TCP_SERVER

1.0

Generated by Doxygen 1.9.1

1 Data Structure Documentation 1

1.1 s_process_data Struct Reference . . . 1

1.1.1 Field Documentation . . . 1

1.1.1.1 p_data . . . 1

1.1.1.2 p_ringbuffer . . . 1

1.1.1.3 thread . . . 1

1.2 s_send_tcp_server Struct Reference . . . 2

1.2.1 Field Documentation . . . 2

1.2.1.1 connection_thread . . . 2

1.2.1.2 kill_thread . . . 2

1.2.1.3 p_address . . . 3

1.2.1.4 p_socket_info . . . 3

1.2.1.5 poll_connection . . . 3

1.2.1.6 port . . . 3

1.2.1.7 recv_process_data . . . 3

1.2.1.8 send_process_data . . . 3

1.3 s_vpi_data Struct Reference . . . 3

1.3.1 Field Documentation . . . 4

1.3.1.1 arg1_handle . . . 4

1.3.1.2 arg2_handle . . . 4

1.3.1.3 array_byte_size . . . 4

1.3.1.4 error . . . 4

1.3.1.5 num_ab_val_pairs . . . 4

1.3.1.6 systf_handle . . . 4

2 File Documentation 5

2.1 messages.h File Reference . . . 5

2.1.1 Detailed Description . . . 5

2.1.2 Function Documentation . . . 6

2.1.2.1 print_error() . . . 6

2.1.2.2 print_info() . . . 6

2.2 tcp_server.c File Reference . . . 6

2.2.1 Function Documentation . . . 7

2.2.1.1 connection_keep_alive() . . . 7

2.2.1.2 end_tcp_server() . . . 7

2.2.1.3 setup_tcp_server() . . . 7

2.2.1.4 start_tcp_server() . . . 8

2.2.2 Variable Documentation . . . 8

2.2.2.1 g_num_of_connections . . . 8

2.2.2.2 g_send_tcp_server . . . 8

2.3 tcp_server.h File Reference . . . 8

2.3.1 Detailed Description . . . 10

2.3.2 Macro Definition Documentation . . . 11
2.3.2.1 BUFFSIZE . . . 11
2.3.2.2 DATACHUNK . . . 11
2.3.2.3 MAX_CONNECTIONS . . . 11
2.3.3 Function Documentation . . . 11
2.3.3.1 end_tcp_server() . . . 11
2.3.3.2 setup_tcp_server() . . . 11
2.3.3.3 start_tcp_server() . . . 12
2.3.4 Variable Documentation . . . 12
2.3.4.1 g_send_tcp_server . . . 12
2.4 vpi_messages.c File Reference . . . 12
2.4.1 Detailed Description . . . 13
2.4.2 Function Documentation . . . 13
2.4.2.1 print_error() . . . 13
2.4.2.2 print_info() . . . 13
2.5 vpi_recv_tcp_server.c File Reference . . . 14
2.5.1 Detailed Description . . . 14
2.5.2 Function Documentation . . . 15
2.5.2.1 recv_tcp_server_calltf() . . . 15
2.5.2.2 recv_tcp_server_compiletf() . . . 15
2.5.2.3 recv_tcp_server_end_sim_cb() . . . 15
2.5.2.4 recv_tcp_server_start_sim_cb() . . . 15
2.5.2.5 recv_thread() . . . 15
2.6 vpi_recv_tcp_server.h File Reference . . . 16
2.6.1 Detailed Description . . . 16
2.6.2 Function Documentation . . . 17
2.6.2.1 recv_tcp_server_calltf() . . . 17
2.6.2.2 recv_tcp_server_compiletf() . . . 17
2.7 vpi_send_tcp_server.c File Reference . . . 18
2.7.1 Detailed Description . . . 18
2.7.2 Function Documentation . . . 19
2.7.2.1 send_tcp_server_calltf() . . . 19
2.7.2.2 send_tcp_server_compiletf() . . . 19
2.7.2.3 send_tcp_server_end_sim_cb() . . . 19
2.7.2.4 send_tcp_server_start_sim_cb() . . . 19
2.7.2.5 send_thread() . . . 19
2.8 vpi_send_tcp_server.h File Reference . . . 20
2.8.1 Detailed Description . . . 20
2.8.2 Function Documentation . . . 21
2.8.2.1 send_tcp_server_calltf() . . . 21
2.8.2.2 send_tcp_server_compiletf() . . . 21
2.9 vpi_tcp_server.c File Reference . . . 21

