

openairinterface5g

Техническая документация

Сгенерировано: 2025-06-16 10:19

Содержание

• Release Notes & Latest Changes	2
• C/C++ project: openair1/PHY/CODING/3gpplte.c File Reference	4
• C/C++ project: openair1/PHY/CODING/3gpplte_sse.c File Reference	12
• C/C++ project: openair1/PHY/CODING/3gpplte_turbo_decoder.c File Reference	20
• C/C++ project: openair1/PHY/CODING/3gpplte_turbo_decoder_avx2_16bit.c File Reference	43
• C/C++ project: openair1/PHY/CODING/3gpplte_turbo_decoder_sse.c File Reference	45
• C/C++ project: openair1/PHY/CODING/3gpplte_turbo_decoder_sse_16bit.c File Reference	55
• C/C++ project: openair1/PHY/CODING/3gpplte_turbo_decoder_sse_8bit.c File Reference	66
• C/C++ project: radio/ETHERNET/benetel/4g/benetel.c File Reference	77
• C/C++ project: radio/ETHERNET/benetel/4g/dpdk_driver.c File Reference	90
• C/C++ project: radio/ETHERNET/benetel/4g/low.c File Reference	112
• C/C++ project: radio/ETHERNET/benetel/4g/low.h File Reference	116
• C/C++ project: radio/ETHERNET/benetel/4g/low.h Source File	118
• C/C++ project: radio/ETHERNET/benetel/4g/low_dpdk.c File Reference	119
• C/C++ project: radio/ETHERNET/benetel/4g/shared_buffers.c File Reference	122
• C/C++ project: radio/ETHERNET/benetel/4g/shared_buffers.h File Reference	128
• C/C++ project: radio/ETHERNET/benetel/4g/shared_buffers.h Source File	132
• C/C++ project: doc/5Gnas.md File Reference	134

- C/C++ project: radio/ETHERNET/benetel/5g/benetel.c File
Reference 135
- C/C++ project: radio/ETHERNET/benetel/5g/dpdk_driver.c File
Reference 145
- C/C++ project: radio/ETHERNET/benetel/5g/low.c File
Reference 167

Release Notes & Latest Changes

2025.w24 — 2025-06-13

Integration 2025 week 24 * !3354 Preparation Work for N2 Handover * !3383 Add configurable values of NR RLC and NR PDCP to the configuration file * !3468 Resolve "SSB frequency at gnb.sa.band78.fr1.24PRB.usrp210.conf is invalid" * !3466 YAML related updates * !3460 SRS configuration * !3474 Fix AMF selection fallback by PLMN ID when no UE identity is present or matching * !3473 Fix various bugs and inconsistencies in config read, SCTP, ITTI, GTP * !3169 NR RU improvements for analog beamforming * !3456 CI: update config for AW2S pipeline * !3369 Add Security Mode Reject lib/unit test and adopt in stack * !3457 Fix NR reestablishment * !3412 [E2 agent] E2AP README update and OAI-FlexRIC CI pipeline improvements

2025.w23 — 2025-06-04

Integration 2025 week 23 * !3302 Enhance UE identity management in Initial UE Message and other NGAP improvements * !3400 T bugfix: check input data a bit better * !3459 Improvements in NR band tables according to Rel.17 * !3465 Fix checking that amf_ip_address section is not set. * !3463 Move RRC radio parameters file to DU * !3389 Relax NR_UE_CAPABILITY_SLOT_RX_TO_TX asserts * !3417 Imscope updates * !3443 Fix data race in NR UE MSG3 scheduling * !3467 remove dead globals

2025.w22 — 2025-05-28

Integration 2025 week 22 * !3415 several fixes and cleanup for nrLDPC_coding_t2 * !3448 T: macpdu2wireshark: dump to file instead of sending UDP packets * !3449 bugfix: pass correct buffer * !3453 CI: revert modification of SSB per RACH occasion in SC-FDMA test * !3440 Update NAS documentation * !3451 fix the number of preambles per SSB at UE in case PRACH is configured with groupB * !3441 Replace hashtable with epoll_event_t in rfsimulator * !3277 Add CI test to force RRC IDLE and new connection setup with 5G-S-TMSI * !3454 Keep old MAC stats after re-establishment * !3450 Make number of UL/DL actors in NR UE fully configurable * !3458 (doc): update README with build icons for dedicated arch and os

2025.w20 — 2025-05-20

Integration 2025 week 20 * !3168 Improvements for LDPC encoding * !3386 E1 Bearer Context Release enc/dec lib and unit test * !3394 Fix Liteon with MTU 1500 and update the FHI docs * !3437 CI: RFsim F1/HO: Use hanging-workaround to avoid blocking of second client * !3384 Add enc/dec library

and unit test for E1 Bearer Context Modification Failure * !3418 Use common function to generate CSI-RS signal * !3379 Add physim tests into ctest framework * !3420 Added intercommunication between namespaces * !3422 Tutorials: updates for NR SA Tutorials * !3436 Beam switching small fixes * !3439 Fix RRC resources periodicity determination according to number of slots per period * !3434 Remove EPC/UE main.py parameters * !3423 Fix PDSCH and PUSCH BWP Start and Size when PXSCH is scheduled with a DCI format x_0 in any type of PDCCH common search space * !3442 Fix UL channel estimates mapping in 2-layer MMSE receiver

2025.w19 — 2025-05-13

Integration 2025 week 19 * !3409 Correctly handle minimum RB condition in MAC * !3421 Fix DMRS for PUCCH format 2 * !3424 limit the number of HARQ processes in case of DCI00 and 10 * !3116 ue txData more contextual, but still global as it is entangled with usrp driver * !3408 Handling 2 search spaces per slot in SIB1 * !3419 PRACH configuration index warning * !3429 CI: Add ` -l` option for the UL iperf test * !3381 Add 5GMM Authentication Failure enc/dec lib and unit test

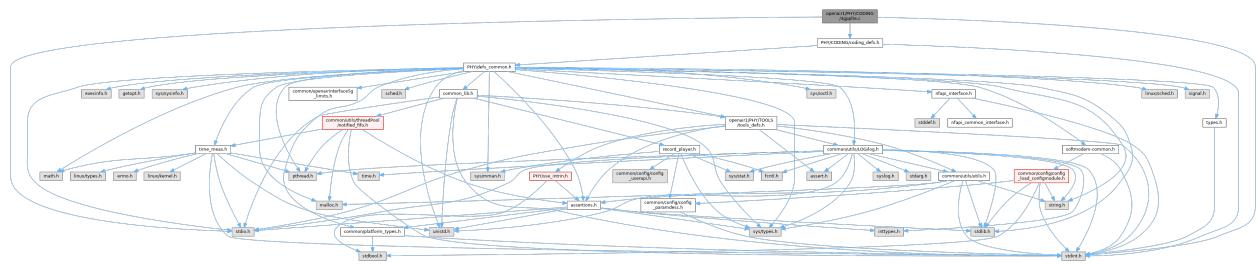
Recent Commits

- `8c0641c` — 2025-06-13 (Robert Schmidt): Merge branch 'integration_2025_w24' into 'develop'
- `736ea53` — 2025-06-13 (luis_pereira87): Only reestablish RLC while processing reconfiguration complete after a RRCCreestablishment and not for every RRCCreconfiguration
- `5b9d770` — 2025-06-13 (Robert Schmidt): Merge remote-tracking branch 'origin/e2-fixes-updates' into integration_2025_w24 (!3412)
- `9513d48` — 2025-06-13 (Robert Schmidt): Merge remote-tracking branch 'origin/fix_nr_reestablishment' into integration_2025_w24 (!3457)
- `dfadocab` — 2025-06-12 (Robert Schmidt): Store spCellConfig during reestablishment for reconfiguration
- `81dbaf0` — 2025-06-12 (Robert Schmidt): Add comment on reestablishment in MAC-RRC DL handler
- `15370e7` — 2025-06-12 (Robert Schmidt): Implement RB suspend at gNB MAC
- `f3802d7` — 2025-06-12 (Robert Schmidt): Merge remote-tracking branch 'origin/nr-ue-nas-sec-mode-reject' into integration_2025_w24 (!3369)
- `690be9c` — 2025-06-12 (Robert Schmidt): Merge remote-tracking branch 'origin/ci-asue-test' into integration_2025_w24 (!3456)
- `27ece7e` — 2025-06-12 (Jaroslava Fiedlerova): CI: update AmariUE configuration file, add RF configuration

3gpplte.c File Reference

```
#include <stdint.h>
#include <stdio.h>
#include "PHY/CODING/coding_defs.h"
```

Include dependency graph for 3gpplte.c:



Functions

```
void threegpplte_interleaver_reset (void)  
uint16_t threegpplte_interleaver (uint16_t f1, uint16_t f2, uint16_t K)  
uint8_t threegpplte_rsc (uint8_t input, uint8_t *state)  
uint8_t threegpplte_rsc_lut (uint8_t input, uint8_t *state)  
void threegpplte_rsc_termination (uint8_t *x, uint8_t *z, uint8_t *state)  
void threegpplte_turbo_encoder (uint8_t *input, uint16_t  
    input_length_bytes, uint8_t *output, uint8_t F)  
short threegpp_interleaver_parameters (uint16_t bytes_per_codeword)  
    attribute ((constructor))
```

Variables

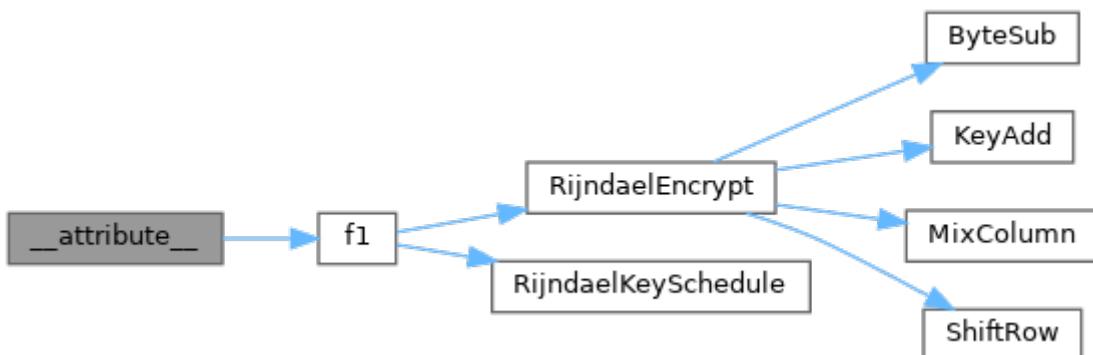
```
uint32_t threegpplte_interleaver_output  
uint32_t threegpplte_interleaver_tmp  
uint8_t output_lut [16]  
uint8_t state_lut [16]  
int turbo_encoder_init = 0  
uint32_t bit_byte_lut [2048]  
const interleaver_TS_36_212_t f1f2 [188]  
t_interleaver_codebook * f1f2mat  
short * il_tb
```

Function Documentation

◆ __attribute__()

```
__attribute__ ( (constructor) )
```

Here is the call graph for this function:

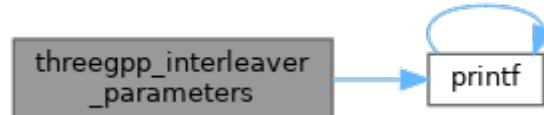


◆ threegpp_interleaver_parameters()

```
short
threegpp_interleaver_parameters ( uint16_t bytes_per_codeword )
```

inline

Here is the call graph for this function:

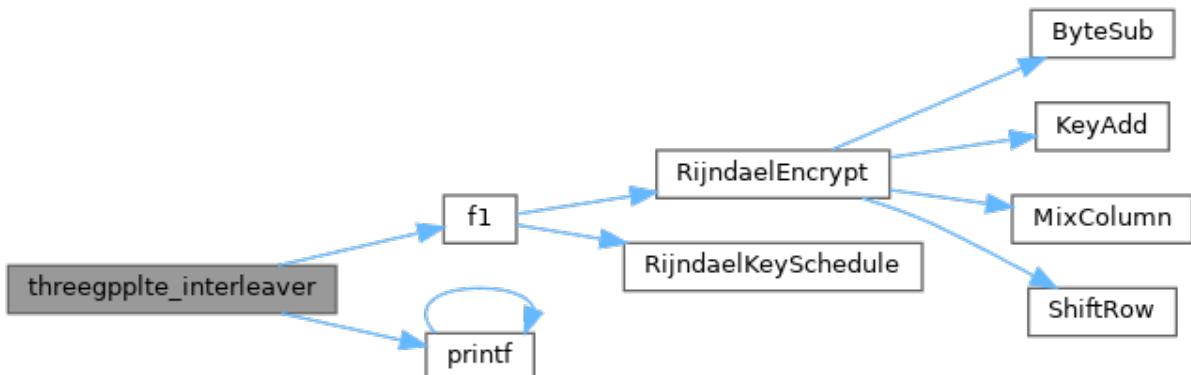


◆ threegpplte_interleaver()

```
uint16_t threegpplte_interleaver ( uint16_t f1,
                                    uint16_t f2,
                                    uint16_t K )
```

inline

Here is the call graph for this function:



Here is the caller graph for this function:



◆ threegpplte_interleaver_reset()

```
void threegpplte_interleaver_reset ( void )
```

inline

Here is the caller graph for this function:

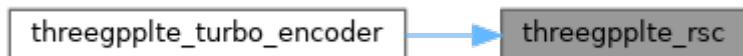


◆ threegpplte_rsc()

```
uint8_t threegpplte_rsc ( uint8_t input,  
                           uint8_t * state )
```

inline

Here is the caller graph for this function:



◆ threegpplte_rsc_lut()

<code>uint8_t threegpplte_rsc_lut</code>	(<code>uint8_t</code>	input,
		<code>uint8_t *</code>	state)

inline

Here is the caller graph for this function:



◆ threegpplte_rsc_termination()

<code>void threegpplte_rsc_termination</code>	(<code>uint8_t *</code>	x,
		<code>uint8_t *</code>	z,
		<code>uint8_t *</code>	state)

inline

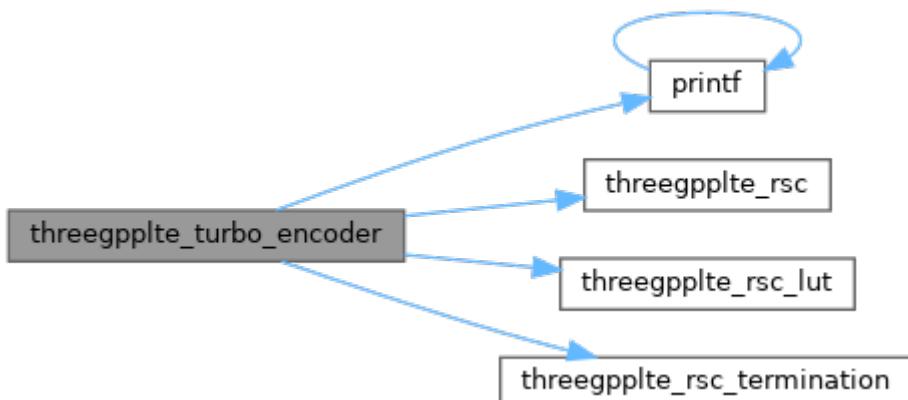
Here is the caller graph for this function:



◆ threegpplte_turbo_encoder()

```
void threegpplte_turbo_encoder ( uint8_t * input,  
                                 uint16_t input_length_bytes,  
                                 uint8_t * output,  
                                 uint8_t F )
```

Here is the call graph for this function:



Variable Documentation

◆ `bit_byte_lut`

```
uint32_t bit_byte_lut[2048]
```

◆ `f1f2`

```
const interleaver_TS_36_212_t f1f2[188]
```

◆ `f1f2mat`

t_interleaver_codebook* f1f2mat

◆ il_tb

short* il_tb

◆ output_lut

uint8_t output_lut[16]

◆ state_lut

uint8_t state_lut[16]

◆ threegpplte_interleaver_output

uint32_t threegpplte_interleaver_output

◆ threegpplte_interleaver_tmp

```
uint32_t threegpplte_interleaver_tmp
```

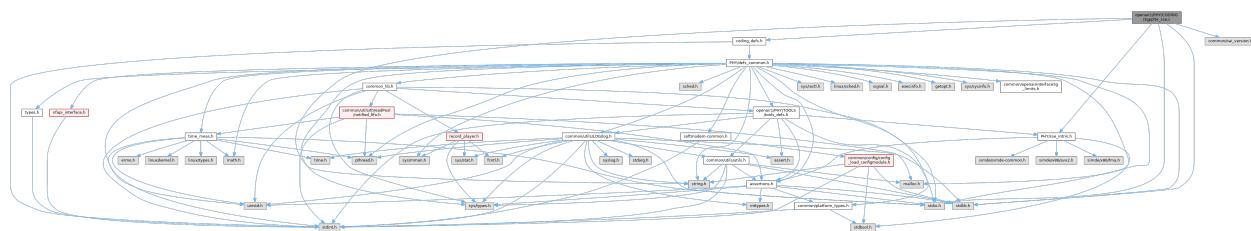
◆ turbo_encoder_init

```
int turbo_encoder_init = 0
```

3gpplte_sse.c File Reference

```
#include "coding_defs.h"  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <stdlib.h>  
  
#include "common/oai_version.h"  
  
#include "PHY/sse_intrin.h"
```

Include dependency graph for 3gpplte_sse.c:



Classes

struct trellis

Macros

```
#define print_bytes(s, x)
#define print_shorts(s, x)
#define print_ints(s, x)
#define print_bytes2(s, x)
```

