

ABC

Presenter: 杜冠樺 Instructor: 江介宏

Outline

- Introduction
- Installation
 - ▣ ABC
 - ▣ Additional Tools
- Commands
 - ▣ ABC commands
 - ▣ User commands

Introduction

□ ABC

- A System for Sequential Synthesis and Verification
- Support BLIF, PLA, CNF, AIGER, Verilog...etc.
- What can it do:
 - Logic restructuring (SOP, AIG, BDD...)
 - Technology mapping
 - SAT
 - Model checking
 - ...

Installation - ABC

□ ABC source code:

- <http://www.eecs.berkeley.edu/~alanmi/abc/>
- Recommended compiler :
 - Windows : Microsoft Visual Studio 6.0
 - Unix/Linux : gcc or g++

The image displays two screenshots related to the ABC source code. The left screenshot shows the Bitbucket repository page for 'alanmi / ABC'. It includes a 'Choose one to download' annotation pointing to the 'get source' button. The right screenshot shows the 'Contents' page of the ABC website, with a 'Getting ABC' section highlighted. This section provides instructions on how to download the latest version of ABC from the Bitbucket repository and lists several earlier versions available for download.

Contents

- Introduction
- Highlights
- Next steps
- Getting ABC
 - Download summary
 - Basic
 - Input
 - Output
 - Printing
 - Synthesis
 - Comb
 - Seq
 - Mapping
 - LUTs

ABC

A System for Sequential Synthesis and Verification

Berkeley Logic Synthesis and Verification Group

ABC is a growing software system for synthesis and verification of binary sequential logic circuits appearing in synchronous hardware designs. ABC combines scalable logic optimization based on And-Inverter Graphs (AIGs), optimal-delay DAG-based technology mapping for look-up tables and standard cells, and innovative algorithms for sequential synthesis and verification.

ABC provides an experimental implementation of these algorithms and a programming environment for building similar applications. Future development will focus on improving the algorithms and making most of the packages stand-alone. This will allow the user to customize ABC for their needs as if it were a tool-box rather than a complete tool.

Getting ABC

The latest version of ABC can be downloaded from <http://bitbucket.org/alanmi/abc/>.

Several earlier versions are also available: [abc51205](#), [abc61225](#), [abc70930](#). Here is [abc70930](#) that works on Mac OS with Xcode (kindly provided by S. M. Rayan Kabir).

A recent [Windows binary](#) and [resource file](#) are also available. To run the binary, place the resource file in the same directory.

In papers and reports, please refer to ABC as follows: **Berkeley Logic Synthesis and Verification Group, ABC: A System for Sequential Synthesis and Verification, Release YMMDD**. <http://www.eecs.berkeley.edu/~alanmi/abc/>

In the above reference, instead of YMMDD, substitute the number of the release version used, abbreviated as follows: Y[year]M[month]D[day]

Author	Revision	Message	Date
alanmi	6ced8e142152	Misc changes.	2012-05-21
alanmi	631a68008b0c	Changing 'Y' to allow for delay optimization on sequential paths only.	2012-05-20
alanmi	9a681338e412	Changing 'Y' to allow for delay optimization on sequential paths only.	2012-05-20
alanmi	e971218329cb	Do not allow quitting bmc3 after exploring 2^num_of frames if jump-forward is	2012-05-20
alanmi	be61429b20c6	Misc changes.	2012-05-19

Installation - Additional Tools

□ Graphviz:

- ▣ Program for generating PostScript file.
- ▣ Needed by commands “show” and “show_bdd”
- ▣ <http://www.graphviz.org/>

□ GSview:

- ▣ Program for opening PostScript file
- ▣ <http://pages.cs.wisc.edu/~ghost/gsview/>

□ Ghostscript:

- ▣ script needed for GSview
- ▣ <http://pages.cs.wisc.edu/~ghost/doc/GPL/index.htm>

Commands

- After compiling, you can type “./abc” to run ABC in shell mode.

```
[~] 22:11 ~/ABC/alanmi-abc-aced8e142152]$ ./abc
UC Berkeley, ABC 1.01 (compiled Jun  4 2012 21:13:32)
abc 01> █
```

Commands

- `command -h` : Print help information for “*command*”
- Basic commands:
 - ▣ `help` : Present all commands
 - ▣ `quit` : Exit the ABC
- I/O commands:
 - ▣ `read_blif filename` : Read in blif file
 - ▣ `write_blif filename` : Write out blif file

Commands

- Printing commands:
 - ▣ `show` : Show the network by PostScript file
 - ▣ `show_bdd` : Show BDD by PostScript file
 - ▣ `print_stats` : Print network information
- Synthesis commands:
 - ▣ `collapse` : Translate network to BDD
 - ▣ `strash` : Translate network to AIG

Commands

□ Add new command:

▣ src/base/abci/abc.c :

■ Define new function.

```
static int Abc_CommandAbc9Undo      ( Abc_Frame_t * pAbc, int argc, char ** argv );
static int Abc_CommandAbc9Iso      ( Abc_Frame_t * pAbc, int argc, char ** argv );
static int Abc_CommandAbc9Test     ( Abc_Frame_t * pAbc, int argc, char ** argv );

static int Abc_CommandAbcTestNew   ( Abc_Frame_t * pAbc, int argc, char ** argv );
```

■ Add the function into Abc_Init.

```
void Abc_Init( Abc_Frame_t * pAbc )
{
    Cmd_CommandAdd( pAbc, "Printing", "print_stats", Abc_CommandPrintStats, 0 );
    Cmd_CommandAdd( pAbc, "Printing", "print_exdc", Abc_CommandPrintExdc, 0 );
    Cmd_CommandAdd( pAbc, "Printing", "print_io", Abc_CommandPrintIo, 0 );
}
```

■ Implement

```
int Abc_CommandTest( Abc_Frame_t * pAbc, int argc, char ** argv )
{
    ...
}
```

Q&A