2.9.1 Function Documentation . . . 22
2.9.1.1 recv_tcp_server_reg_systf() . . . 22
2.9.1.2 send_tcp_server_reg_systf() . . . 22
2.9.1.3 setup_tcp_server_calltf() . . . 23
2.9.1.4 setup_tcp_server_compiletf() . . . 23
2.9.1.5 setup_tcp_server_end_sim_cb() . . . 23
2.9.1.6 setup_tcp_server_reg_systf() . . . 23
2.9.1.7 setup_tcp_server_start_sim_cb() . . . 23
2.9.1.8 tcp_server_sizetf() . . . 23
2.9.2 Variable Documentation . . . 24
2.9.2.1 vlog_startup_routines . . . 24
2.10 vpi_tcp_server.h File Reference . . . 24
2.10.1 Detailed Description . . . 25
2.10.2 Macro Definition Documentation . . . 26
2.10.2.1 RECV_NAME . . . 26
2.10.2.2 SEND_NAME . . . 26
2.10.2.3 SETUP_NAME . . . 26

Index **27**

Chapter 1

Data Structure Documentation

1.1 s_process_data Struct Reference

`#include <tcp_server.h>`

Data Fields

- struct s_ringBuffer * p_ringbuffer
- pthread_t thread
- void * p_data

1.1.1 Field Documentation

1.1.1.1 p_data

`void* s_process_data::p_data`

1.1.1.2 p_ringbuffer

`struct s_ringBuffer* s_process_data::p_ringbuffer`

1.1.1.3 thread

`pthread_t s_process_data::thread`

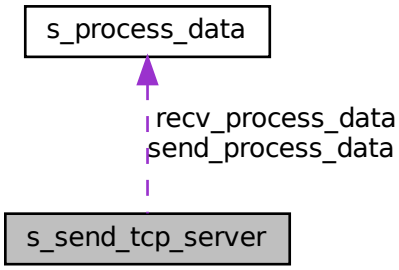
The documentation for this struct was generated from the following file:

- tcp_server.h

1.2 s_send_tcp_server Struct Reference

`#include <tcp_server.h>`

Collaboration diagram for s_send_tcp_server:



Data Fields

- int kill_thread
- struct pollfd poll_connection
- struct sockaddr_in * p_socket_info
- pthread_t connection_thread
- char * p_address
- unsigned short port
- struct s_process_data recv_process_data
- struct s_process_data send_process_data

1.2.1 Field Documentation

1.2.1.1 connection_thread

`pthread_t s_send_tcp_server::connection_thread`

1.2.1.2 kill_thread

`int s_send_tcp_server::kill_thread`

1.2.1.3 p_address

```
char* s_send_tcp_server::p_address
```

1.2.1.4 p_socket_info

```
struct sockaddr_in* s_send_tcp_server::p_socket_info
```

1.2.1.5 poll_connection

```
struct pollfd s_send_tcp_server::poll_connection
```

1.2.1.6 port

```
unsigned short s_send_tcp_server::port
```

1.2.1.7 recv_process_data

```
struct s_process_data s_send_tcp_server::recv_process_data
```

1.2.1.8 send_process_data

```
struct s_process_data s_send_tcp_server::send_process_data
```

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

- tcp_server.h

1.3 s_vpi_data Struct Reference

```
#include <vpi_tcp_server.h>
```

Data Fields

- PLI_INT32 error
- PLI_INT32 num_ab_val_pairs
- PLI_INT32 array_byte_size
- vpiHandle systf_handle
- vpiHandle arg1_handle
- vpiHandle arg2_handle

1.3.1 Field Documentation

1.3.1.1 arg1_handle

```
vpiHandle s_vpi_data::arg1_handle
```

1.3.1.2 arg2_handle

```
vpiHandle s_vpi_data::arg2_handle
```

1.3.1.3 array_byte_size

```
PLI_INT32 s_vpi_data::array_byte_size
```

1.3.1.4 error

```
PLI_INT32 s_vpi_data::error
```

1.3.1.5 num_ab_val_pairs

```
PLI_INT32 s_vpi_data::num_ab_val_pairs
```

1.3.1.6 systf_handle

```
vpiHandle s_vpi_data::systf_handle
```

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

- vpi_tcp_server.h

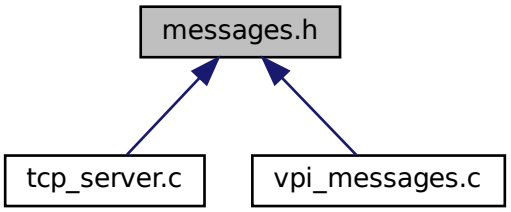
Chapter 2

File Documentation

2.1 messages.h File Reference

Functions to create multiple TCP servers.