Functions

```
struct trellis __attribute__ ((aligned(64)))
uint8_t systematic_andp1[24] __attribute__ ((aligned(32)))
static unsigned char threegpplte_rsc (unsigned char input, unsigned char
*state)
static void threegpplte_rsc_termination (unsigned char *x,
unsigned char *z, unsigned char *state)
static void trellis_table_init (void)
char interleave_compact_byte (short *base_interleaver,
unsigned char *input, unsigned char *output, int n)
void threegpplte_turbo_encoder_sse (unsigned char
*input, unsigned short input_length_bytes, unsigned
char *output, unsigned char F)
void init_encoder_sse (void)
int coding_checkbuildver (char
*mainexec_buildversion, char **shlib_buildversion)
```

Variables

```
int exit_state
struct trellis all_trellis [8][256]
int all_trellis_initialized =0
```

Macro Definition Documentation

◆ print_bytes

```
#define print_byte
```

(

S₁

x)

Value:

```
printf("%s %x,%x,%x,%x,%x,%x,%x,%x,%x,%x,%x,%x,%x,%x,%x,%x,%x\n",s,(x)[0],(x)[1],(x)[2],(x)[3],(x)[4],(x)[5],(x)[6],(x)[7],(x)[8],(x)[9],(x)[10],(x)[11],(x)[12],(x)[13],(x)[14],(x)[15])
```

◆ print_bytes2

```
#define print bytes2
```

(

S.

x)

Value:

◆ print_ints

```
#define print_ints
```

(

6

Y)

Value:

```
printf("%s %x %x %x %x\n",s,(x)[0],(x)[1],(x)[2],(x)[3])
```

◆ print_shorts

```
#define print_shorts ( s,  
                      x )
```

Value:

```
printf("%s %x,%x,%x,%x,%x,%x,%x,%x\n",s,(x)[0],(x)[1],(x)[2],(x)[3],(x)[4],(x)[5],(x)[6],  
(x)[7])
```

Function Documentation

◆ _attribute_() [1/2]

```
uint8_t systematic_andp1[24] __attribute__::__attribute__ ( (aligned(32)) )
```

◆ _attribute_() [2/2]

```
struct trellis __attribute__ ( (aligned(64)) )
```

◆ coding_checkbuildver()

int coding_checkbuildver	(char *	mainexec_buildversion,
	char **	shlib_buildversion))

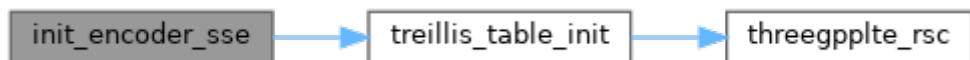
Here is the call graph for this function:



◆ init_encoder_sse()

void init_encoder_sse	(void)
-----------------------	---	------	---

Here is the call graph for this function:



◆ interleave_compact_byte()

char interleave_compact_byte	(short *	base_interleaver,
	unsigned char *	input,	
	unsigned char *	output,	
	int	n)	

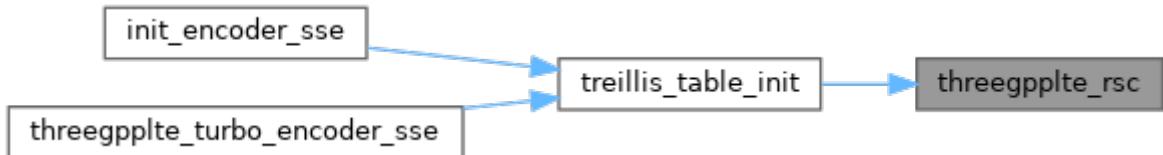
Here is the caller graph for this function:



◆ threegpplte_rsc()

```
static unsigned char threegpplte_rsc ( unsigned char input,  
                                     unsigned char * state )  
    inline static
```

Here is the caller graph for this function:



◆ `threegpplte_rsc_termination()`

```
static void threegpplte_rsc_termination ( unsigned char * x,  
                                         unsigned char * z,  
                                         unsigned char * state )  
    inline static
```

Here is the caller graph for this function:



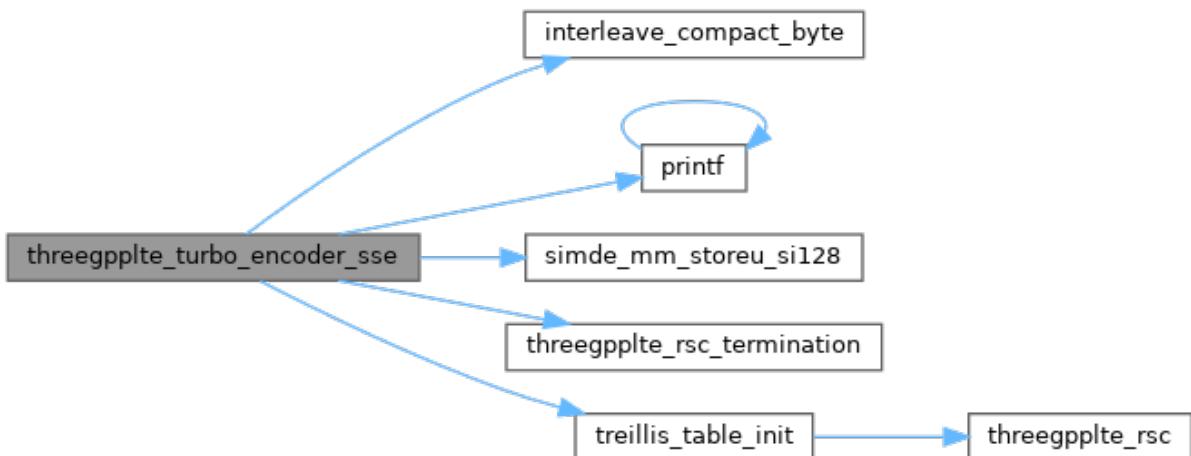
◆ `threegpplte_turbo_encoder_sse()`

```

void threegpplte_turbo_encoder_sse ( unsigned char * input,
                                      unsigned short input_length_bytes,
                                      unsigned char * output,
                                      unsigned char F )

```

Here is the call graph for this function:



◆ trellis_table_init()

```

static void trellis_table_init ( void )

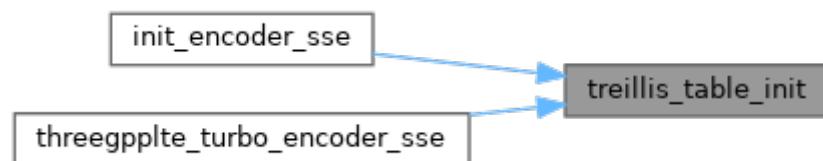
```

static

Here is the call graph for this function:



Here is the caller graph for this function:



Variable Documentation

◆ all_treillis

```
struct treillis all_treillis[8][256]
```

◆ all_treillis_initialized

```
int all_treillis_initialized =0
```

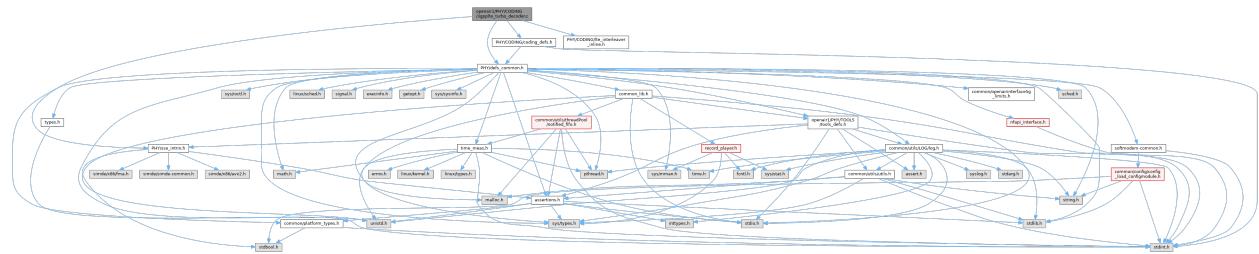
◆ exit_state

```
int exit_state
```

3gpplte_turbo_decoder.c File Reference

```
#include "PHY/defs_common.h"
#include "PHY/CODING/coding_defs.h"
#include "PHY/CODING/lte_interleaver_inline.h"
#include "PHY/sse_intrin.h"
```

Include dependency graph for 3gpplte_turbo_decoder.c:



Macros

```
#define m00 (-m11)
#define m01 (-m10)
#define M0T m00
#define M0B m11
#define M1T m11
#define M1B m00
#define M2T m10
#define M2B m01
#define M3T m01
#define M3B m10
#define M4T m01
#define M4B m10
#define M5T m10
#define M5B m01
#define M6T m11
#define M6B m00
#define M7T m00
#define M7B m11
#define M0T_TERM m00
#define M1T_TERM m11
#define M2T_TERM m10
#define M3T_TERM m01
#define M4T_TERM m01
#define M5T_TERM m10
#define M6T_TERM m11
#define M7T_TERM m00
#define ALPHA_BETA_1m00 alpha[(k-1)*STATES+0] + beta[k*STATES+0]
#define ALPHA_BETA_1m11 alpha[(k-1)*STATES+0] + beta[k*STATES+4]
#define ALPHA_BETA_2m11 alpha[(k-1)*STATES+1] + beta[k*STATES+0]
#define ALPHA_BETA_2m00 alpha[(k-1)*STATES+1] + beta[k*STATES+4]
#define ALPHA_BETA_1m10 alpha[(k-1)*STATES+2] + beta[k*STATES+1]
#define ALPHA_BETA_1m01 alpha[(k-1)*STATES+2] + beta[k*STATES+5]
#define ALPHA_BETA_2m10 alpha[(k-1)*STATES+3] + beta[k*STATES+5]
#define ALPHA_BETA_2m01 alpha[(k-1)*STATES+3] + beta[k*STATES+1]
```

```
#define ALPHA_BETA_3m10 alpha[(k-1)*STATES+4] + beta[k*STATES+6]
#define ALPHA_BETA_3m01 alpha[(k-1)*STATES+4] + beta[k*STATES+2]
#define ALPHA_BETA_4m01 alpha[(k-1)*STATES+5] + beta[k*STATES+6]
#define ALPHA_BETA_4m10 alpha[(k-1)*STATES+5] + beta[k*STATES+2]
#define ALPHA_BETA_3m11 alpha[(k-1)*STATES+6] + beta[k*STATES+3]
#define ALPHA_BETA_3m00 alpha[(k-1)*STATES+6] + beta[k*STATES+7]
#define ALPHA_BETA_4m00 alpha[(k-1)*STATES+7] + beta[k*STATES+3]
#define ALPHA_BETA_4m11 alpha[(k-1)*STATES+7] + beta[k*STATES+7]
#define LLR_MAX 32767
#define LLR_MIN -32768
#define LLRTOT 16
#define MAX 32767
#define FRAME_LENGTH_MAX 6144
#define STATES 8
```

Typedefs

```
typedef char Binary
typedef short llr_t
typedef short channel_t
```

Functions

```
void log_map_s (lIr_t *systematic, channel_t *y_parity, lIr_t *ext,
unsigned short frame_length, unsigned char term_flag, unsigned char
F)
void compute_gamma_s (lIr_t *m11, lIr_t *m10, lIr_t *systematic,
channel_t *y_parity, unsigned short frame_length, unsigned char
term_flag)
void compute_alpha_s (lIr_t *alpha, lIr_t *m11, lIr_t *m10, unsigned
short frame_length, unsigned char F)
void compute_beta_s (lIr_t *beta, lIr_t *m11, lIr_t *m10, lIr_t *alpha,
unsigned short frame_length, unsigned char F)
void compute_ext_s (lIr_t *alpha, lIr_t *beta, lIr_t *m11, lIr_t *m10,
lIr_t *extrinsic, lIr_t *ap, unsigned short frame_length)
int SAT_ADD (int a, int b, int m)
unsigned char phy_threegpplte_turbo_decoder_scalar (lIr_t *y, lIr_t *y2,
unsigned char *decoded_bytes, unsigned char *decoded_bytes2,
unsigned short n, unsigned short f1, unsigned short f2, unsigned char
max_iterations, unsigned char crc_type, unsigned char F,
time_stats_t *init_stats, time_stats_t *alpha_stats, time_stats_t
*beta_stats, time_stats_t *gamma_stats, time_stats_t *ext_stats,
time_stats_t *intl1_stats, time_stats_t *intl2_stats)
```

Variables

```
lIr_t alpha [(FRAME_LENGTH_MAX+3+1) *8]
lIr_t beta [(FRAME_LENGTH_MAX+3+1) *8]
lIr_t m11 [(FRAME_LENGTH_MAX+3)]
lIr_t m10 [(FRAME_LENGTH_MAX+3)]
short systematic0 [6144]
short systematic1 [6144]
short systematic2 [6144]
short yparity1 [6144]
short yparity2 [6144]
```

Macro Definition Documentation

◆ ALPHA_BETA_1m00

```
#define ALPHA_BETA_1m00 alpha[(k-1)*STATES+0] + beta[k*STATES+0]
```

◆ ALPHA_BETA_1m01

```
#define ALPHA_BETA_1m01 alpha[(k-1)*STATES+2] + beta[k*STATES+5]
```

◆ ALPHA_BETA_1m10

```
#define ALPHA_BETA_1m10 alpha[(k-1)*STATES+2] + beta[k*STATES+1]
```

◆ ALPHA_BETA_1m11

```
#define ALPHA_BETA_1m11 alpha[(k-1)*STATES+0] + beta[k*STATES+4]
```

◆ ALPHA_BETA_2m00

```
#define ALPHA_BETA_2m00 alpha[(k-1)*STATES+1] + beta[k*STATES+4]
```

◆ ALPHA_BETA_2m01

```
#define ALPHA_BETA_2m01 alpha[(k-1)*STATES+3] + beta[k*STATES+1]
```

◆ ALPHA_BETA_2m10

```
#define ALPHA_BETA_2m10 alpha[(k-1)*STATES+3] + beta[k*STATES+5]
```

◆ ALPHA_BETA_2m11

```
#define ALPHA_BETA_2m11 alpha[(k-1)*STATES+1] + beta[k*STATES+0]
```

◆ ALPHA_BETA_3m00

```
#define ALPHA_BETA_3m00 alpha[(k-1)*STATES+6] + beta[k*STATES+7]
```

◆ ALPHA_BETA_3m01

```
#define ALPHA_BETA_3m01 alpha[(k-1)*STATES+4] + beta[k*STATES+2]
```

◆ ALPHA_BETA_3m10

```
#define ALPHA_BETA_3m10 alpha[(k-1)*STATES+4] + beta[k*STATES+6]
```

◆ ALPHA_BETA_3m11

```
#define ALPHA_BETA_3m11 alpha[(k-1)*STATES+6] + beta[k*STATES+3]
```

◆ ALPHA_BETA_4m00

```
#define ALPHA_BETA_4m00 alpha[(k-1)*STATES+7] + beta[k*STATES+3]
```

◆ ALPHA_BETA_4m01

```
#define ALPHA_BETA_4m01 alpha[(k-1)*STATES+5] + beta[k*STATES+6]
```

◆ ALPHA_BETA_4m10

```
#define ALPHA_BETA_4m10 alpha[(k-1)*STATES+5] + beta[k*STATES+2]
```

◆ ALPHA_BETA_4m11

```
#define ALPHA_BETA_4m11 alpha[(k-1)*STATES+7] + beta[k*STATES+7]
```

◆ FRAME_LENGTH_MAX

```
#define FRAME_LENGTH_MAX 6144
```

◆ LLR_MAX

```
#define LLR_MAX 32767
```

◆ LLR_MIN

```
#define LLR_MIN -32768
```

◆ LLRTOT

```
#define LLRTOT 16
```

◆ m00

```
#define m00 (-m11)
```

◆ m01

```
#define m01 (-m10)
```

◆ M0B

```
#define M0B m11
```

◆ MOT

```
#define MOT m00
```

◆ MOT_TERM

```
#define MOT_TERM m00
```

◆ M1B

```
#define M1B  m00
```

◆ M1T

```
#define M1T  m11
```

◆ M1T_TERM

```
#define M1T_TERM  m11
```

◆ M2B

```
#define M2B  m01
```

◆ M2T

```
#define M2T  m10
```

◆ M2T_TERM

```
#define M2T_TERM m10
```

◆ M3B

```
#define M3B m10
```

◆ M3T

```
#define M3T m01
```

◆ M3T_TERM

```
#define M3T_TERM m01
```

◆ M4B

```
#define M4B m10
```

◆ M4T

```
#define M4T  m01
```

◆ M4T_TERM

```
#define M4T_TERM  m01
```

◆ M5B

```
#define M5B  m01
```

◆ M5T

```
#define M5T  m10
```

◆ M5T_TERM

```
#define M5T_TERM  m10
```

◆ M6B

```
#define M6B  m00
```

◆ M6T

```
#define M6T  m11
```

◆ M6T_TERM

```
#define M6T_TERM  m11
```

◆ M7B

```
#define M7B  m11
```

◆ M7T

```
#define M7T  m00
```

◆ M7T_TERM

```
#define M7T_TERM m00
```

◆ MAX

```
#define MAX 32767
```

◆ STATES

```
#define STATES 8
```

Typedef Documentation

◆ Binary

```
typedef char Binary
```

◆ channel_t

```
typedef short channel_t
```

◆ llr_t

```
typedef short llr_t
```

Function Documentation

◆ compute_alpha_s()

```
void compute_alpha_s ( llr_t * alpha,  
                      llr_t * m11,  
                      llr_t * m10,  
                      unsigned short frame_length,  
                      unsigned char F )
```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ compute_beta_s()

void compute_beta_s	(<i>llr_t</i> *	beta,
		<i>llr_t</i> *	m11,
		<i>llr_t</i> *	m10,
		<i>llr_t</i> *	alpha,
		unsigned short	frame_length,
		unsigned char	F)

Here is the call graph for this function:



Here is the caller graph for this function:



◆ compute_ext_s()

```

void compute_ext_s (    llr_t * alpha,
                        llr_t * beta,
                        llr_t * m11,
                        llr_t * m10,
                        llr_t * extrinsic,
                        llr_t * ap,
                        unsigned short frame_length )

