 - bidly.h, 56
- Bidly_OPcacheFind
 - bidly.h, 57
- Bidly_OPcacheInsert
 - bidly.h, 57
- Bidly_OPcacheOverwrite
 - bidly.h, 57
- Bidly_OPcacheSearch
 - bidly.h, 57
- Bidly_Or
 - bidly.h, 57
- Bidly_Permitsym
 - bidly.h, 58
- Bidly_PrintInfo
 - bidly.h, 59
- Bidly_PrintfBDD
 - bidly.h, 58
- Bidly_PrintfMinterms
 - bidly.h, 58
- Bidly_PrintfSOP
 - bidly.h, 58
- Bidly_PrintfTable
 - bidly.h, 58
- Bidly_Product
 - bidly.h, 59
- Bidly_Purge
 - bidly.h, 59
- Bidly_PurgeAndReorder
 - bidly.h, 59
- Bidly_RandomFunction
 - bidly.h, 59
- Bidly_RandomSet
 - bidly.h, 60
- Bidly_ReadBddview
 - bidly.h, 60
- Bidly_ReadMemoryInUse
 - bidly.h, 60
- Bidly_ReadVerilogFile
 - bidly.h, 60

- Bidly_Refresh
 - [bidly.h, 60](#)
- Bidly_Regular
 - [bidly.h, 61](#)
- Bidly_ReplaceByKeyword
 - [bidly.h, 61](#)
- Bidly_ResetVariablesValue
 - [bidly.h, 61](#)
- Bidly_Restrict
 - [bidly.h, 61](#)
- Bidly_SelectFunction
 - [bidly.h, 61](#)
- Bidly_SelectNode
 - [bidly.h, 62](#)
- Bidly_SelectiveProduct
 - [bidly.h, 62](#)
- Bidly_SetAlphabeticOrdering
 - [bidly.h, 62](#)
- Bidly_SetManagerParameters
 - [bidly.h, 62](#)
- Bidly_SetMark
 - [bidly.h, 62](#)
- Bidly_SetOrdering
 - [bidly.h, 63](#)
- Bidly_SetTag
 - [bidly.h, 63](#)
- Bidly_SetVariableData
 - [bidly.h, 63](#)
- Bidly_SetVariableValue
 - [bidly.h, 63](#)
- Bidly_Sifting
 - [bidly.h, 63](#)
- Bidly_Simplify
 - [bidly.h, 64](#)
- Bidly_Stretch
 - [bidly.h, 64](#)
- Bidly_String, [15](#)
- Bidly_Subset
 - [bidly.h, 64](#)
- Bidly_Support
 - [bidly.h, 64](#)
- Bidly_Supset
 - [bidly.h, 64](#)
- Bidly_SwapWithHigher
 - [bidly.h, 65](#)
- Bidly_SwapWithLower
 - [bidly.h, 65](#)
- Bidly_TaggedFoaNode
 - [bidly.h, 65](#)
- Bidly_TransferMark
 - [bidly.h, 65](#)
- Bidly_UnivAbstract
 - [bidly.h, 66](#)
- Bidly_Untagged
 - [bidly.h, 66](#)
- Bidly_VarSubset
 - [bidly.h, 66](#)
- Bidly_Variable, [15](#)
- Bidly_VariableTableNum
 - [bidly.h, 66](#)
- Bidly_WriteBDD
 - [bidly.h, 66](#)
- Bidly_WriteBddview
 - [bidly.h, 67](#)
- Bidly_WriteDot
 - [bidly.h, 67](#)
- Bidly_WriteSOP
 - [bidly.h, 67](#)
- Bidly_WriteTable
 - [bidly.h, 67](#)
- Bidly_Xnor
 - [bidly.h, 68](#)
- Bidly_Xor
 - [bidly.h, 68](#)
- Bidly_A
 - [bidly.h, 30](#)
- Bidly_E
 - [bidly.h, 38](#)
- [bidlyInOut.c, 137](#)
 - [Bidly_Managed_Eval0, 139](#)
 - [Bidly_Managed_Eval1x, 139](#)
 - [Bidly_Managed_Eval2, 139](#)
 - [Bidly_Managed_PrintfBDD, 140](#)
 - [Bidly_Managed_PrintfSOP, 140](#)
 - [Bidly_Managed_PrintfTable, 141](#)
 - [Bidly_Managed_ReadBddview, 141](#)
 - [Bidly_Managed_ReadVerilogFile, 142](#)
 - [Bidly_Managed_WriteBDD, 142](#)
 - [Bidly_Managed_WriteBddview, 142](#)
 - [Bidly_Managed_WriteDot, 143](#)
 - [Bidly_Managed_WriteSOP, 143](#)
 - [Bidly_Managed_WriteTable, 144](#)
 - [VerilogFileGateName, 144](#)
- [bidlyInt.h, 145](#)