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



Functions

- int print_error (const char *format,...)
- int print_info (const char *format,...)

2.1.1 Detailed Description

Functions to create multiple TCP servers.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-03-02

2.1.2 Function Documentation

2.1.2.1 print_error()

```
int print_error (
            const char * format,
             ...  )
```

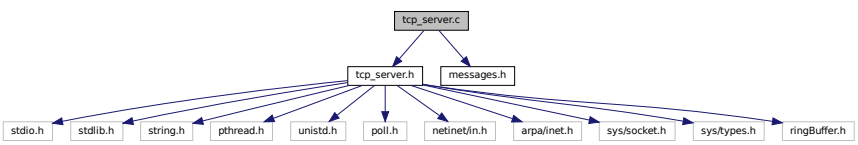
2.1.2.2 print_info()

```
int print_info (
            const char * format,
             ...  )
```

2.2 tcp_server.c File Reference

```
#include "tcp_server.h"
#include "messages.h"
```

Include dependency graph for tcp_server.c:



Functions

- int * setup_tcp_server (char *p_address, int port)
  SETUP TCP SERVER.
- int start_tcp_server (int *p_index)
  START TCP SERVER.
- int end_tcp_server (int *p_index)
  END TCP SERVER.
- void * connection_keep_alive (void *p_data)

Variables

- unsigned int g_num_of_connections = 0
- struct s_send_tcp_server g_send_tcp_server [MAX_CONNECTIONS]

2.2.1 Function Documentation

2.2.1.1 connection_keep_alive()

```
void * connection_keep_alive (
            void * p_data )
```

2.2.1.2 end_tcp_server()

```
int end_tcp_server (
            int * p_index )
```

END TCP SERVER.

2.2.1.3 setup_tcp_server()

```
int * setup_tcp_server (
            char * p_address,
            int port )
```

SETUP TCP SERVER.

2.2.1.4 start_tcp_server()

```
int start_tcp_server (
            int * p_index )
```

START TCP SERVER.

2.2.2 Variable Documentation

2.2.2.1 g_num_of_connections

```
unsigned int g_num_of_connections = 0
```

2.2.2.2 g_send_tcp_server

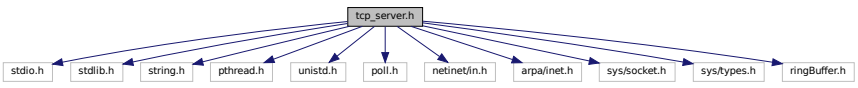
```
struct s_send_tcp_server g_send_tcp_server[MAX_CONNECTIONS]
```

2.3 tcp_server.h File Reference

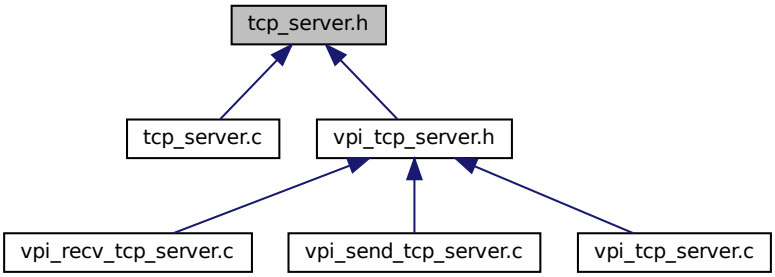
Functions to write raw binary files properly in verilog.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <unistd.h>
#include <poll.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/types.h>
#include "ringBuffer.h"
```

Include dependency graph for tcp_server.h:



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



Data Structures

- struct s_process_data
- struct s_send_tcp_server

Macros

- #define BUFFSIZE (1 << 23)
- #define DATACHUNK (1 << 21)
- #define MAX_CONNECTIONS 256

Functions

- int * setup_tcp_server (char *p_address, int port)
  SETUP TCP SERVER.
- int start_tcp_server (int *p_index)
  START TCP SERVER.
- int end_tcp_server (int *p_index)
  END TCP SERVER.

Variables

- struct s_send_tcp_server g_send_tcp_server [MAX_CONNECTIONS]

2.3.1 Detailed Description

Functions to write raw binary files properly in verilog.

Functions to create multiple TCP servers.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-22-02

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-23-02

2.3.2 Macro Definition Documentation

2.3.2.1 BUFFSIZE