```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ compute_gamma_s()

```

void compute_gamma_s (    llr_t * m11,
                        llr_t * m10,
                        llr_t * systematic,
                        channel_t * y_parity,
                        unsigned short frame_length,
                        unsigned char term_flag )

```

Here is the caller graph for this function:



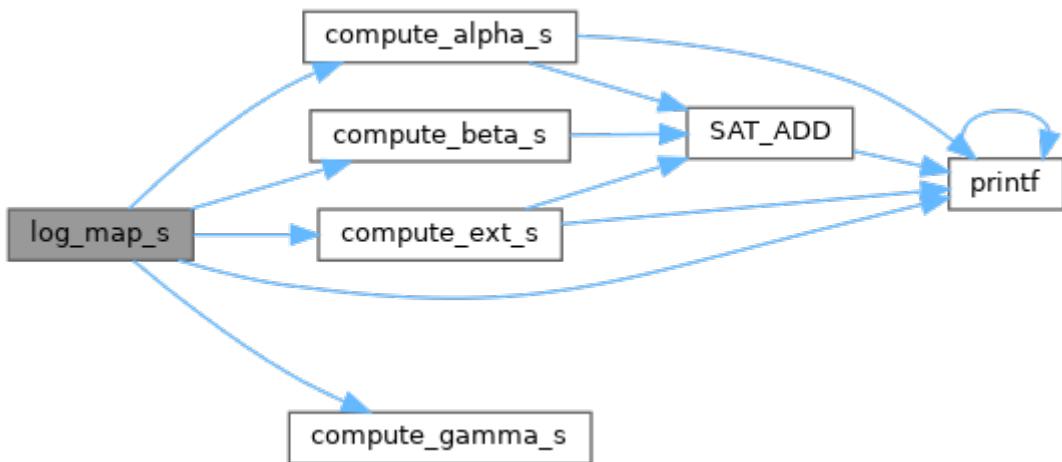
◆ log_map_s()

```

void log_map_s ( IIR_t * systematic,
                  channel_t * y_parity,
                  IIR_t * ext,
                  unsigned short frame_length,
                  unsigned char term_flag,
                  unsigned char F )

```

Here is the call graph for this function:



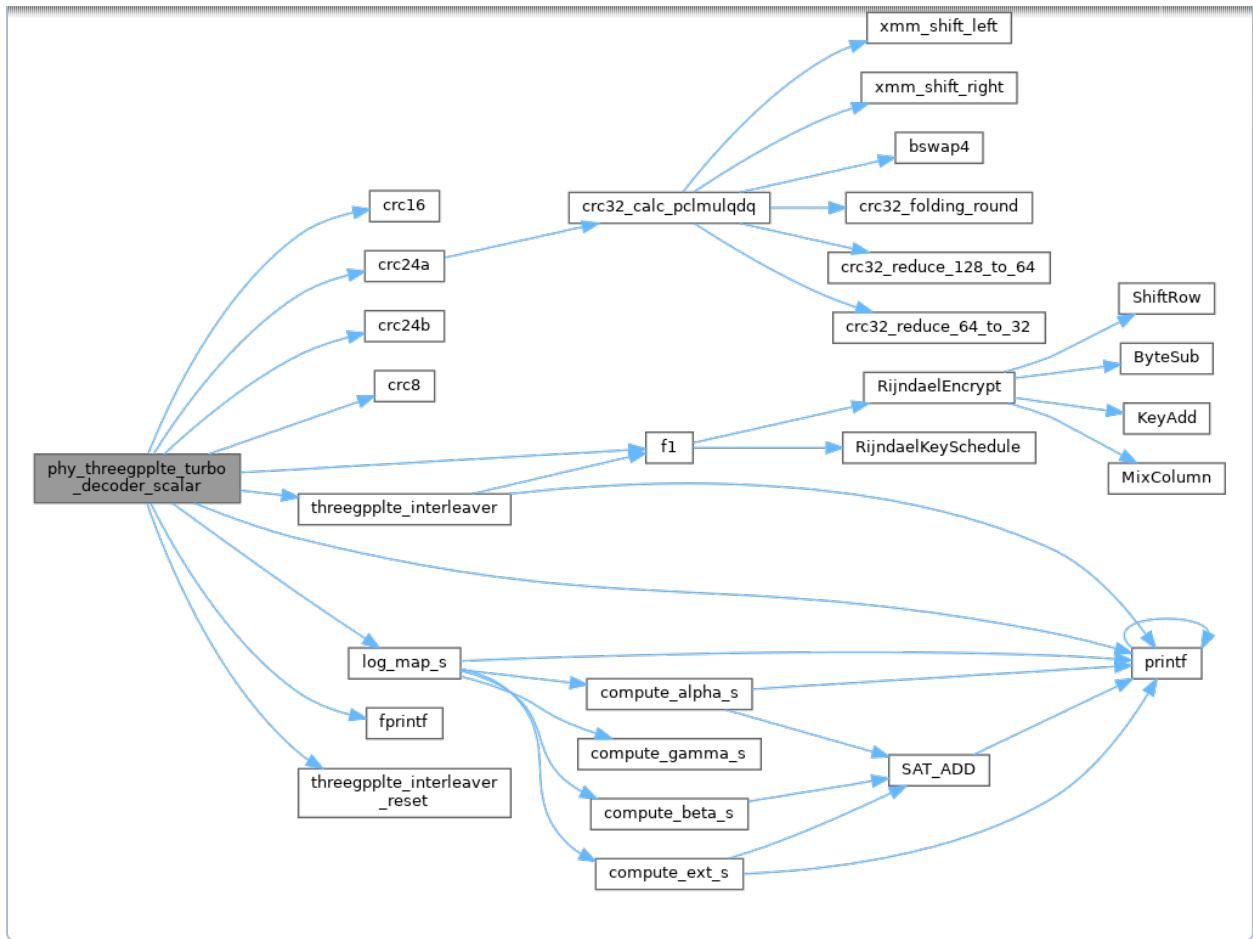
Here is the caller graph for this function:



◆ phy_threegpplte_turbo_decoder_scalar()

<pre> unsigned char phy_threegpplte_turbo_decoder_scalar </pre>	<pre> (IIR_t * y, IIR_t * y2, unsigned char * decoded_bytes, unsigned char * decoded_bytes2, unsigned short n, unsigned short f1, unsigned short f2, unsigned char max_iterations, unsigned char crc_type, unsigned char F, time_stats_t * init_stats, time_stats_t * alpha_stats, time_stats_t * beta_stats, time_stats_t * gamma_stats, time_stats_t * ext_stats, time_stats_t * intl1_stats, time_stats_t * intl2_stats) </pre>
---	--

Here is the call graph for this function:



◆ SAT_ADD()

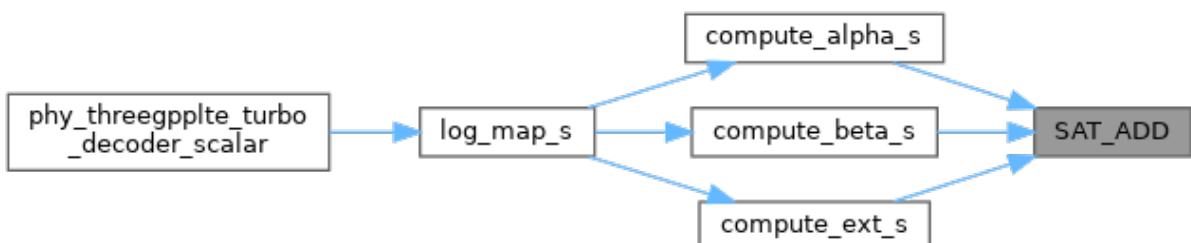
```
int SAT_ADD ( int a,  
              int b,  
              int m )
```

inline

Here is the call graph for this function:



Here is the caller graph for this function:



Variable Documentation

◆ beta

```
d d d beta
```

◆ m10

```
Ilr_t m10[FRAME_LENGTH_MAX+3]
```

◆ m11

```
Ilr_t m11[({FRAME_LENGTH_MAX+3})]
```

◆ systematic0

```
short systematic0[6144]
```

◆ systematic1

```
short systematic1[6144]
```

◆ systematic2

```
short systematic2[6144]
```

◆ yparity1

```
short yparity1[6144]
```

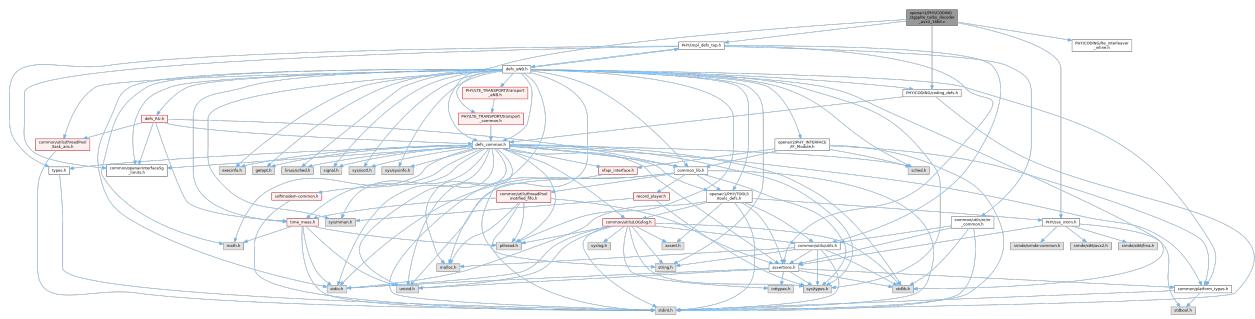
◆ yparity2

short yparity2[6144]

3gpplte_turbo_decoder_avx2_16bit.c File Reference

```
#include "PHY/impl_defs_top.h"
#include "PHY/defs_common.h"
#include "PHY/CODING/coding_defs.h"
#include "PHY/CODING/lte_interleaver_inline.h"
#include "PHY/sse_intrin.h"
```

Include dependency graph for 3gpplte_turbo_decoder_avx2_16bit.c:



Functions

```
unsigned char phy_threegpplte_turbo_decoder16avx2 (int16_t *y, int16_t *y2,
uint8_t *decoded_bytes, uint8_t *decoded_bytes2, uint16_t n,
uint8_t max_iterations, uint8_t crc_type, uint8_t F, time_stats_t
*init_stats, time_stats_t *alpha_stats, time_stats_t *beta_stats,
time_stats_t *gamma_stats, time_stats_t *ext_stats, time_stats_t
*int1l1_stats, time_stats_t *int1l2_stats)
void free_td16avx2 (void)
void init_td16avx2 (void)
```

Function Documentation

◆ **free_td16avx2()**

```
void free_td16avx2
```

```
( void )
```

◆ init_td16avx2()

```
void init_td16avx2
```

```
( void )
```

◆ phy_threegpplte_turbo_decoder16avx2()

```
unsigned char
```

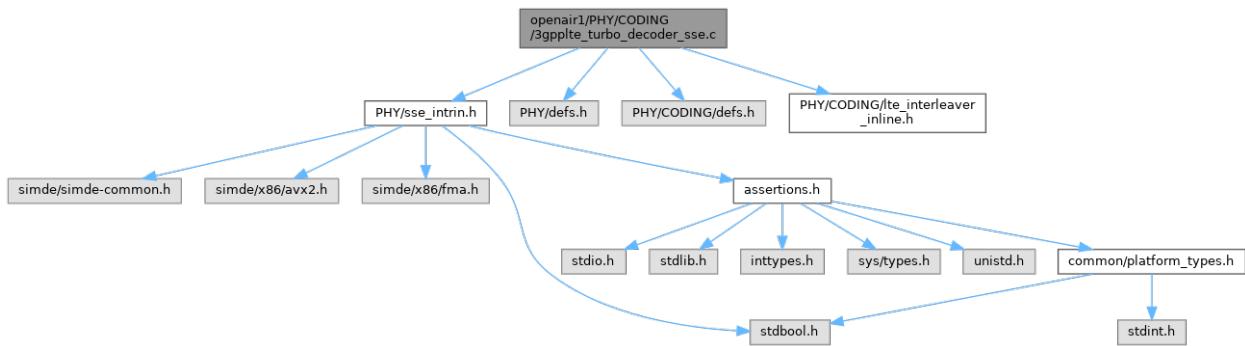
```
phy_threegpplte_turbo_decoder16avx2
```

```
( int16_t * y,  
  int16_t * y2,  
  uint8_t * decoded_bytes,  
  uint8_t * decoded_bytes2,  
  uint16_t n,  
  uint8_t max_iterations,  
  uint8_t crc_type,  
  uint8_t F,  
  time_stats_t * init_stats,  
  time_stats_t * alpha_stats,  
  time_stats_t * beta_stats,  
  time_stats_t * gamma_stats,  
  time_stats_t * ext_stats,  
  time_stats_t * intl1_stats,  
  time_stats_t * intl2_stats )
```

3gpplte_turbo_decoder_sse.c File Reference

```
#include "PHY/sse_intrin.h"
#include "PHY/defs.h"
#include "PHY/CODING/defs.h"
#include "PHY/CODING/lte_interleaver_inline.h"
```

Include dependency graph for 3gpplte_turbo_decoder_sse.c:



Macros

```
#define SHUFFLE16(a, b, c, d, e, f, g, h)
#define MAX 256
#define L 40
```

Typedefs

```
typedef int16_t llr_t
typedef int16_t channel_t
```

Functions

```
void log_map (llr_t *systematic, channel_t *y_parity, llr_t *m11, llr_t
*m10, llr_t *alpha, llr_t *beta, llr_t *ext, unsigned short
frame_length, unsigned char term_flag, unsigned char F, int
offset8_flag, time_stats_t *alpha_stats, time_stats_t *beta_stats,
time_stats_t *gamma_stats, time_stats_t *ext_stats)

void compute_gamma (llr_t *m11, llr_t *m10, llr_t *systematic,
channel_t *y_parity, unsigned short frame_length, unsigned char
term_flag)

void compute_alpha (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10,
unsigned short frame_length, unsigned char F)

void compute_beta (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10,
unsigned short frame_length, unsigned char F, int offset8_flag)

void compute_ext (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10, llr_t
*extrinsic, llr_t *ap, unsigned short frame_length)

void print_bytes (char *s, simde_m128i *x)

void free_td ()

void init_td ()

unsigned char phy_threegpplte_turbo_decoder (short *y, unsigned char
*decoded_bytes, unsigned short n, unsigned char max_iterations,
unsigned char crc_type, unsigned char F, time_stats_t *init_stats,
time_stats_t *alpha_stats, time_stats_t *beta_stats, time_stats_t
*gamma_stats, time_stats_t *ext_stats, time_stats_t *intl1_stats,
time_stats_t *intl2_stats)
```

Variables

```
int * pi2tab [188]

int * pi5tab [188]

int * pi4tab [188]

int * pi6tab [188]
```

Macro Definition Documentation



```
#define L 40
```

◆ MAX

```
#define MAX 256
```

◆ SHUFFLE16

```
#define SHUFFLE16
```

(
	a,
	b,
	c,
	d,
	e,
	f,
	g,
	h)

Value:

```
simde_mm_set_epi8(h == -1 ? -1 : h * 2 + 1, \
    h == -1 ? -1 : h * 2, \
    g == -1 ? -1 : g * 2 + 1, \
    g == -1 ? -1 : g * 2, \
    f == -1 ? -1 : f * 2 + 1, \
    f == -1 ? -1 : f * 2, \
    e == -1 ? -1 : e * 2 + 1, \
    e == -1 ? -1 : e * 2, \
    d == -1 ? -1 : d * 2 + 1, \
    d == -1 ? -1 : d * 2, \
    c == -1 ? -1 : c * 2 + 1, \
    c == -1 ? -1 : c * 2, \
    b == -1 ? -1 : b * 2 + 1, \
    b == -1 ? -1 : b * 2, \
    a == -1 ? -1 : a * 2 + 1, \
    a == -1 ? -1 : a * 2);
```

Typedef Documentation

◆ channel_t

```
typedef int16_t channel_t
```

◆ llr_t

```
typedef int16_t llr_t
```

Function Documentation

◆ compute_alpha()

void compute_alpha	(llr_t *	alpha,
		llr_t *	beta,
		llr_t *	m11,
		llr_t *	m10,
		unsigned short	frame_length,
		unsigned char	F)

Here is the caller graph for this function:



◆ compute_beta()

```

void compute_beta ( IIR_t * alpha,
                   IIR_t * beta,
                   IIR_t * m11,
                   IIR_t * m10,
                   unsigned short frame_length,
                   unsigned char F,
                   int offset8_flag )

```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ compute_ext()

```

void compute_ext ( IIR_t * alpha,
                   IIR_t * beta,
                   IIR_t * m11,
                   IIR_t * m10,
                   IIR_t * extrinsic,
                   IIR_t * ap,
                   unsigned short frame_length )

```

Here is the caller graph for this function:



◆ compute_gamma()

```
void compute_gamma ( llr_t * m11,  
                     llr_t * m10,  
                     llr_t * systematic,  
                     channel_t * y_parity,  
                     unsigned short frame_length,  
                     unsigned char term_flag )
```

Here is the caller graph for this function:



◆ free_td()

```
void free_td ( )
```

◆ init_td()

```
void init_td ( )
```

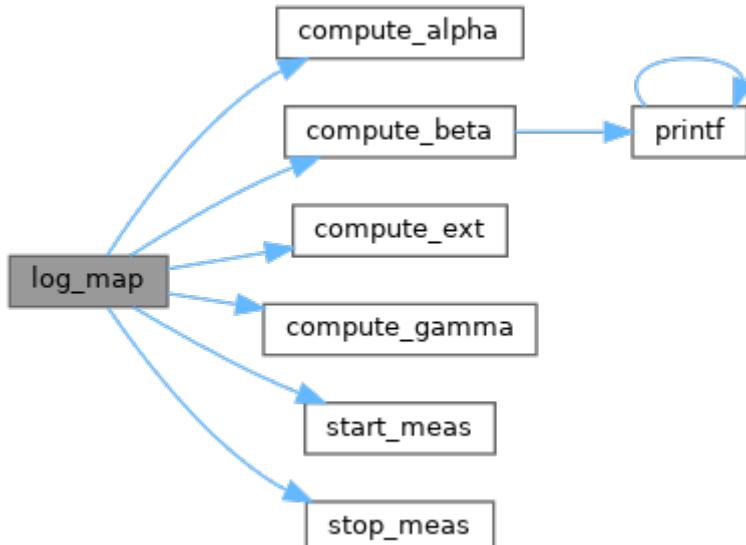
◆ log_map()

```

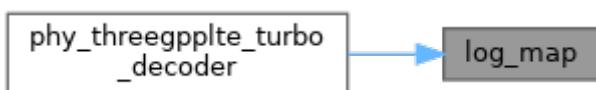
void log_map( lir_t *systematic,
              channel_t *y_parity,
              lir_t *m11,
              lir_t *m10,
              lir_t *alpha,
              lir_t *beta,
              lir_t *ext,
              unsigned short frame_length,
              unsigned char term_flag,
              F,
              int offset8_flag,
              time_stats_t *alpha_stats,
              time_stats_t *beta_stats,
              time_stats_t *gamma_stats,
              time_stats_t *ext_stats )

```

Here is the call graph for this function:



Here is the caller graph for this function:



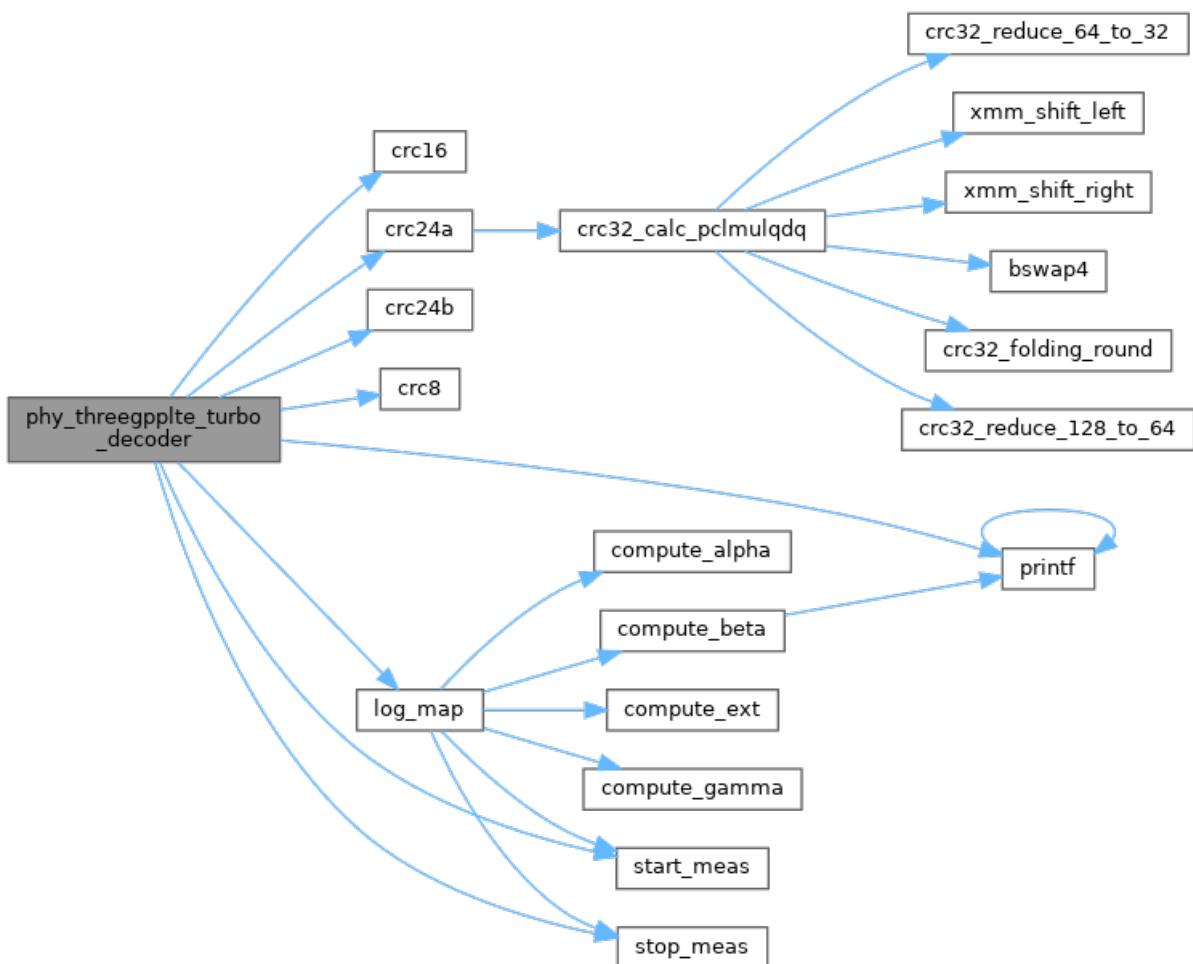
◆ `phy_threegpplte_turbo_decoder()`

```

unsigned char phy_threegpplte_turbo_decoder( short * y,
                                             unsigned char * decoded_bytes,
                                             unsigned short n,
                                             unsigned char max_iterations,
                                             unsigned char crc_type,
                                             unsigned char F,
                                             time_stats_t * init_stats,
                                             time_stats_t * alpha_stats,
                                             time_stats_t * beta_stats,
                                             time_stats_t * gamma_stats,
                                             time_stats_t * ext_stats,
                                             time_stats_t * intl1_stats,
                                             time_stats_t * intl2_stats )

```

Here is the call graph for this function:



◆ `print_bytes()`

```
void print_bytes ( char * s,  
simde_m128i * x )
```

Here is the call graph for this function:



Variable Documentation

◆ pi2tab

```
int* pi2tab[188]
```

◆ pi4tab

```
int * pi4tab[188]
```

◆ pi5tab

```
int * pi5tab[188]
```

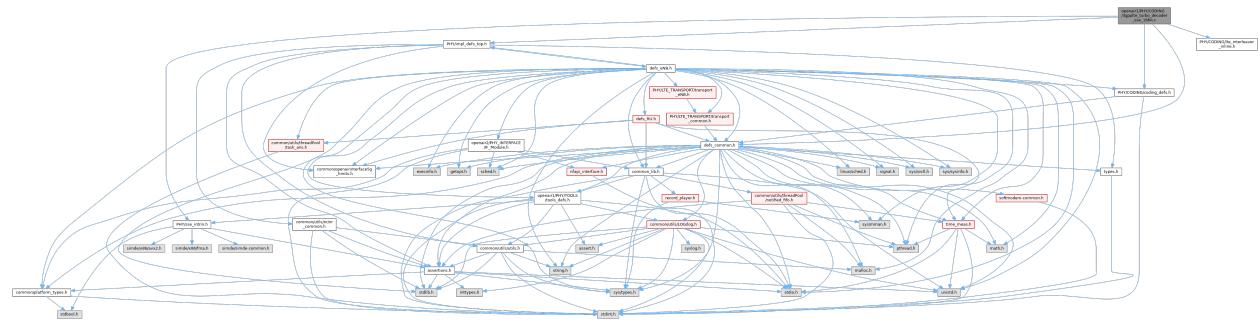
◆ pi6tab

```
int * pi6tab[188]
```

3gpplte_turbo_decoder_sse_16bit.c File Reference

```
#include "PHY/sse_intrin.h"
#include "PHY/impl_defs_top.h"
#include "PHY/defs_common.h"
#include "PHY/CODING/coding_defs.h"
#include "PHY/CODING/lte_interleaver_inline.h"
```

Include dependency graph for 3gpplte_turbo_decoder_sse_16bit.c:



Macros

```
#define MAX 256
#define L 40
```

Functions

```
void log_map16 (llr_t *systematic, channel_t *y_parity, llr_t *m11, llr_t *m10,
llr_t *alpha, llr_t *beta, llr_t *ext, unsigned short frame_length, unsigned
char term_flag, unsigned char F, int offset8_flag, time_stats_t *alpha_stats,
time_stats_t *beta_stats, time_stats_t *gamma_stats, time_stats_t
*ext_stats)

void compute_gamma16 (llr_t *m11, llr_t *m10, llr_t *systematic, channel_t
*y_parity, unsigned short frame_length, unsigned char term_flag)

void compute_alpha16 (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10,
unsigned short frame_length, unsigned char F)

void compute_beta16 (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10,
unsigned short frame_length, unsigned char F, int offset8_flag)

void compute_ext16 (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10, llr_t
*extrinsic, llr_t *ap, unsigned short frame_length)

void free_td16 (void)

void init_td16 (void)

uint8_t phy_threegpplte_turbo_decoder16 (int16_t *y, int16_t *y2, uint8_t
*decoded_bytes, uint8_t *decoded_bytes2, uint16_t n, uint8_t
max_iterations, uint8_t crc_type, uint8_t F, time_stats_t *init_stats,
time_stats_t *alpha_stats, time_stats_t *beta_stats, time_stats_t
*gamma_stats, time_stats_t *ext_stats, time_stats_t *intl1_stats,
time_stats_t *intl2_stats, decode_abort_t *ab)
```

Variables

```
int * pi2tab16 [188]

int * pi5tab16 [188]

int * pi4tab16 [188]

int * pi6tab16 [188]
```

Macro Definition Documentation



```
#define L 40
```

◆ MAX

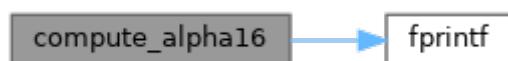
```
#define MAX 256
```

Function Documentation

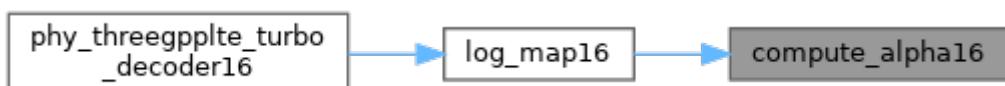
◆ compute_alpha16()

void compute_alpha16	(llr_t *	alpha,
		llr_t *	beta,
		llr_t *	m11,
		llr_t *	m10,
		unsigned short	frame_length,
		unsigned char	F)

Here is the call graph for this function:



Here is the caller graph for this function:



◆ compute_beta16()

void compute_beta16	(llr_t *	alpha,
		llr_t *	beta,
		llr_t *	m11,
		llr_t *	m10,
		unsigned short	frame_length,
		unsigned char	F,
		int	offset8_flag)

Here is the call graph for this function:



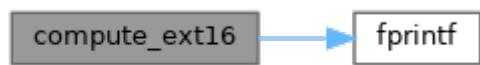
Here is the caller graph for this function:



◆ compute_ext16()

void compute_ext16	(llr_t *	alpha,
		llr_t *	beta,
		llr_t *	m11,
		llr_t *	m10,
		llr_t *	extrinsic,
		llr_t *	ap,
		unsigned short	frame_length)

Here is the call graph for this function:



Here is the caller graph for this function:



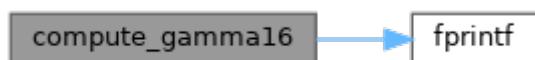
◆ compute_gamma16()

```

void compute_gamma16 ( llr_t * m11,
                      llr_t * m10,
                      llr_t * systematic,
                      channel_t * y_parity,
                      unsigned short frame_length,
                      unsigned char term_flag )

```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ free_td16()

```

void free_td16 ( void )

```

◆ init_td16()

```

void init_td16 ( void )

```

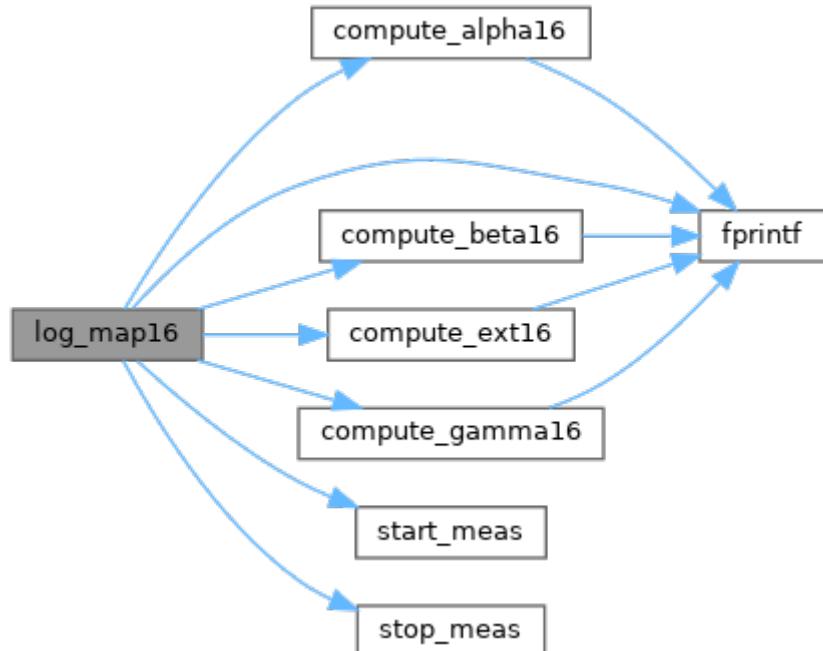
◆ log_map16()

```

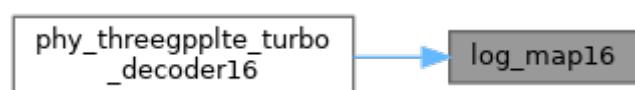
void log_map16( IIR_t *systematic,
                channel_t *y_parity,
                IIR_t *m11,
                IIR_t *m10,
                IIR_t *alpha,
                IIR_t *beta,
                IIR_t *ext,
                unsigned short frame_length,
                unsigned char term_flag,
                unsigned char F,
                int offset8_flag,
                time_stats_t *alpha_stats,
                time_stats_t *beta_stats,
                time_stats_t *gamma_stats,
                time_stats_t *ext_stats )

```

Here is the call graph for this function:



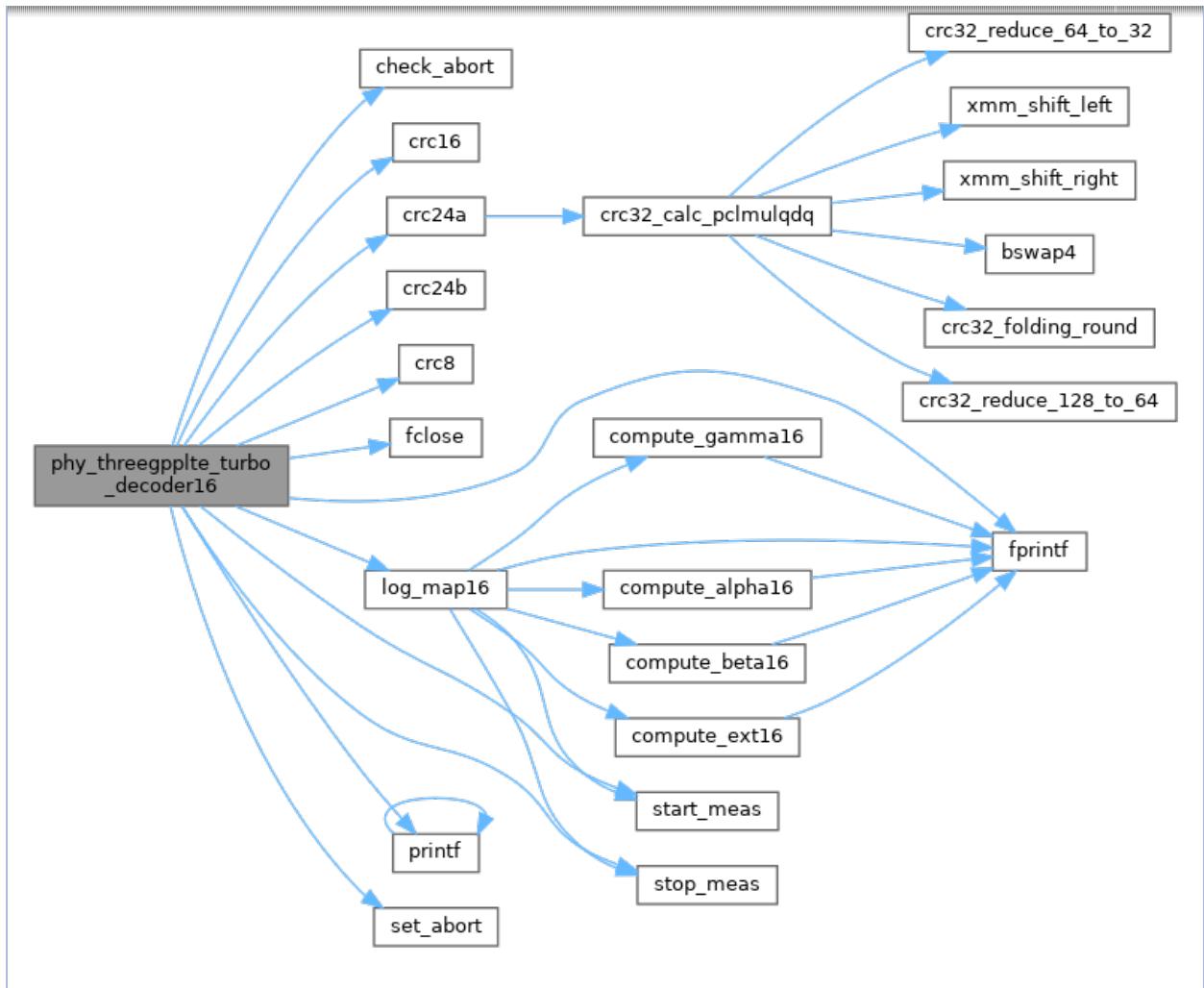
Here is the caller graph for this function:



◆ phy_threegpplte_turbo_decoder16()

```
uint8_t phy_threegpplte_turbo_decoder16( int16_t *y,  
                                         int16_t *y2,  
                                         uint8_t *decoded_bytes,  
                                         uint8_t *decoded_bytes2,  
                                         uint16_t n,  
                                         uint8_t max_iterations,  
                                         uint8_t crc_type,  
                                         uint8_t F,  
                                         time_stats_t *init_stats,  
                                         time_stats_t *alpha_stats,  
                                         time_stats_t *beta_stats,  
                                         time_stats_t *gamma_stats,  
                                         time_stats_t *ext_stats,  
                                         time_stats_t *intI1_stats,  
                                         time_stats_t *intI2_stats,  
                                         decode_abort_t *ab )
```

Here is the call graph for this function:



Variable Documentation

◆ pi2tab16

```
int* pi2tab16[188]
```

◆ pi4tab16

```
int * pi4tab16[188]
```

◆ pi5tab16

```
int * pi5tab16[188]
```

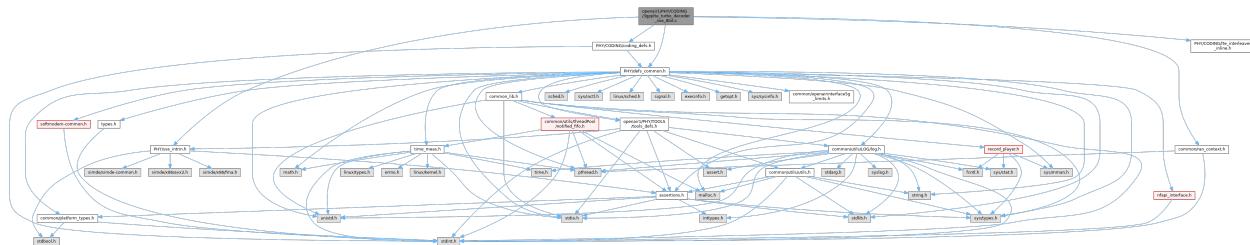
◆ pi6tab16

```
int * pi6tab16[188]
```

3gpplte_turbo_decoder_sse_8bit.c File Reference

```
#include "PHY/sse_intrin.h"
#include "PHY/defs_common.h"
#include "PHY/CODING/coding_defs.h"
#include "PHY/CODING/lte_interleaver_inline.h"
#include "common/ran_context.h"
```

Include dependency graph for 3gpplte_turbo_decoder_sse_8bit.c:



Macros

```
#define SHUFFLE16(a, b, c, d, e, f, g, h)
#define MAX8 127
#define L 16
```

Typedefs

```
typedef int8_t llr_t
typedef int8_t channel_t
```

Functions

```
void log_map8 (llr_t *systematic, channel_t *y_parity, llr_t *m11, llr_t *m10,  
llr_t *alpha, llr_t *beta, llr_t *ext, unsigned short frame_length, unsigned  
char term_flag, unsigned char F, int offset8_flag, time_stats_t *alpha_stats,  
time_stats_t *beta_stats, time_stats_t *gamma_stats, time_stats_t  
*ext_stats)  
void compute_gamma8 (llr_t *m11, llr_t *m10, llr_t *systematic, channel_t  
*y_parity, unsigned short frame_length, unsigned char term_flag)  
void compute_alpha8 (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10,  
unsigned short frame_length, unsigned char F)  
void compute_beta8 (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10, unsigned  
short frame_length, unsigned char F, int offset8_flag)  
void compute_ext8 (llr_t *alpha, llr_t *beta, llr_t *m11, llr_t *m10, llr_t  
*extrinsic, llr_t *ap, unsigned short frame_length)  
void print_bytes (char *s, int8_t *x)  
void free_td8 (void)  
void init_td8 (void)  
uint8_t phy_threegpplte_turbo_decoder8 (int16_t *y, int16_t *y2, uint8_t  
*decoded_bytes, uint8_t *decoded_bytes2, uint16_t n, uint8_t  
max_iterations, uint8_t crc_type, uint8_t F, time_stats_t *init_stats,  
time_stats_t *alpha_stats, time_stats_t *beta_stats, time_stats_t  
*gamma_stats, time_stats_t *ext_stats, time_stats_t *intl1_stats,  
time_stats_t *intl2_stats, decode_abort_t *ab)
```

Variables

```
int * pi2tab8 [188]  
int * pi5tab8 [188]  
int * pi4tab8 [188]  
int * pi6tab8 [188]  
RAN_CONTEXT_t RC
```

Macro Definition Documentation



```
#define L 16
```

◆ MAX8

```
#define MAX8 127
```

◆ SHUFFLE16

```
#define SHUFFLE16
```

(
	a,
	b,
	c,
	d,
	e,
	f,
	g,
	h)

Value:

```
simde_mm_set_epi8(h == -1 ? -1 : h * 2 + 1, \
h == -1 ? -1 : h * 2, \
g == -1 ? -1 : g * 2 + 1, \
g == -1 ? -1 : g * 2, \
f == -1 ? -1 : f * 2 + 1, \
f == -1 ? -1 : f * 2, \
e == -1 ? -1 : e * 2 + 1, \
e == -1 ? -1 : e * 2, \
d == -1 ? -1 : d * 2 + 1, \
d == -1 ? -1 : d * 2, \
c == -1 ? -1 : c * 2 + 1, \
c == -1 ? -1 : c * 2, \
b == -1 ? -1 : b * 2 + 1, \
b == -1 ? -1 : b * 2, \
a == -1 ? -1 : a * 2 + 1, \
a == -1 ? -1 : a * 2);
```

Typedef Documentation

◆ channel_t

```
typedef int8_t channel_t
```

◆ llr_t

```
typedef int8_t llr_t
```

Function Documentation

◆ compute_alpha8()

void compute_alpha8	(llr_t *	alpha,
		llr_t *	beta,
		llr_t *	m11,
		llr_t *	m10,
		unsigned short	frame_length,
		unsigned char	F)

Here is the caller graph for this function:



◆ compute_beta8()

```

void compute_beta8 ( llr_t * alpha,
                     llr_t * beta,
                     llr_t * m11,
                     llr_t * m10,
                     unsigned short frame_length,
                     unsigned char F,
                     int offset8_flag )

```

Here is the caller graph for this function:



◆ compute_ext8()

```

void compute_ext8 ( llr_t * alpha,
                    llr_t * beta,
                    llr_t * m11,
                    llr_t * m10,
                    llr_t * extrinsic,
                    llr_t * ap,
                    unsigned short frame_length )

```

Here is the call graph for this function:



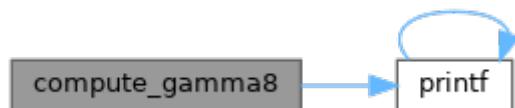
Here is the caller graph for this function:



◆ compute_gamma8()

```
void compute_gamma8 ( llr_t * m11,
                     llr_t * m10,
                     llr_t * systematic,
                     channel_t * y_parity,
                     unsigned short frame_length,
                     unsigned char term_flag )
```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ free_td8()

```
void free_td8 ( void )
```

◆ init_td8()

```
void init_td8 ( void )
```

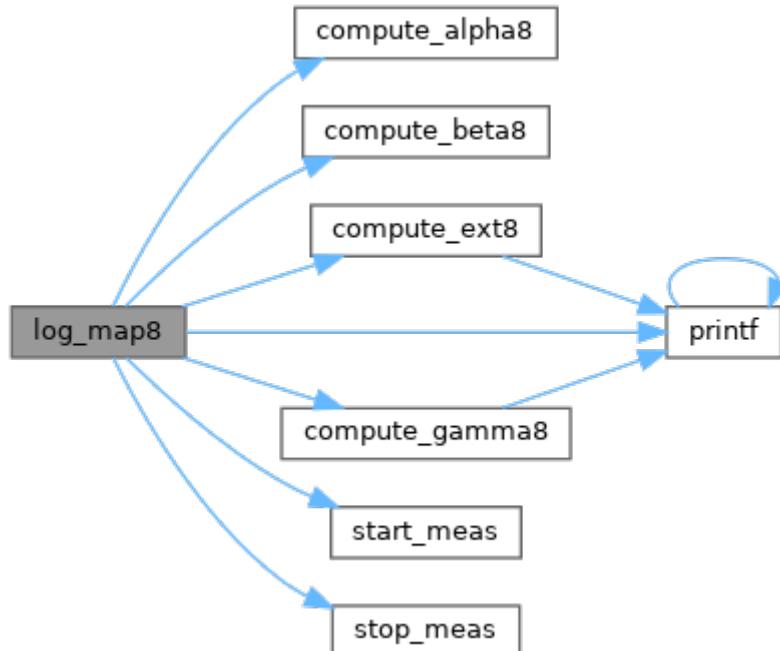
◆ log_map8()

```

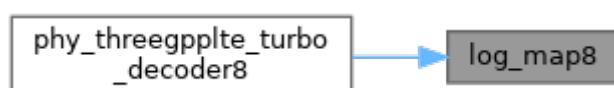
void log_map8( IIR_t *systematic,
               channel_t *y_parity,
               IIR_t *m11,
               IIR_t *m10,
               IIR_t *alpha,
               IIR_t *beta,
               IIR_t *ext,
               unsigned short frame_length,
               unsigned char term_flag,
               unsigned char F,
               int offset8_flag,
               time_stats_t *alpha_stats,
               time_stats_t *beta_stats,
               time_stats_t *gamma_stats,
               time_stats_t *ext_stats )

```

Here is the call graph for this function:



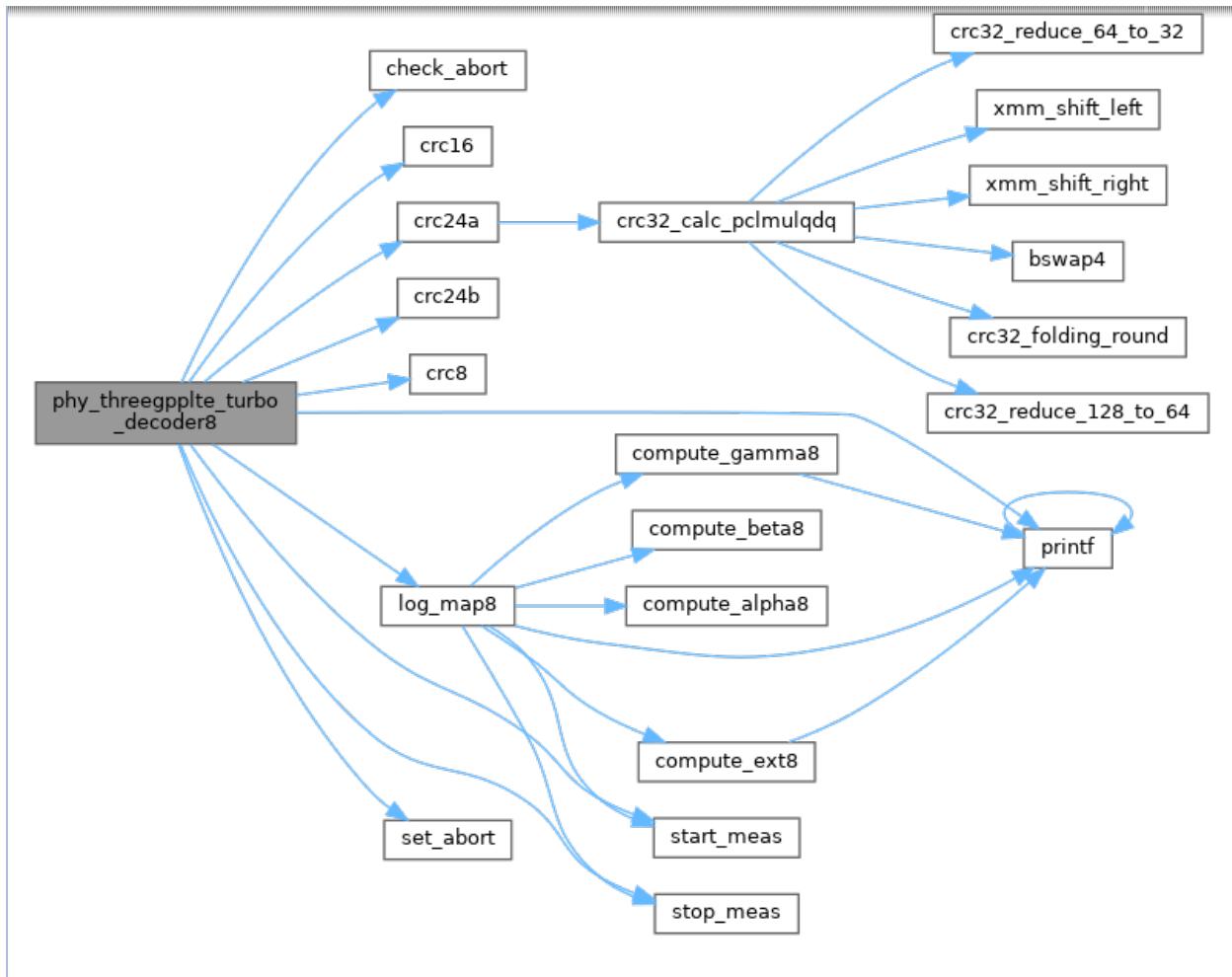
Here is the caller graph for this function:



◆ phy_threegpplte_turbo_decoder8()

```
uint8_t phy_threegpplte_turbo_decoder8 ( int16_t * y,  
                                         int16_t * y2,  
                                         uint8_t * decoded_bytes,  
                                         uint8_t * decoded_bytes2,  
                                         uint16_t n,  
                                         uint8_t max_iterations,  
                                         uint8_t crc_type,  
                                         uint8_t F,  
                                         time_stats_t * init_stats,  
                                         time_stats_t * alpha_stats,  
                                         time_stats_t * beta_stats,  
                                         time_stats_t * gamma_stats,  
                                         time_stats_t * ext_stats,  
                                         time_stats_t * intl1_stats,  
                                         time_stats_t * intl2_stats,  
                                         decode_abort_t * ab )
```

Here is the call graph for this function:



◆ print_bytes()

```
void print_bytes( char * s, int8_t * x )
```

Here is the call graph for this function:



Variable Documentation

◆ pi2tab8

`int* pi2tab8[188]`

◆ `pi4tab8`

`int * pi4tab8[188]`

◆ `pi5tab8`

`int * pi5tab8[188]`

◆ `pi6tab8`

`int * pi6tab8[188]`

◆ `RC`

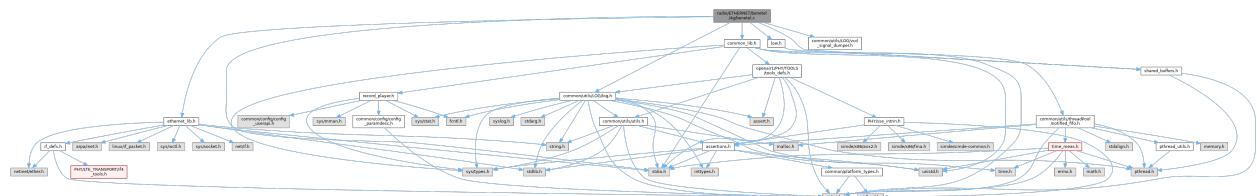
`RAN_CONTEXT_t RC`

`extern`

benetel.c File Reference

```
#include <stdio.h>
#include "common_lib.h"
#include "ethernet_lib.h"
#include "shared_buffers.h"
#include "low.h"
#include "common/utils/LOG/log.h"
#include "common/utils/LOG/vcd_signal_dumper.h"
```

Include dependency graph for benetel.c:



Classes

```
struct benetel_eth_state_t
```

Functions

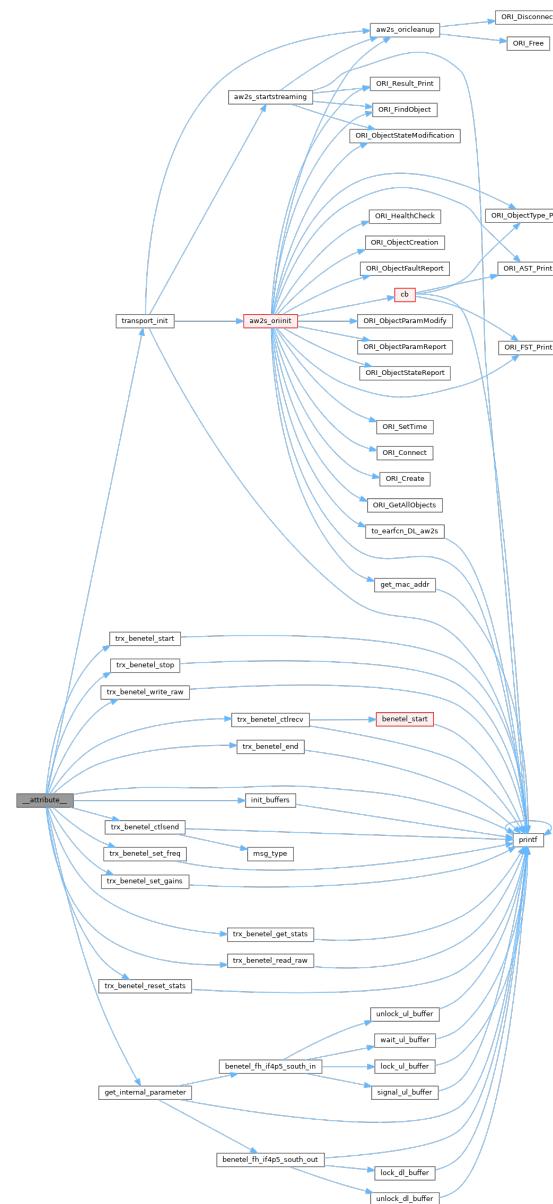
```
int trx_benetel_start (openair0_device *device)
void trx_benetel_end (openair0_device *device)
int trx_benetel_stop (openair0_device *device)
int trx_benetel_set_freq (openair0_device *device, openair0_config_t
*openair0_cfg)
int trx_benetel_set_gains (openair0_device *device, openair0_config_t
*openair0_cfg)
int trx_benetel_get_stats (openair0_device *device)
int trx_benetel_reset_stats (openair0_device *device)
int ethernet_tune (openair0_device *device, unsigned int option, int value)
    this function allows you to configure certain ethernet parameters in socket or
    device level
int trx_benetel_write_raw (openair0_device *device, openair0_timestamp
    timestamp, void **buff, int nsamps, int cc, int flags)
int trx_benetel_read_raw (openair0_device *device, openair0_timestamp
    *timestamp, void **buff, int nsamps, int cc)
char * msg_type (int t)
int trx_benetel_ctlsend (openair0_device *device, void *msg, ssize_t msg_len)
int trx_benetel_ctlrecv (openair0_device *device, void *msg, ssize_t msg_len)
void benetel_fh_if4p5_south_in (RU_t *ru, int *frame, int *subframe)
void benetel_fh_if4p5_south_out (RU_t *ru, int frame, int subframe, uint64_t
    timestamp)
void * get_internal_parameter (char *name)
    _attribute_ ((_visibility_("default")))
```