- [bidlyMain.c, 148](#)
 - [Bidly_About, 153](#)
 - [Bidly_ExitMNG, 153](#)
 - [Bidly_GetElse, 153](#)
 - [Bidly_GetThen, 154](#)
 - [Bidly_GetTopVariable, 154](#)
 - [Bidly_InitMNG, 155](#)
 - [Bidly_Managed_AddCache, 155](#)
 - [Bidly_Managed_AddElementByName, 155](#)
 - [Bidly_Managed_AddFormula, 156](#)
 - [Bidly_Managed_AddVariableAbove, 157](#)
 - [Bidly_Managed_AddVariableBelow, 157](#)
 - [Bidly_Managed_AddVariableByName, 157](#)
 - [Bidly_Managed_ChangeVariableName, 158](#)
 - [Bidly_Managed_Clean, 158](#)
 - [Bidly_Managed_ClearVariablesData, 159](#)
 - [Bidly_Managed_ConstructBDD, 159](#)
 - [Bidly_Managed_DeleteFormula, 159](#)
 - [Bidly_Managed_DeletelthFormula, 160](#)
 - [Bidly_Managed_DeselectAll, 160](#)
 - [Bidly_Managed_DeselectNode, 161](#)
 - [Bidly_Managed_EvalProbability, 161](#)

- Bidly_Managed_FindFormula, 162
- Bidly_Managed_FoaVariable, 162
- Bidly_Managed_GC, 163
- Bidly_Managed_GetBaseSet, 163
- Bidly_Managed_GetConstantOne, 164
- Bidly_Managed_GetConstantZero, 164
- Bidly_Managed_GetElementEdge, 164
- Bidly_Managed_GetlthFormula, 165
- Bidly_Managed_GetlthFormulaName, 165
- Bidly_Managed_GetlthVariable, 166
- Bidly_Managed_GetLowestVariable, 166
- Bidly_Managed_GetManagerName, 166
- Bidly_Managed_GetManagerType, 167
- Bidly_Managed_GetNextVariable, 167
- Bidly_Managed_GetPrevVariable, 167
- Bidly_Managed_GetTerminal, 168
- Bidly_Managed_GetTopVariableChar, 168
- Bidly_Managed_GetTopVariableEdge, 168
- Bidly_Managed_GetTopVariableName, 169
- Bidly_Managed_GetVariable, 169
- Bidly_Managed_GetVariableData, 170
- Bidly_Managed_GetVariableEdge, 170
- Bidly_Managed_GetVariableName, 170
- Bidly_Managed_GetVariableValue, 171
- Bidly_Managed_IncTag, 171
- Bidly_Managed_IsEqv, 171
- Bidly_Managed_IsHighest, 172
- Bidly_Managed_IsLowest, 172
- Bidly_Managed_IsOK, 173
- Bidly_Managed_IsSelected, 173
- Bidly_Managed_IsSmaller, 173
- Bidly_Managed_MaximizeBDD, 174
- Bidly_Managed_MinimizeBDD, 174
- Bidly_Managed_Purge, 175
- Bidly_Managed_PurgeAndReorder, 175
- Bidly_Managed_Refresh, 176
- Bidly_Managed_ResetVariablesValue, 176
- Bidly_Managed_SelectFunction, 177
- Bidly_Managed_SelectNode, 177
- Bidly_Managed_SetAlphabeticOrdering, 177
- Bidly_Managed_SetManagerParameters, 178
- Bidly_Managed_SetOrdering, 178
- Bidly_Managed_SetVariableData, 179
- Bidly_Managed_SetVariableValue, 179
- Bidly_Managed_Sifting, 180
- Bidly_Managed_SwapWithHigher, 180
- Bidly_Managed_SwapWithLower, 181
- Bidly_Managed_TaggedFoaNode, 181
- Bidly_Managed_TransferMark, 182
- bidlyOp.c, 182
 - Bidly_Managed_And, 185
 - Bidly_Managed_AndAbstract, 186
 - Bidly_Managed_Change, 186
 - Bidly_Managed_Compose, 187
 - Bidly_Managed_Constrain, 187
 - Bidly_Managed_CreateFunction, 188
 - Bidly_Managed_CreateMinterm, 188
 - Bidly_Managed_ElementAbstract, 189