```
#define BUFFSIZE (1 << 23)
```

2.3.2.2 DATACHUNK

```
#define DATACHUNK (1 << 21)
```

2.3.2.3 MAX_CONNECTIONS

```
#define MAX_CONNECTIONS 256
```

2.3.3 Function Documentation

2.3.3.1 end_tcp_server()

```
int end_tcp_server (
            int * p_index )
```

END TCP SERVER.

2.3.3.2 setup_tcp_server()

```
int* setup_tcp_server (
            char * p_address,
            int port )
```

SETUP TCP SERVER.

2.3.3.3 start_tcp_server()

```
int start_tcp_server (
            int * p_index )
```

START TCP SERVER.

2.3.4 Variable Documentation

2.3.4.1 g_send_tcp_server

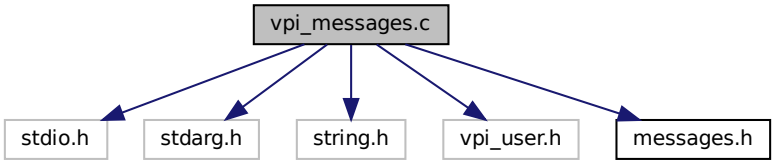
```
struct s_send_tcp_server g_send_tcp_server[MAX_CONNECTIONS]  [extern]
```

2.4 vpi_messages.c File Reference

Functions print messages using the vpi interface.

```
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <vpi_user.h>
#include "messages.h"
```

Include dependency graph for vpi_messages.c:



Functions

- int print_error (const char *format,...)
- int print_info (const char *format,...)

2.4.1 Detailed Description

Functions print messages using the vpi interface.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-03-02

2.4.2 Function Documentation

2.4.2.1 print_error()

```
int print_error (
            const char * format,
             ...  )
```

2.4.2.2 print_info()

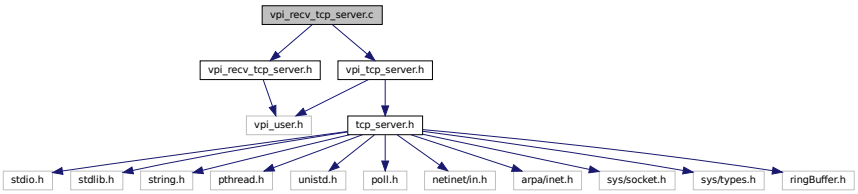
```
int print_info (
            const char * format,
             ...  )
```

2.5 vpi_recv_tcp_server.c File Reference

Functions for TCP server data receive.

```
#include "vpi_tcp_server.h"
#include "vpi_recv_tcp_server.h"
```

Include dependency graph for vpi_recv_tcp_server.c:



Functions

- void ∗ recv_thread (void ∗data)

  RECV TCP SETUP THREAD TO FILL RINGBUFFER.
- PLI_INT32 recv_tcp_server_end_sim_cb (p_cb_data data)

  RECEIVE TCP SERVER DATA END SIM CALLBACK.
- PLI_INT32 recv_tcp_server_start_sim_cb (p_cb_data data)

  RECEIVE TCP SERVER DATA START SIM CALLBACK.
- PLI_INT32 recv_tcp_server_compiletf (PLI_BYTE8 ∗user_data)

  Compile time call, check the arguments for validity.
- PLI_INT32 recv_tcp_server_calltf (PLI_BYTE8 ∗user_data)

  recv_tcp_server_calltf is a callback for the recv_tcp_server function.

2.5.1 Detailed Description

Functions for TCP server data receive.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-02-22

2.5.2 Function Documentation

2.5.2.1 recv_tcp_server_calltf()

```
PLI_INT32 recv_tcp_server_calltf (
             PLI_BYTE8 * user_data )
```

recv_tcp_server_calltf is a callback for the recv_tcp_server function.

2.5.2.2 recv_tcp_server_compiletf()

```
PLI_INT32 recv_tcp_server_compiletf (
             PLI_BYTE8 * user_data )
```

Compile time call, check the arguments for validity.

RECEIVE TCP SERVER DATA COMPILE SETUP.

2.5.2.3 recv_tcp_server_end_sim_cb()

```
PLI_INT32 recv_tcp_server_end_sim_cb (
             p_cb_data data )
```

RECEIVE TCP SERVER DATA END SIM CALLBACK.

2.5.2.4 recv_tcp_server_start_sim_cb()