Function Documentation

◆ **_attribute_()**

```
_attribute_ ( __visibility__("default") )
```

Here is the call graph for this function:



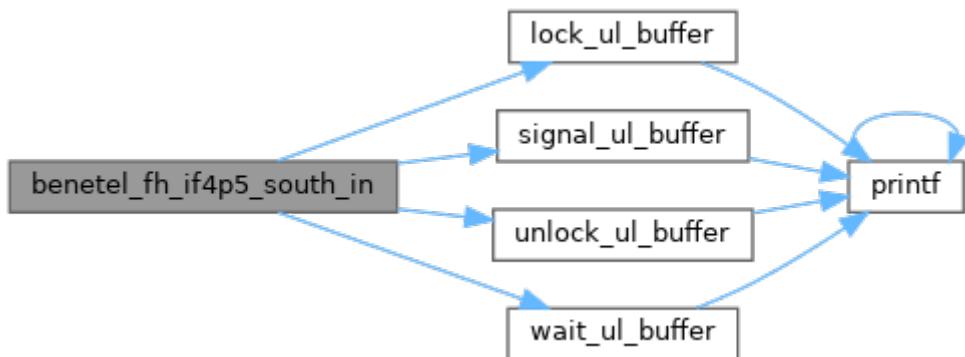
◆ **benetel_fh_if4p5_south_in()**

```

void benetel_fh_if4p5_south_in ( RU_t * ru,
                                 int * frame,
                                 int * subframe )

```

Here is the call graph for this function:



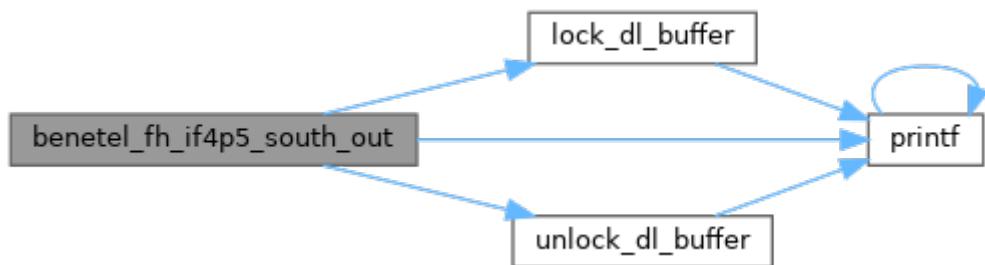
Here is the caller graph for this function:



◆ benetel_fh_if4p5_south_out()

void benetel_fh_if4p5_south_out	(RU_t *	ru,
	int	frame,	
	int	subframe,	
	uint64_t	timestamp)	

Here is the call graph for this function:



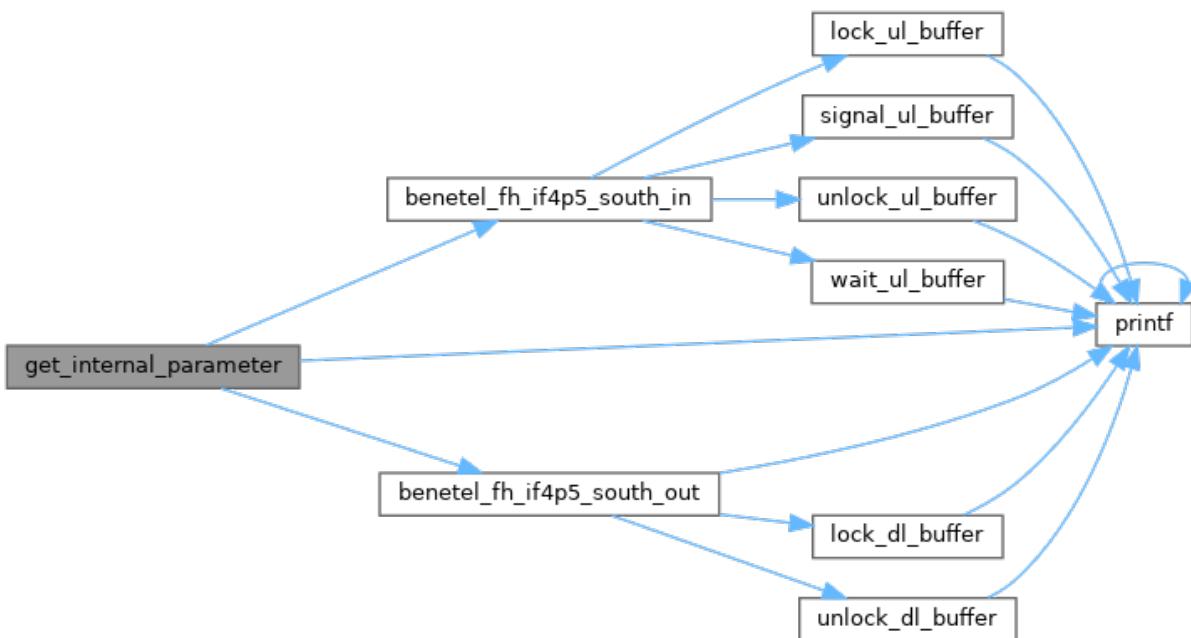
Here is the caller graph for this function:



◆ get_internal_parameter()

```
void * get_internal_parameter ( char * name )
```

Here is the call graph for this function:



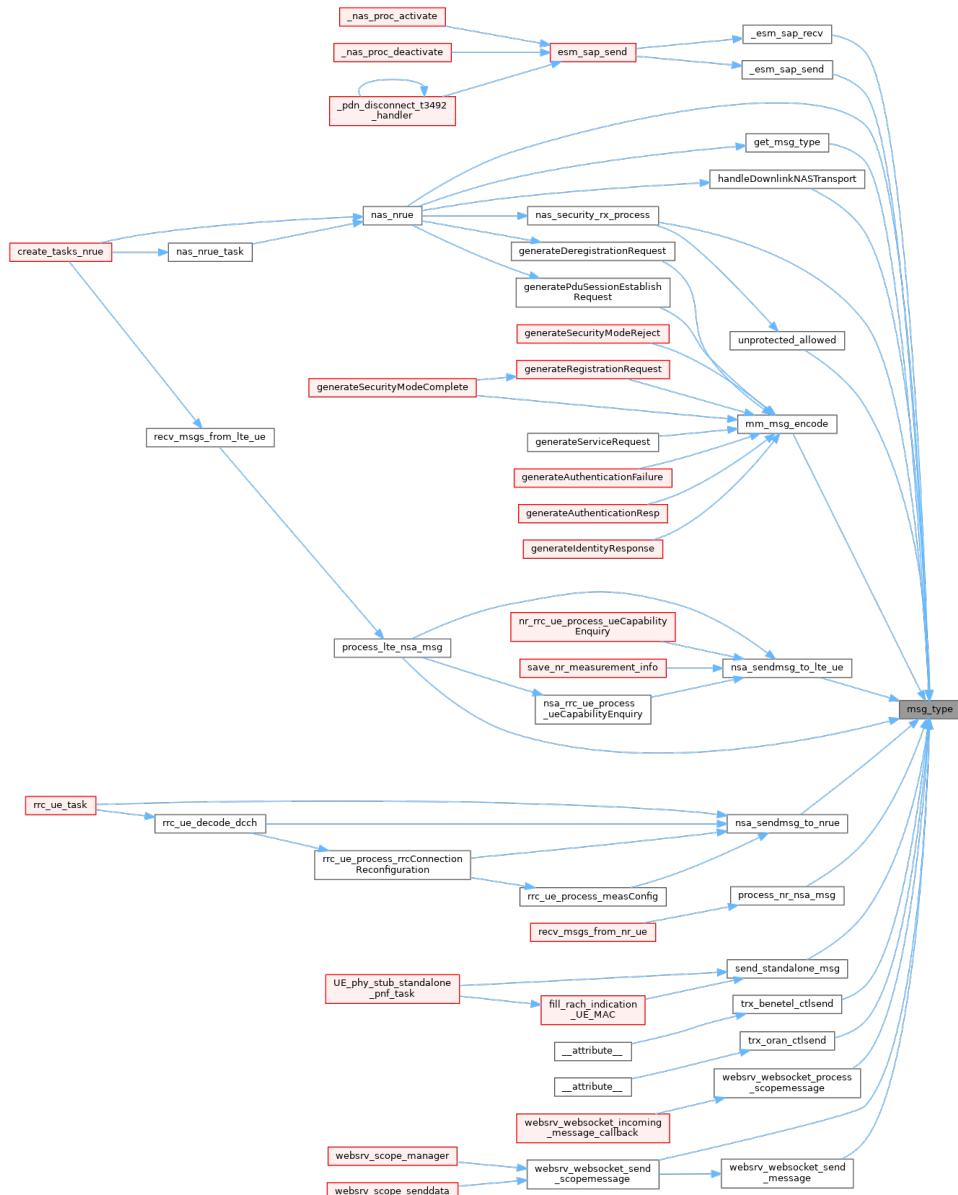
Here is the caller graph for this function:



◆ `msg_type()`

char * msg_type	(int	t)
-----------------	---	-----	---	---

Here is the caller graph for this function:



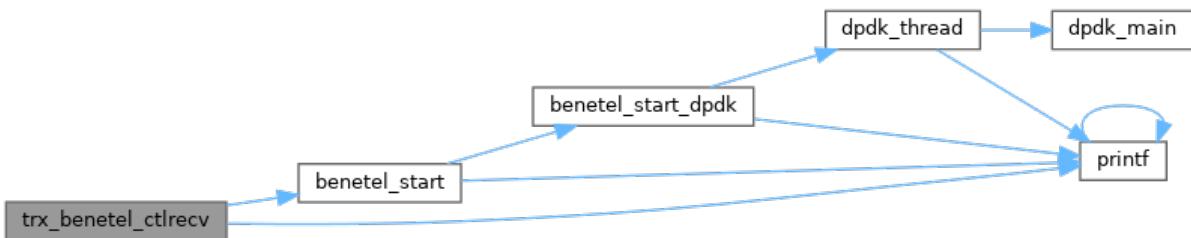
◆ `trx_benetel_ctlrecv()`

```

int trx_benetel_ctlrecv
(
    openair0_device * device,
    void * msg,
    ssize_t msg_len
)

```

Here is the call graph for this function:



Here is the caller graph for this function:



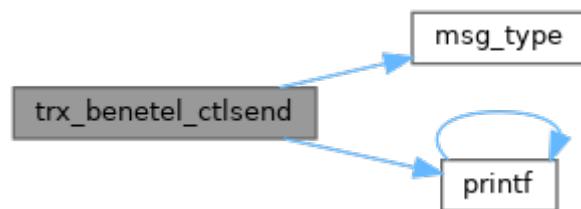
◆ `trx_benetel_ctlsend()`

```

int trx_benetel_ctlsend
(
    openair0_device * device,
    void * msg,
    ssize_t msg_len
)

```

Here is the call graph for this function:



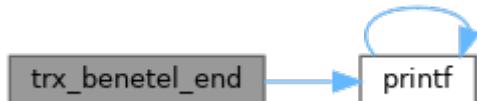
Here is the caller graph for this function:



◆ `trx_benetel_end()`

```
void trx_benetel_end ( openair0_device * device )
```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ `trx_benetel_get_stats()`

```
int trx_benetel_get_stats ( openair0_device * device )
```

Here is the call graph for this function:



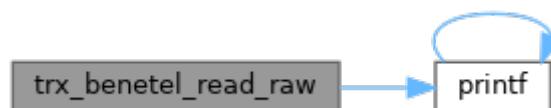
Here is the caller graph for this function:



◆ `trx_benetel_read_raw()`

int trx_benetel_read_raw	(openair0_device *	device,
		openair0_timestamp *	timestamp,
		void **	buff,
		int	nsamps,
		int	cc)

Here is the call graph for this function:



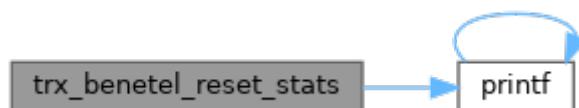
Here is the caller graph for this function:



◆ **trx_benetel_reset_stats()**

int trx_benetel_reset_stats	(openair0_device *	device)
-----------------------------	---	--------------------------	--------	---

Here is the call graph for this function:



Here is the caller graph for this function:



◆ **trx_benetel_set_freq()**

```
int trx_benetel_set_freq ( openair0_device * device,  
                           openair0_config_t * openair0_cfg )
```

Here is the call graph for this function:



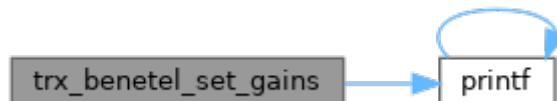
Here is the caller graph for this function:



◆ **trx_benetel_set_gains()**

```
int trx_benetel_set_gains ( openair0_device * device,  
                           openair0_config_t * openair0_cfg )
```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ **trx_benetel_start()**

```
int trx_benetel_start ( openair0_device * device )
```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ **trx_benetel_stop()**

```
int trx_benetel_stop ( openair0_device * device )
```

Here is the call graph for this function:



Here is the caller graph for this function:



◆ **trx_benetel_write_raw()**

int trx_benetel_write_raw	(openair0_device *	device,
		openair0_timestamp	timestamp,
		void **	buff,
		int	nsamps,
		int	cc,
		int	flags)

Here is the call graph for this function:



Here is the caller graph for this function:



dpdk_driver.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <inttypes.h>
#include <sys/types.h>
#include <sys/queue.h>
#include <netinet/in.h>
#include <setjmp.h>
#include <stdarg.h>
#include <ctype.h>
#include <errno.h>
#include <getopt.h>
#include <signal.h>
#include <stdbool.h>
#include <rte_common.h>
#include <rte_log.h>
#include <rte_malloc.h>
#include <rte_memory.h>
#include <rte_memcpy.h>
#include <rte_eal.h>
#include <rte_launch.h>
#include <rte_atomic.h>
#include <rte_cycles.h>
#include <rte_prefetch.h>
#include <rte_lcore.h>
#include <rte_per_lcore.h>
#include <rte_branch_prediction.h>
#include <rte_interrupts.h>
#include <rte_random.h>
#include <rte_debug.h>
#include <rte_ether.h>
```

```
#include <rte_ethdev.h>  
#include <rte_mempool.h>  
#include <rte_mbuf.h>  
#include "low.h"
```

Include dependency graph for dpdk_driver.c:



Classes

```
struct lcore_queue_conf  
struct l2fwd_port_statistics
```

Macros

```
#define ANT_NUM buf[23]
#define PAYLOAD_1 buf[20]
#define PAYLOAD_2 buf[21]
#define ETH_TYPE buf[17]
#define SYMBOL buf[29]
#define SUBFRAME buf[28]
#define FRAME buf[27]
#define RTE_LOGTYPE_L2FWD RTE_LOGTYPE_USER1
#define MAX_PKT_BURST 32
#define BURST_TX_DRAIN_US 5 /* TX drain every ~100us */
#define MEMPOOL_CACHE_SIZE 256
#define RTE_TEST_RX_DESC_DEFAULT 1024
#define RTE_TEST_TX_DESC_DEFAULT 1024
#define MAX_RX_QUEUE_PER_LCORE 16
#define MAX_TX_QUEUE_PER_PORT 16
#define MAX_TIMER_PERIOD 86400 /* 1 day max */
#define CMD_LINE_OPT_MAC_UPDATING "mac-updating"
#define CMD_LINE_OPT_NO_MAC_UPDATING "no-mac-updating"
#define CHECK_INTERVAL 100 /* 100ms */
#define MAX_CHECK_TIME 90 /* 9s (90 * 100ms) in total */
```

Enumerations

```
enum { CMD_LINE_OPT_MIN_NUM = 256 }
```

Functions

```
static void print_stats (void)
static void I2fwd_simple_forward (struct rte_mbuf *m, unsigned portid,
                                  benetel_t *bs)
static void I2fwd_main_loop (benetel_t *bs)
static int I2fwd_launch_one_lcore (void *bs)
static void I2fwd_usage (const char *prgname)
static int I2fwd_parse_portmask (const char *portmask)
static unsigned int I2fwd_parse_nqueue (const char *q_arg)
static int I2fwd_parse_timer_period (const char *q_arg)
static int I2fwd_parse_args (int argc, char **argv)
static void check_all_ports_link_status (uint32_t port_mask)
static void signal_handler (int signum)
int dpdk_main (int argc, char **argv, benetel_t *bs)
```

Variables