 - Bidly_Managed_ExistAbstract, 189
 - Bidly_Managed_ExtractMinterm, 190
 - Bidly_Managed_Gt, 190
 - Bidly_Managed_ITE, 191
 - Bidly_Managed_IsLeq, 191
 - Bidly_Managed_IsVariableDependent, 191
 - Bidly_Managed_Leq, 192
 - Bidly_Managed_Nand, 192
 - Bidly_Managed_Nor, 193
 - Bidly_Managed_Not, 193
 - Bidly_Managed_Or, 194
 - Bidly_Managed_Permitsym, 194
 - Bidly_Managed_Product, 194
 - Bidly_Managed_RandomFunction, 195
 - Bidly_Managed_RandomSet, 195
 - Bidly_Managed_ReplaceByKeyword, 196
 - Bidly_Managed_Restrict, 196
 - Bidly_Managed_SelectiveProduct, 197
 - Bidly_Managed_Simplify, 197
 - Bidly_Managed_Stretch, 198
 - Bidly_Managed_Subset, 198
 - Bidly_Managed_Support, 199
 - Bidly_Managed_Supset, 199
 - Bidly_Managed_UnivAbstract, 200
 - Bidly_Managed_VarSubset, 200
 - Bidly_Managed_Xnor, 201
 - Bidly_Managed_Xor, 201
 - Bidly_Managed_A, 185
 - Bidly_Managed_E, 188
- bidlyStat.c, 202
 - Bidly_AvgLevel, 205
 - Bidly_Managed_CountComplementedEdges, 205
 - Bidly_Managed_CountMinterms, 206
 - Bidly_Managed_CountNodes, 206
 - Bidly_Managed_CountNodesPlain, 207
 - Bidly_Managed_CountPaths, 207
 - Bidly_Managed_DensityOfBDD, 207
 - Bidly_Managed_DensityOfFunction, 208
 - Bidly_Managed_DependentVariableNumber, 208
 - Bidly_Managed_FormulaTableNum, 209
 - Bidly_Managed_ListAvgLength, 209
 - Bidly_Managed_ListMaxLength, 209
 - Bidly_Managed_ListUsed, 210
 - Bidly_Managed_MaxNodes, 210
 - Bidly_Managed_MinNodes, 210
 - Bidly_Managed_NodeTableANDORNumber, 211
 - Bidly_Managed_NodeTableANDORRecursive↔
Number, 211
 - Bidly_Managed_NodeTableAddNumber, 211
 - Bidly_Managed_NodeTableBlockNumber, 212
 - Bidly_Managed_NodeTableCompareNumber, 212
 - Bidly_Managed_NodeTableDRTTime, 212
 - Bidly_Managed_NodeTableFindNumber, 213
 - Bidly_Managed_NodeTableFoaNumber, 213
 - Bidly_Managed_NodeTableGCNumber, 213
 - Bidly_Managed_NodeTableGCObsoleteNumber,
214
 - Bidly_Managed_NodeTableGCTime, 214

Biddy_Managed_NodeTableGenerated, [214](#)
Biddy_Managed_NodeTableITENumber, [215](#)
Biddy_Managed_NodeTableITERecursiveNumber,
[215](#)
Biddy_Managed_NodeTableMax, [215](#)
Biddy_Managed_NodeTableNum, [216](#)
Biddy_Managed_NodeTableNumVar, [216](#)
Biddy_Managed_NodeTableResizeNumber, [216](#)
Biddy_Managed_NodeTableSiftingNumber, [217](#)
Biddy_Managed_NodeTableSize, [217](#)
Biddy_Managed_NodeTableSwapNumber, [217](#)
Biddy_Managed_NodeTableXORNumber, [218](#)
Biddy_Managed_NodeTableXORRecursive↔
Number, [218](#)
Biddy_Managed_OPCacheFind, [218](#)
Biddy_Managed_OPCacheInsert, [219](#)
Biddy_Managed_OPCacheOverwrite, [219](#)
Biddy_Managed_OPCacheSearch, [219](#)
Biddy_Managed_PrintInfo, [220](#)
Biddy_Managed_ReadMemoryInUse, [220](#)
Biddy_Managed_VariableTableNum, [220](#)
Biddy_MaxLevel, [221](#)

VerilogFileGateName
bidlyInOut.c, [144](#)