Example Program for the Embecosm Proxy RSP Server

Generated by Doxygen 1.5.7.1

Sun Apr 12 18:15:46 2009

Contents

Chapter 1

Example Program for the Embecosm Proxy GDB RSP Server

1.1 About

This is a simple program which can be compiled with the OpenRISC 1000 toolchain to create a binary image compatible with the Proxy RSP Server configuration file "or1k.cfg".

1.2 Installation

This directory is not intended to be installed directly. If the OpenRISC 1000 toolchain is stalled, the binary can be rebuilt using the Makefile. However unless the toolchain matches exactly that used when creating the example, the result may not be compatible with or1k.cfg. For this reason, the binary image (file hello) is also distributed.

The Makefile may need editing to change the name and/or location of the OpenRISC 1000 compiler and linker tools.

The Makefile target "doc" can be used to rebuild this Doxygen documentation. It requires doxygen (see <http://www.doxygen.org/>) to be installed. PDF output also requires TeX (see <http://www.tug.org/>) to be installed.

1.3 Copying

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

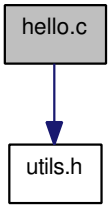
Chapter 3

File Documentation

3.1 hello.c File Reference

```
#include "utils.h"
```

Include dependency graph for hello.c:



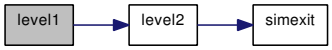
Functions

- void level2 ()
- void level1 ()
- main ()

3.1.1 Function Documentation

3.1.1.1 void level1 ()

Here is the call graph for this function:

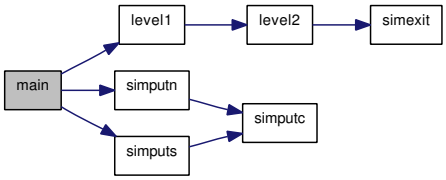


3.1.1.2 void level2 ()

Here is the call graph for this function:

3.1.1.3 main ()

Here is the call graph for this function:



3.2 mainpage File Reference

3.2.1 Detailed Description

3.3 utils.c File Reference

Defines

- #define NOP_NOP 0x0000
- #define NOP_EXIT 0x0001
- #define NOP_REPORT 0x0002
- #define NOP_PRINTF 0x0003
- #define NOP_PUTC 0x0004
- #define NOP_CNT_RESET 0x0005

Functions

- void simexit (int rc)
- void simputc (int c)
- void simputh (unsigned long int i)
- void simputn (unsigned long int i)
- void simputs (char *str)

3.3.1 Define Documentation

3.3.1.1 #define NOP_CNT_RESET 0x0005

3.3.1.2 #define NOP_EXIT 0x0001

3.3.1.3 #define NOP_NOP 0x0000

3.3.1.4 #define NOP_PRINTF 0x0003

3.3.1.5 #define NOP_PUTC 0x0004

3.3.1.6 #define NOP_REPORT 0x0002

3.3.2 Function Documentation

3.3.2.1 void simexit (int *rc*)

Exit the simulator ! !

Parameters:

rc Return code (not used)

3.3.2.2 void simputc (int *c*)

Print a character ! !

Parameters:

c Character to print

3.3.2.3 void simputh (unsigned long int *i*)

Print a hex number ! !

Parameters:

i Number to print

Here is the call graph for this function:

3.3.2.4 void simputn (unsigned long int *i*)

Print a decimal number ! !

Parameters:

i Number to print

Here is the call graph for this function:

3.3.2.5 void simputs (char * *str*)

Print a string ! !

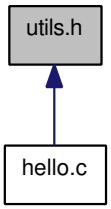
Parameters:

str String to print

Here is the call graph for this function:

3.4 utils.h File Reference

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



Functions

- void simexit (int rc)
- void simputc (int c)
- void simputh (int i)
- void simputs (char ∗str)

3.4.1 Function Documentation

3.4.1.1 void simexit (int *rc*)

Exit the simulator ! !

Parameters:

rc Return code (not used)

3.4.1.2 void simputc (int *c*)

Print a character ! !

Parameters:

c Character to print

3.4.1.3 void simputh (int *i*)

3.4.1.4 void simputs (char ∗ *str*)

Print a string ! !

Parameters:

str String to print

Here is the call graph for this function:

Index