```
static volatile bool force_quit
    unsigned short iq [33600]
    unsigned char iq_swap [67200]
    unsigned short * iq_ptr [14]
        unsigned int dl_start = 0
            static int mac_updating = 1
    static uint16_t nb_rxd = RTE_TEST_RX_DESC_DEFAULT
    static uint16_t nb_txd = RTE_TEST_TX_DESC_DEFAULT
    static struct ether_addr I2fwd_ports_eth_addr [RTE_MAX_ETHPORTS]
        static uint32_t I2fwd_enabled_port_mask = 0
        static uint32_t I2fwd_dst_ports [RTE_MAX_ETHPORTS]
        static unsigned int I2fwd_rx_queue_per_lcore = 1
    struct lcore_queue_conf _rte_cache_aligned
        struct lcore_queue_conf lcore_queue_conf [RTE_MAX_LCORE]
    static struct rte_eth_dev_tx_buffer * tx_buffer [RTE_MAX_ETHPORTS]
    static struct rte_eth_conf port_conf
        struct rte_mempool * I2fwd_pktnbuf_pool = NULL
    struct I2fwd_port_statistics port_statistics [RTE_MAX_ETHPORTS]
        static uint64_t timer_period = 10
        static const char short_options []
    static const struct option Igopts []
```

Macro Definition Documentation

◆ ANT_NUM

```
#define ANT_NUM buf[23]
```

◆ BURST_TX_DRAIN_US

```
#define BURST_TX_DRAIN_US 5 /* TX drain every ~100us */
```

◆ CHECK_INTERVAL

```
#define CHECK_INTERVAL 100 /* 100ms */
```

◆ CMD_LINE_OPT_MAC_UPDATING

```
#define CMD_LINE_OPT_MAC_UPDATING "mac-updating"
```

◆ CMD_LINE_OPT_NO_MAC_UPDATING

```
#define CMD_LINE_OPT_NO_MAC_UPDATING "no-mac-updating"
```

◆ ETH_TYPE

```
#define ETH_TYPE buf[17]
```

◆ FRAME

```
#define FRAME buf[27]
```

◆ MAX_CHECK_TIME

```
#define MAX_CHECK_TIME 90 /* 9s (90 * 100ms) in total */
```

◆ MAX_PKT_BURST

```
#define MAX_PKT_BURST 32
```

◆ MAX_RX_QUEUE_PER_LCORE

```
#define MAX_RX_QUEUE_PER_LCORE 16
```

◆ MAX_TIMER_PERIOD

```
#define MAX_TIMER_PERIOD 86400 /* 1 day max */
```

◆ MAX_TX_QUEUE_PER_PORT

```
#define MAX_TX_QUEUE_PER_PORT 16
```

◆ MEMPOOL_CACHE_SIZE

```
#define MEMPOOL_CACHE_SIZE 256
```

◆ PAYLOAD_1

```
#define PAYLOAD_1 buf[20]
```

◆ PAYLOAD_2

```
#define PAYLOAD_2 buf[21]
```

◆ RTE_LOGTYPE_L2FWD

```
#define RTE_LOGTYPE_L2FWD RTE_LOGTYPE_USER1
```

◆ RTE_TEST_RX_DESC_DEFAULT

```
#define RTE_TEST_RX_DESC_DEFAULT 1024
```

◆ RTE_TEST_TX_DESC_DEFAULT

```
#define RTE_TEST_TX_DESC_DEFAULT 1024
```

◆ SUBFRAME

```
#define SUBFRAME buf[28]
```

◆ SYMBOL

```
#define SYMBOL buf[29]
```

Enumeration Type Documentation

◆ anonymous enum

anonymous enum

Enumerator

CMD_LINE_OPT_MIN_NUM

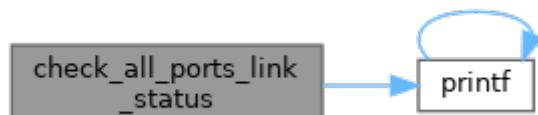
Function Documentation

◆ check_all_ports_link_status()

static void check_all_ports_link_status (**uint32_t** port_mask)

static

Here is the call graph for this function:



Here is the caller graph for this function:



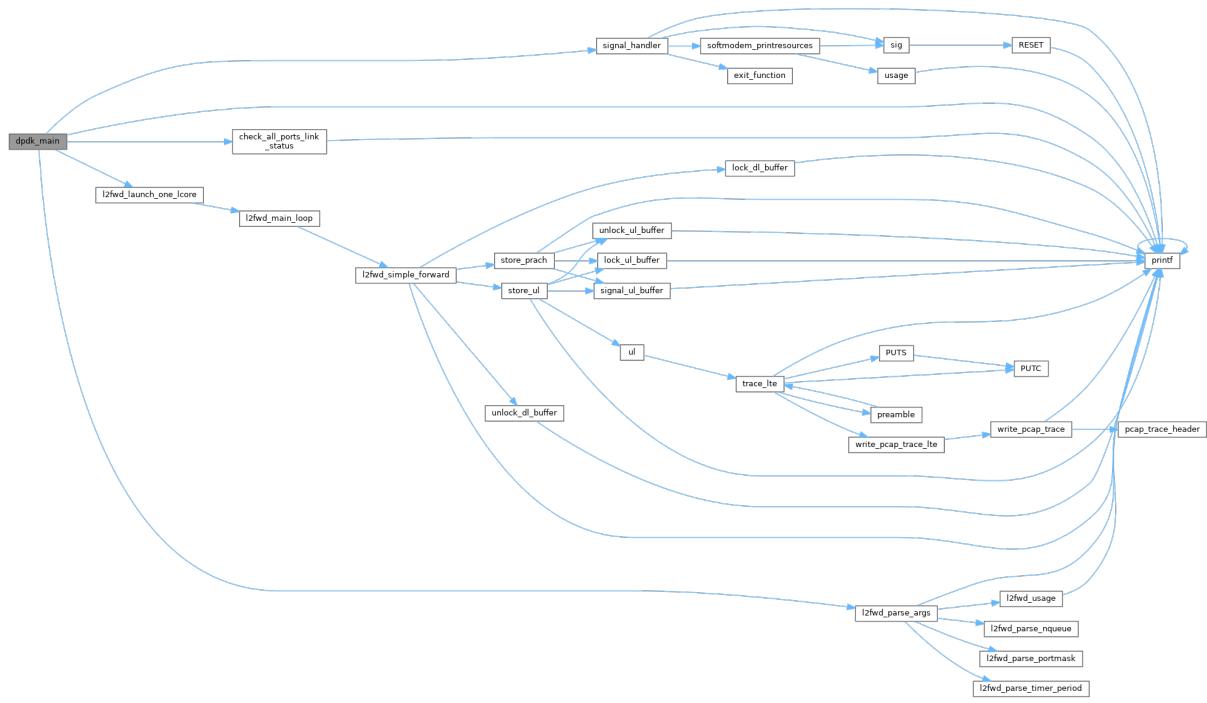
◆ dpdk_main()

```

int dpdk_main
(
    int argc,
    char **argv,
    benetel_t *bs
)

```

Here is the call graph for this function:



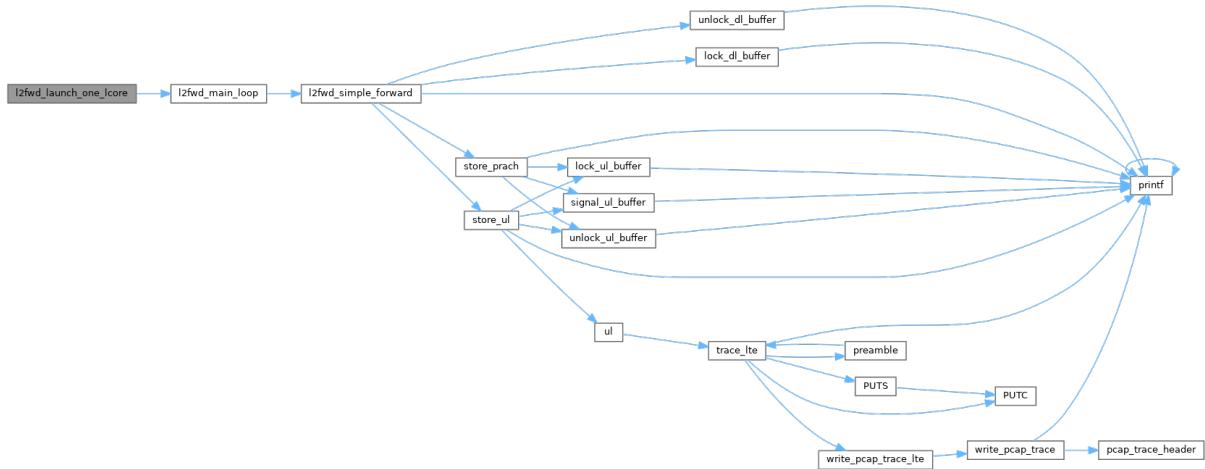
◆ l2fwd_launch_one_lcore()

```
static int l2fwd_launch_one_lcore
```

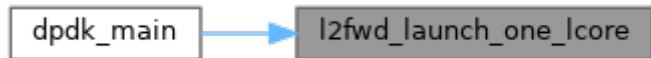
```
( void * bs )
```

```
static
```

Here is the call graph for this function:



Here is the caller graph for this function:

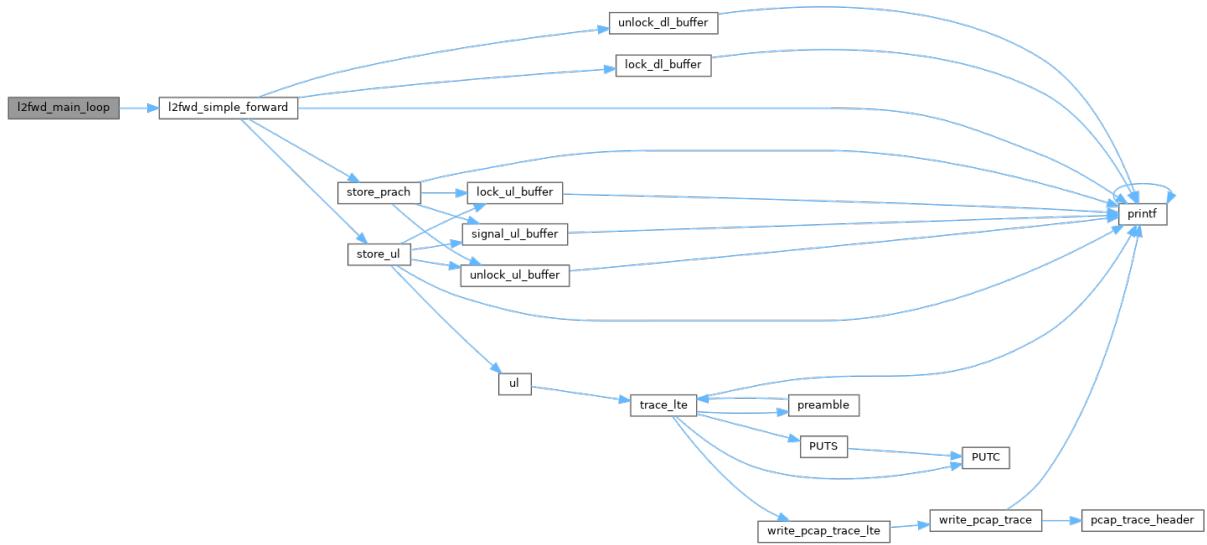


◆ l2fwd_main_loop()

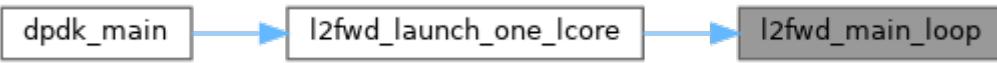
```
static void l2fwd_main_loop (      benetel_t *      bs      )
```

static

Here is the call graph for this function:



Here is the caller graph for this function:

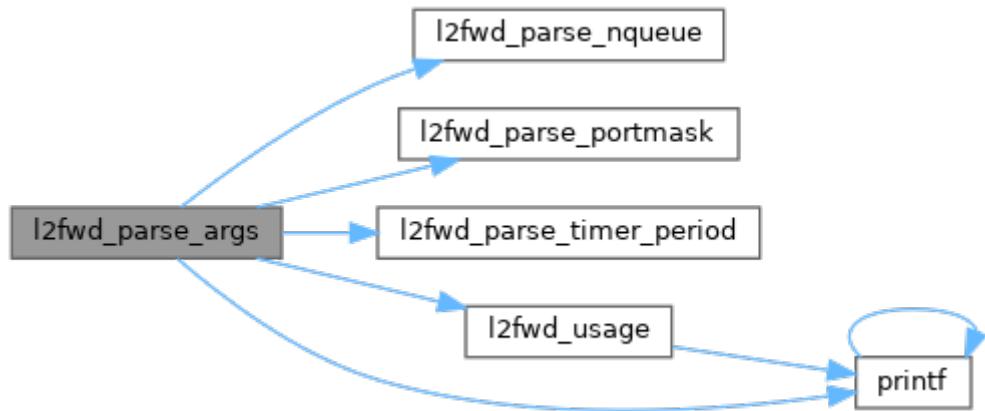


◆ `l2fwd_parse_args()`

```
static int l2fwd_parse_args ( int argc,  
                           char ** argv )
```

static

Here is the call graph for this function:



Here is the caller graph for this function:

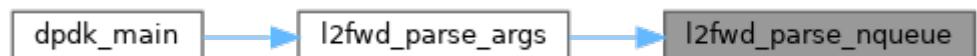


◆ l2fwd_parse_nqueue()

```
static unsigned int l2fwd_parse_nqueue ( const char * q_arg )
```

static

Here is the caller graph for this function:



◆ l2fwd_parse_portmask()

```
static int l2fwd_parse_portmask ( const char * portmask )
```

static

Here is the caller graph for this function:



◆ l2fwd_parse_timer_period()

```
static int l2fwd_parse_timer_period ( const char * q_arg )
```

static

Here is the caller graph for this function:



◆ l2fwd_simple_forward()

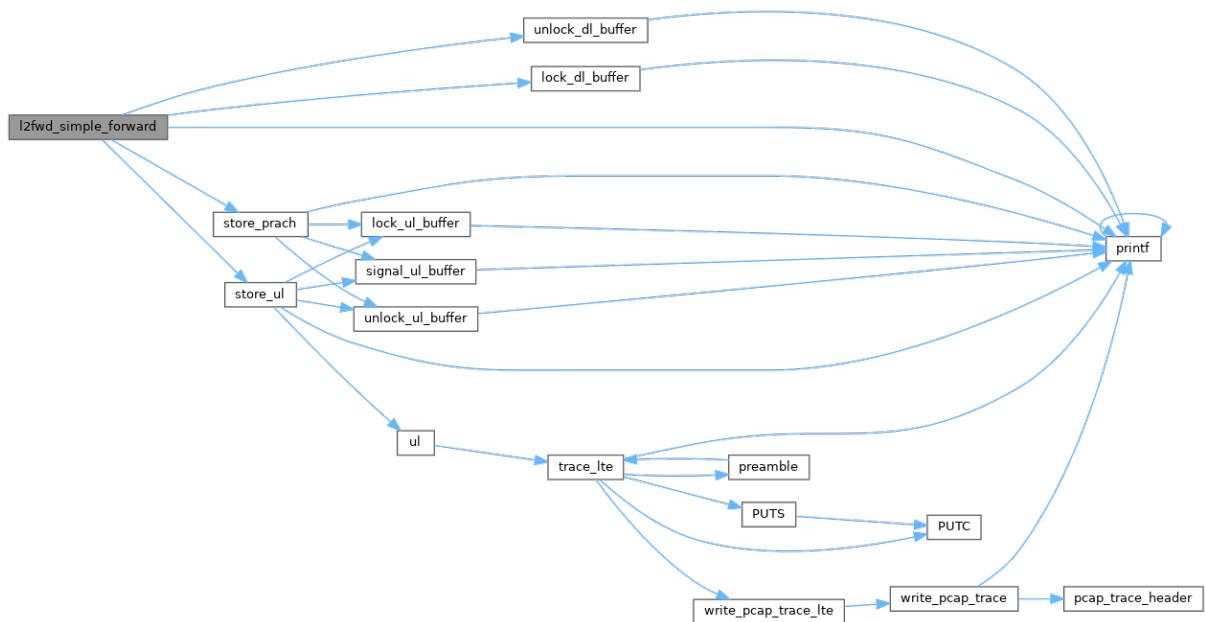
```

static void l2fwd_simple_forward
(
    struct rte_mbuf * m,
    unsigned portid,
    benetel_t * bs )

```

static

Here is the call graph for this function:



Here is the caller graph for this function:



◆ l2fwd_usage()

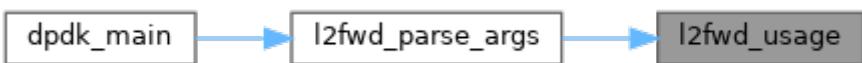
```
static void l2fwd_usage ( const char * prgname )
```

static

Here is the call graph for this function:



Here is the caller graph for this function:



◆ print_stats()

```
static void print_stats ( void )
```

static

Here is the call graph for this function:



Here is the caller graph for this function:



◆ signal_handler()

```
static void signal_handler ( int signum )
```

static

Here is the call graph for this function:



Variable Documentation

◆ _rte_cache_aligned

```
struct l2fwd_port_statistics _rte_cache_aligned
```

◆ dl_start

```
unsigned int dl_start = 0
```

◆ force_quit

```
volatile bool force_quit
```

static

◆ iq

```
unsigned short iq[33600]
```

◆ iq_ptr

```
unsigned short* iq_ptr[14]
```

◆ iq_swap

```
unsigned char iq_swap[67200]
```

◆ l2fwd_dst_ports

```
uint32_t l2fwd_dst_ports[RTE_MAX_ETHPORTS]
```

static

◆ l2fwd_enabled_port_mask

```
uint32_t l2fwd_enabled_port_mask = 0
```

static

◆ l2fwd_pktmbuf_pool

```
struct rte_mempool* l2fwd_pktmbuf_pool = NULL
```

◆ l2fwd_ports_eth_addr

```
struct ether_addr l2fwd_ports_eth_addr[RTE_MAX_ETHPORTS]
```

static

◆ l2fwd_rx_queue_per_lcore