```
PLI_INT32 recv_tcp_server_start_sim_cb (
             p_cb_data data )
```

RECEIVE TCP SERVER DATA START SIM CALLBACK.

2.5.2.5 recv_thread()

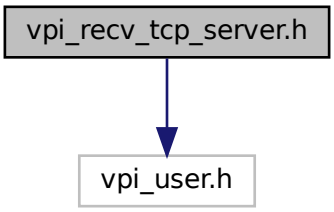
```
void* recv_thread (
             void * data )
```

RECV TCP SETUP THREAD TO FILL RINGBUFFER.

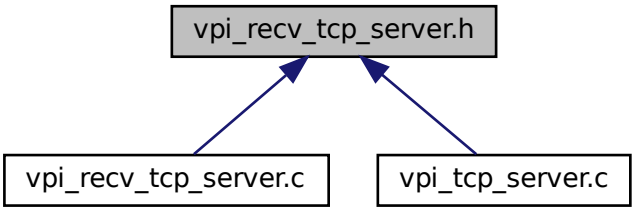
2.6 vpi_recv_tcp_server.h File Reference

Functions for TCP server data receive.

`#include <vpi_user.h>`
Include dependency graph for vpi_recv_tcp_server.h:



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



Functions

- PLI_INT32 recv_tcp_server_compiletf (PLI_BYTE8 *user_data)
  RECEIVE TCP SERVER DATA COMPILE SETUP.
- PLI_INT32 recv_tcp_server_calltf (PLI_BYTE8 *user_data)
  recv_tcp_server_calltf is a callback for the recv_tcp_server function.

2.6.1 Detailed Description

Functions for TCP server data receive.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-02-22

$recv_tcp_server takes 2 arguments. First a port matching setup_tcp_server, and second a vector of bytes for placing data.

2.6.2 Function Documentation

2.6.2.1 recv_tcp_server_calltf()

```
PLI_INT32 recv_tcp_server_calltf (
             PLI_BYTE8 * user_data )
```

recv_tcp_server_calltf is a callback for the recv_tcp_server function.

2.6.2.2 recv_tcp_server_compiletf()

```
PLI_INT32 recv_tcp_server_compiletf (
             PLI_BYTE8 * user_data )
```

RECEIVE TCP SERVER DATA COMPILE SETUP.

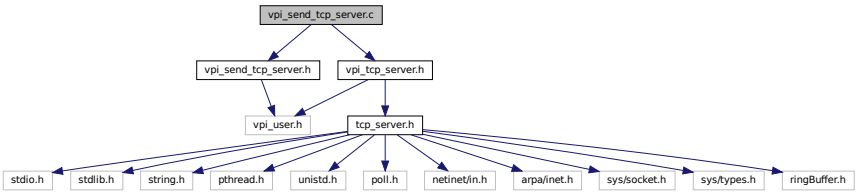
RECEIVE TCP SERVER DATA COMPILE SETUP.

2.7 vpi_send_tcp_server.c File Reference

Functions for TCP server data send.

```
#include "vpi_tcp_server.h"
#include "vpi_send_tcp_server.h"
```

Include dependency graph for vpi_send_tcp_server.c:



Functions

- void ∗ send_thread (void ∗data)

  SEND TCP SERVER THREAD TO EMPTY RINGBUFFER.
- PLI_INT32 send_tcp_server_end_sim_cb (p_cb_data data)

  SEND TCP SERVER DATA END SIM CALLBACK.
- PLI_INT32 send_tcp_server_start_sim_cb (p_cb_data data)

  SEND TCP SERVER DATA START SIM CALLBACK.
- PLI_INT32 send_tcp_server_compiletf (PLI_BYTE8 ∗user_data)

  Compile time call, check the arguments for validity.
- PLI_INT32 send_tcp_server_calltf (PLI_BYTE8 ∗user_data)

  Called by the simulator, each time it is requested.

2.7.1 Detailed Description

Functions for TCP server data send.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-23-02

2.7.2 Function Documentation

2.7.2.1 send_tcp_server_calltf()

```
PLI_INT32 send_tcp_server_calltf (
            PLI_BYTE8 * user_data )
```

Called by the simulator, each time it is requested.

2.7.2.2 send_tcp_server_compiletf()

```
PLI_INT32 send_tcp_server_compiletf (
            PLI_BYTE8 * user_data )
```

Compile time call, check the arguments for validity.

SEND TCP SERVER DATA COMPILE SETUP.

2.7.2.3 send_tcp_server_end_sim_cb()

```
PLI_INT32 send_tcp_server_end_sim_cb (
            p_cb_data data )
```

SEND TCP SERVER DATA END SIM CALLBACK.

2.7.2.4 send_tcp_server_start_sim_cb()

```
PLI_INT32 send_tcp_server_start_sim_cb (
            p_cb_data data )
```

SEND TCP SERVER DATA START SIM CALLBACK.

2.7.2.5 send_thread()

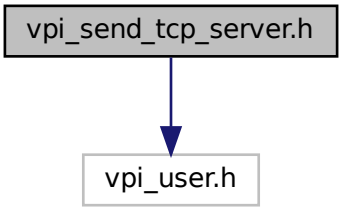
```
void* send_thread (
            void * data )
```

SEND TCP SERVER THREAD TO EMPTY RINGBUFFER.

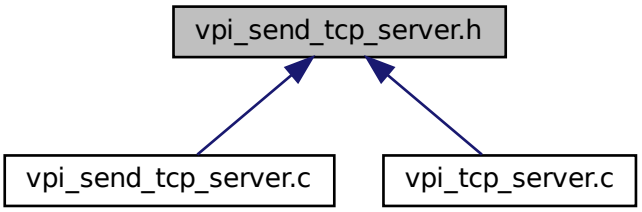
2.8 vpi_send_tcp_server.h File Reference

Function to send data over a tcp server.

`#include <vpi_user.h>`
Include dependency graph for vpi_send_tcp_server.h:



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



Functions

- PLI_INT32 send_tcp_server_compiletf (PLI_BYTE8 *user_data)
  SEND TCP SERVER DATA COMPILE SETUP.
- PLI_INT32 send_tcp_server_calltf (PLI_BYTE8 *user_data)
  Called by the simulator, each time it is requested.

2.8.1 Detailed Description

Function to send data over a tcp server.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-24-2

2.8.2 Function Documentation

2.8.2.1 send_tcp_server_calltf()

```
PLI_INT32 send_tcp_server_calltf (
            PLI_BYTE8 * user_data )
```

Called by the simulator, each time it is requested.

2.8.2.2 send_tcp_server_compiletf()

```
PLI_INT32 send_tcp_server_compiletf (
            PLI_BYTE8 * user_data )
```

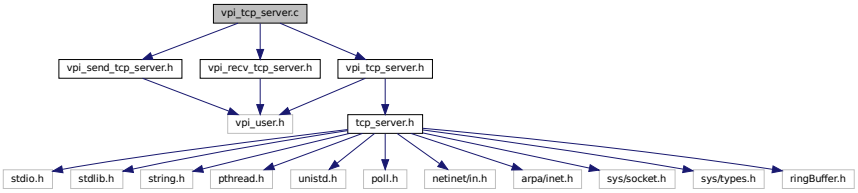
SEND TCP SERVER DATA COMPILE SETUP.

SEND TCP SERVER DATA COMPILE SETUP.

2.9 vpi_tcp_server.c File Reference

```
#include "vpi_send_tcp_server.h"
#include "vpi_recv_tcp_server.h"
#include "vpi_tcp_server.h"
```

Include dependency graph for vpi_tcp_server.c:



Functions

- PLI_INT32 setup_tcp_server_start_sim_cb (p_cb_data data)

  SETUP TCP SERVER DATA START SIM CALLBACK.
- PLI_INT32 setup_tcp_server_end_sim_cb (p_cb_data data)

  SETUP TCP SERVER END SIM CALLBACK.
- PLI_INT32 tcp_server_sizetf (PLI_BYTE8 *user_data)

  Returns the size, in bits, of the function return type.
- PLI_INT32 setup_tcp_server_compiletf (PLI_BYTE8 *user_data)

  Compile time call, check the arguments for validity.
- PLI_INT32 setup_tcp_server_calltf (PLI_BYTE8 *user_data)

  setup_tcp_server_calltf is the callback for the setup_tcp_server function.
- void recv_tcp_server_reg_systf (void)

  Setup recv_tcp_server function.
- void send_tcp_server_reg_systf (void)

  Setup send_tcp_server function.
- void setup_tcp_server_reg_systf (void)

  Setup setup_tcp_server function.

Variables

- void(* vlog_startup_routines [ ])(void)

  register the new file functions

2.9.1 Function Documentation

2.9.1.1 recv_tcp_server_reg_systf()

```
void recv_tcp_server_reg_systf (
            void  )
```

Setup recv_tcp_server function.

2.9.1.2 send_tcp_server_reg_systf()

```
void send_tcp_server_reg_systf (
            void  )
```

Setup send_tcp_server function.

2.9.1.3 setup_tcp_server_calltf()

```
PLI_INT32 setup_tcp_server_calltf (
             PLI_BYTE8 * user_data )
```

setup_tcp_server_calltf is the callback for the setup_tcp_server function.

2.9.1.4 setup_tcp_server_compiletf()

```
PLI_INT32 setup_tcp_server_compiletf (
             PLI_BYTE8 * user_data )
```

Compile time call, check the arguments for validity.

2.9.1.5 setup_tcp_server_end_sim_cb()

```
PLI_INT32 setup_tcp_server_end_sim_cb (
             p_cb_data data )
```

SETUP TCP SERVER END SIM CALLBACK.

2.9.1.6 setup_tcp_server_reg_systf()

```
void setup_tcp_server_reg_systf (
             void  )
```

Setup setup_tcp_server function.

2.9.1.7 setup_tcp_server_start_sim_cb()

```
PLI_INT32 setup_tcp_server_start_sim_cb (
             p_cb_data data )
```

SETUP TCP SERVER DATA START SIM CALLBACK.

2.9.1.8 tcp_server_sizetf()

```
PLI_INT32 tcp_server_sizetf (
             PLI_BYTE8 * user_data )
```

Returns the size, in bits, of the function return type.

2.9.2 Variable Documentation

2.9.2.1 vlog_startup_routines

```
void(* vlog_startup_routines[])(void) (
             void  )
```

Initial value:

```
= {
  recv_tcp_server_reg_systf,
  send_tcp_server_reg_systf,
  setup_tcp_server_reg_systf,
  0
}
```

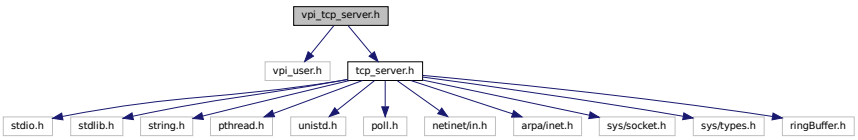
register the new file functions

2.10 vpi_tcp_server.h File Reference

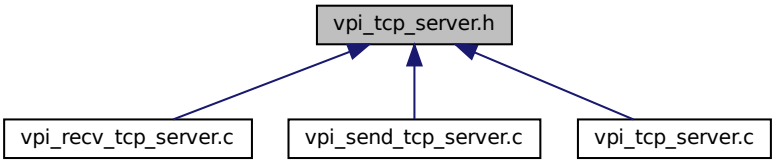
Functions to write raw binary files properly in verilog.

```
#include <vpi_user.h>
#include "tcp_server.h"
```

Include dependency graph for vpi_tcp_server.h:



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



Data Structures

- struct s_vpi_data

Macros

- #define RECV_NAME "$recv_tcp_server"
- #define SEND_NAME "$send_tcp_server"
- #define SETUP_NAME "$setup_tcp_server"

2.10.1 Detailed Description

Functions to write raw binary files properly in verilog.

Functions to create multiple TCP servers.

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-22-02

Author

Jay Convertino(johnathan.convertino.1@us.af.mil)

Date

2024-23-02

2.10.2 Macro Definition Documentation

2.10.2.1 RECV_NAME

```
#define RECV_NAME "$recv_tcp_server"
```

2.10.2.2 SEND_NAME

```
#define SEND_NAME "$send_tcp_server"
```

2.10.2.3 SETUP_NAME

```
#define SETUP_NAME "$setup_tcp_server"
```

Index

arg1_handle
    s_vpi_data, 4
arg2_handle
    s_vpi_data, 4
array_byte_size
    s_vpi_data, 4

BUFFSIZE
    tcp_server.h, 11

connection_keep_alive
    tcp_server.c, 7
connection_thread
    s_send_tcp_server, 2

DATACHUNK
    tcp_server.h, 11

end_tcp_server
    tcp_server.c, 7
    tcp_server.h, 11
error
    s_vpi_data, 4

g_num_of_connections
    tcp_server.c, 8
g_send_tcp_server
    tcp_server.c, 8
    tcp_server.h, 12

kill_thread
    s_send_tcp_server, 2

MAX_CONNECTIONS
    tcp_server.h, 11
messages.h, 5
    print_error, 6
    print_info, 6

num_ab_val_pairs
    s_vpi_data, 4

p_address
    s_send_tcp_server, 2
p_data
    s_process_data, 1
p_ringbuffer
    s_process_data, 1
p_socket_info
    s_send_tcp_server, 3
poll_connection
    s_send_tcp_server, 3
port
    s_send_tcp_server, 3
print_error
    messages.h, 6
    vpi_messages.c, 13
print_info
    messages.h, 6
    vpi_messages.c, 13

RECV_NAME
    vpi_tcp_server.h, 26
recv_process_data
    s_send_tcp_server, 3
recv_tcp_server_calltf
    vpi_recv_tcp_server.c, 15
    vpi_recv_tcp_server.h, 17
recv_tcp_server_compiletf
    vpi_recv_tcp_server.c, 15
    vpi_recv_tcp_server.h, 17
recv_tcp_server_end_sim_cb
    vpi_recv_tcp_server.c, 15
recv_tcp_server_reg_systf
    vpi_tcp_server.c, 22
recv_tcp_server_start_sim_cb
    vpi_recv_tcp_server.c, 15
recv_thread
    vpi_recv_tcp_server.c, 15

s_process_data, 1
    p_data, 1
    p_ringbuffer, 1
    thread, 1
s_send_tcp_server, 2
    connection_thread, 2
    kill_thread, 2
    p_address, 2
    p_socket_info, 3
    poll_connection, 3
    port, 3
    recv_process_data, 3
    send_process_data, 3
s_vpi_data, 3
    arg1_handle, 4
    arg2_handle, 4
    array_byte_size, 4
    error, 4
    num_ab_val_pairs, 4
    systf_handle, 4
SEND_NAME

vpi_tcp_server.h, 26
send_process_data
s_send_tcp_server, 3
send_tcp_server_calltf
vpi_send_tcp_server.c, 19
vpi_send_tcp_server.h, 21
send_tcp_server_compiletf
vpi_send_tcp_server.c, 19
vpi_send_tcp_server.h, 21
send_tcp_server_end_sim_cb
vpi_send_tcp_server.c, 19
send_tcp_server_reg_systf
vpi_tcp_server.c, 22
send_tcp_server_start_sim_cb
vpi_send_tcp_server.c, 19
send_thread
vpi_send_tcp_server.c, 19
SETUP_NAME
vpi_tcp_server.h, 26
setup_tcp_server
tcp_server.c, 7
tcp_server.h, 11
setup_tcp_server_calltf
vpi_tcp_server.c, 22
setup_tcp_server_compiletf
vpi_tcp_server.c, 23
setup_tcp_server_end_sim_cb
vpi_tcp_server.c, 23
setup_tcp_server_reg_systf
vpi_tcp_server.c, 23
setup_tcp_server_start_sim_cb
vpi_tcp_server.c, 23
start_tcp_server
tcp_server.c, 7
tcp_server.h, 11
systf_handle
s_vpi_data, 4

tcp_server.c, 6
connection_keep_alive, 7
end_tcp_server, 7
g_num_of_connections, 8
g_send_tcp_server, 8
setup_tcp_server, 7
start_tcp_server, 7
tcp_server.h, 8
BUFFSIZE, 11
DATACHUNK, 11
end_tcp_server, 11
g_send_tcp_server, 12
MAX_CONNECTIONS, 11
setup_tcp_server, 11
start_tcp_server, 11
tcp_server_sizetf
vpi_tcp_server.c, 23
thread
s_process_data, 1

vlog_startup_routines
vpi_tcp_server.c, 24
vpi_messages.c, 12
print_error, 13
print_info, 13
vpi_recv_tcp_server.c, 14
recv_tcp_server_calltf, 15
recv_tcp_server_compiletf, 15
recv_tcp_server_end_sim_cb, 15
recv_tcp_server_start_sim_cb, 15
recv_thread, 15
vpi_recv_tcp_server.h, 16
recv_tcp_server_calltf, 17
recv_tcp_server_compiletf, 17
vpi_send_tcp_server.c, 18
send_tcp_server_calltf, 19
send_tcp_server_compiletf, 19
send_tcp_server_end_sim_cb, 19
send_tcp_server_start_sim_cb, 19
send_thread, 19
vpi_send_tcp_server.h, 20
send_tcp_server_calltf, 21
send_tcp_server_compiletf, 21
vpi_tcp_server.c, 21
recv_tcp_server_reg_systf, 22
send_tcp_server_reg_systf, 22
setup_tcp_server_calltf, 22
setup_tcp_server_compiletf, 23
setup_tcp_server_end_sim_cb, 23
setup_tcp_server_reg_systf, 23
setup_tcp_server_start_sim_cb, 23
tcp_server_sizetf, 23
vlog_startup_routines, 24
vpi_tcp_server.h, 24
RECV_NAME, 26
SEND_NAME, 26
SETUP_NAME, 26