```
unsigned int l2fwd_rx_queue_per_lcore = 1
```

static

◆ lcore_queue_conf

```
struct lcore_queue_conf lcore_queue_conf[RTE_MAX_LCORE]
```

◆ lgopts

```
const struct option lgopts[]
```

static

Initial value:

```
= {  
    { CMD_LINE_OPT_MAC_UPDATING, no_argument, &mac_updating, 1},  
    { CMD_LINE_OPT_NO_MAC_UPDATING, no_argument, &mac_updating, 0},  
    {NULL, 0, 0, 0}  
}
```

◆ mac_updating

```
int mac_updating = 1
```

static

◆ nb_rxd

```
uint16_t nb_rxd = RTE_TEST_RX_DESC_DEFAULT
```

static

◆ nb_txd

```
uint16_t nb_txd = RTE_TEST_TX_DESC_DEFAULT
```

static

◆ port_conf

```
struct rte_eth_conf port_conf
```

static

Initial value:

```
= {  
    .rxmode = {  
        .split_hdr_size = 0,  
        .offloads      = DEV_RX_OFFLOAD_JUMBO_FRAME,  
        .split_hdr_size = 0,  
        .max_rx_pkt_len = 9500,  
    },  
    .txmode = {  
        .mq_mode = ETH_MQ_TX_NONE,  
    },  
}
```

◆ port_statistics

```
struct l2fwd_port_statistics port_statistics[RTE_MAX_ETHPORTS]
```

◆ short_options

```
const char short_options[]
```

static

Initial value:

```
=  
    "p:  
    "q:  
    "T:"
```

◆ timer_period

```
uint64_t timer_period = 10
```

static

◆ tx_buffer

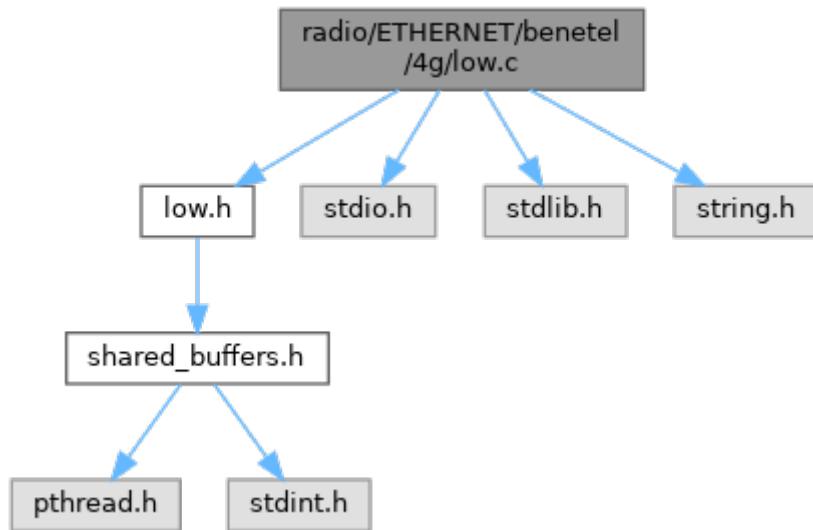
```
struct rte_eth_dev_tx_buffer* tx_buffer[RTE_MAX_ETHPORTS]
```

static

low.c File Reference

```
#include "low.h"  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>
```

Include dependency graph for low.c:



Functions

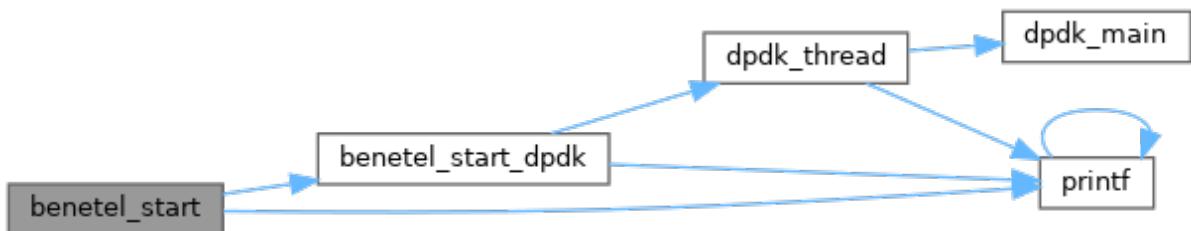
```
void store_ul (benetel_t *bs, ul_packet_t *ul)  
void store_prach (benetel_t *bs, int frame, int subframe, void *data)  
void * benetel_start_dpdk (char *ifname, shared_buffers *buffers, char  
*dpdk_main_command_line)  
void * benetel_start (char *ifname, shared_buffers *buffers, char  
*dpdk_main_command_line)
```

Function Documentation

◆ **benetel_start()**

void * benetel_start	(char *	ifname,
		shared_buffers *	buffers,
		char *	dpdk_main_command_line)

Here is the call graph for this function:



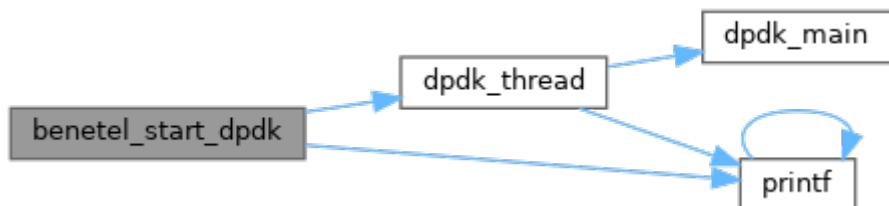
Here is the caller graph for this function:



◆ benetel_start_dpdk()

void * benetel_start_dpdk	(char *	ifname,
		shared_buffers *	buffers,
		char *	dpdk_main_command_line)

Here is the call graph for this function:



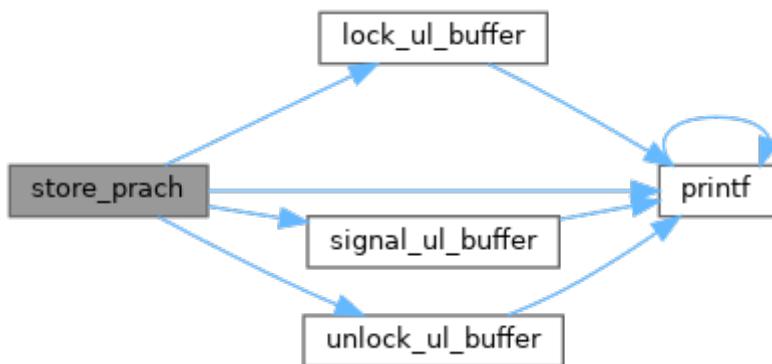
Here is the caller graph for this function:



◆ store_prach()

```
void store_prach ( benetel_t *bs,  
                   int frame,  
                   int subframe,  
                   void *data )
```

Here is the call graph for this function:



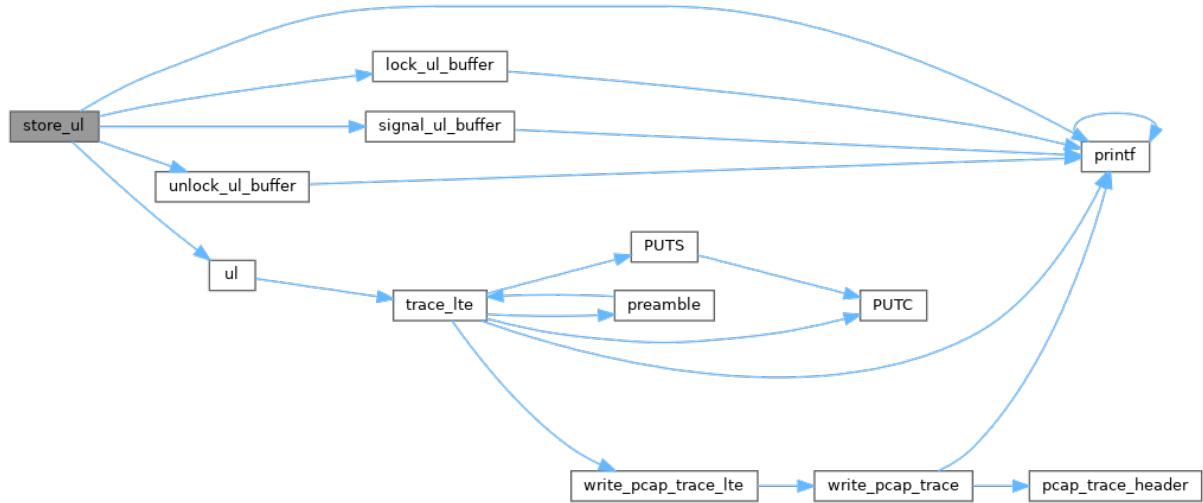
Here is the caller graph for this function:



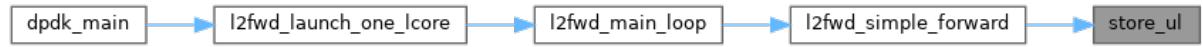
◆ store_ul()

void store_ul	(benetel_t *	bs,
		ul_packet_t *	ul)

Here is the call graph for this function:



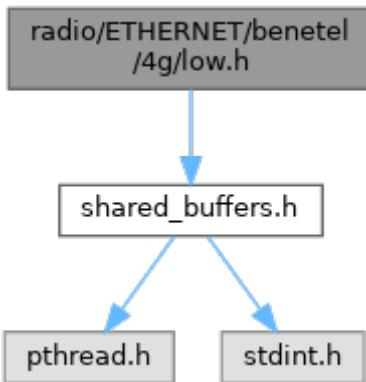
Here is the caller graph for this function:



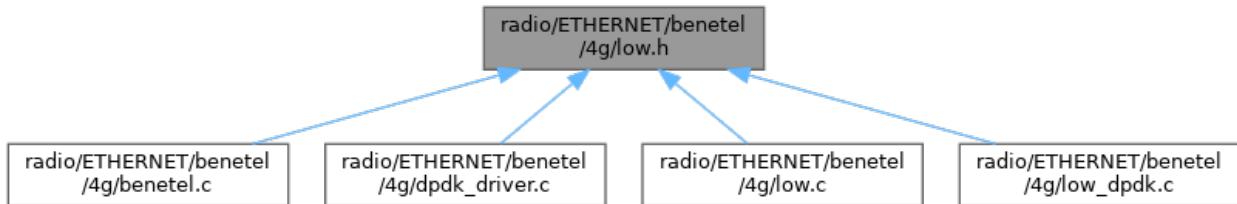
low.h File Reference

```
#include "shared_buffers.h"
```

Include dependency graph for low.h:



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



Go to the source code of this file.

Classes

```
struct benetel_t
```

```
struct ul_packet_t
```

Functions

```
void * benetel_start (char *ifname, shared_buffers *buffers, char  
*dpdk_main_command_line)
```

```
void store_ul (benetel_t *bs, ul_packet_t *ul)
```

```
void store_prach (benetel_t *bs, int frame, int subframe, void *data)
```

Function Documentation

◆ benetel_start()

void * benetel_start	(char *	ifname,
		shared_buffers *	buffers,
		char *	dplk_main_command_line)

◆ store_prach()

void store_prach	(benetel_t *	bs,
		int	frame,
		int	subframe,
		void *	data)

◆ store_ul()

void store_ul	(benetel_t *	bs,
		ul_packet_t *	ul)

low.h

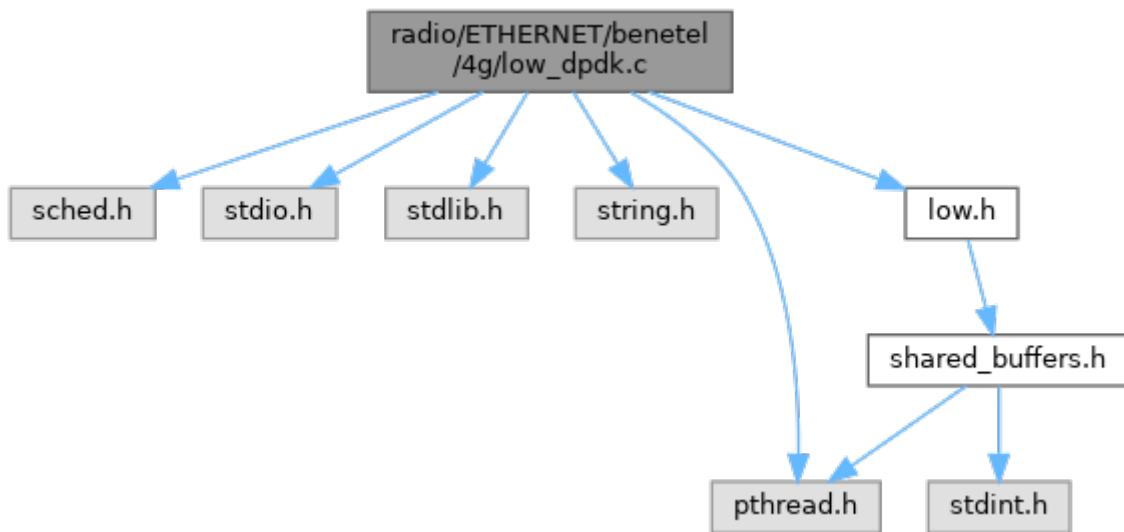
Go to the documentation of this file.

```
1  /*
2   * Licensed to the OpenAirInterface (OAI) Software Alliance under one or more
3   * contributor license agreements. See the NOTICE file distributed with
4   * this work for additional information regarding copyright ownership.
5   * The OpenAirInterface Software Alliance licenses this file to You under
6   * the OAI Public License, Version 1.1 (the "License"); you may not use this file
7   * except in compliance with the License.
8   * You may obtain a copy of the License at
9   *
10  *     http://www.openairinterface.org/?page_id=698
11  *
12  * Unless required by applicable law or agreed to in writing, software
13  * distributed under the License is distributed on an "AS IS" BASIS,
14  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15  * See the License for the specific language governing permissions and
16  * limitations under the License.
17  *-----
18  * For more information about the OpenAirInterface (OAI) Software Alliance:
19  *     contact@openairinterface.org
20  */
21
22 #ifndef _BENETEL_4G_LOW_H_
23 #define _BENETEL_4G_LOW_H_
24
25 #include "shared_buffers.h"
26
27 typedef struct {
28     shared_buffers *buffers;
29     /* [2] is for two antennas */
30     int         next_subframe[2];
31     int         next_symbol[2];
32     int         expected_benetel_frame[2];
33     char        *dpdk_main_command_line;
34 } benetel_t;
35
36 typedef struct {
37     int frame;
38     int subframe;
39     int slot;
40     int symbol;
41     int antenna;
42     unsigned char iq[4800];
43 } ul_packet_t;
44
45 void *benetel_start(char *ifname, shared_buffers *buffers, char
46                      *dpdk_main_command_line);
47 void store_ul(benetel_t *bs, ul_packet_t *ul);
48 void store_prach(benetel_t *bs, int frame, int subframe, void *data);
49
50#endif /* _BENETEL_4G_LOW_H_ */
```

low_dpdk.c File Reference

```
#include <sched.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include "low.h"
```

Include dependency graph for low_dpdk.c:



Macros

```
#define _GNU_SOURCE
```

Functions

```
int dpdk_main (int argc, char **argv, benetel_t *)
void * dpdk_thread (void * _bs)
void * benetel_start_dpdk (char *ifname, shared_buffers *buffers, char
    *dpdk_main_command_line)
```

Macro Definition Documentation

◆ **_GNU_SOURCE**

```
#define _GNU_SOURCE
```

Function Documentation

◆ benetel_start_dpdk()

```
void * benetel_start_dpdk( char * ifname,  
                           shared_buffers * buffers,  
                           char * dpdk_main_command_line )
```

◆ dpdk_main()

```
int dpdk_main( int argc,  
               char ** argv,  
               benetel_t * bs )
```

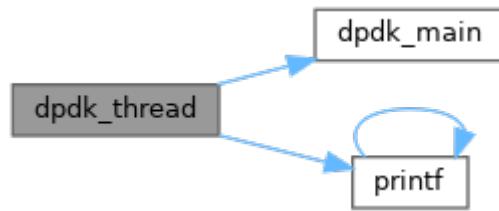
Here is the caller graph for this function:



◆ dpdk_thread()

```
void * dpdk_thread ( void * _bs )
```

Here is the call graph for this function:



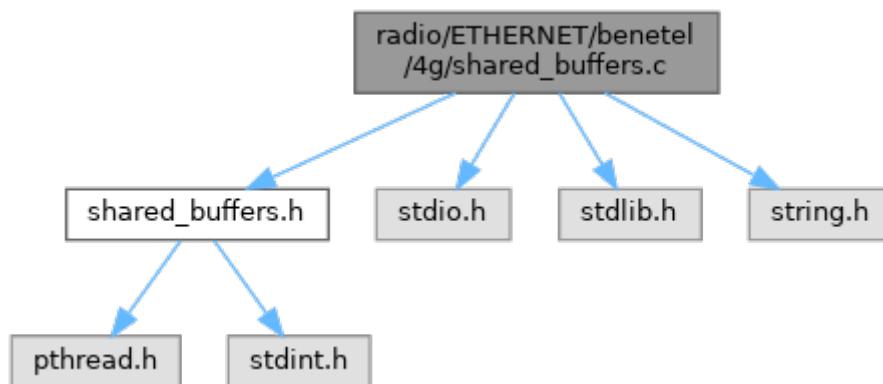
Here is the caller graph for this function:



shared_buffers.c File Reference

```
#include "shared_buffers.h"  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>
```

Include dependency graph for shared_buffers.c:



Functions

```
void init_buffers (shared_buffers *s)  
void lock_dl_buffer (shared_buffers *s, int subframe)  
void unlock_dl_buffer (shared_buffers *s, int subframe)  
void wait_dl_buffer (shared_buffers *s, int subframe)  
void signal_dl_buffer (shared_buffers *s, int subframe)  
void lock_ul_buffer (shared_buffers *s, int subframe)  
void unlock_ul_buffer (shared_buffers *s, int subframe)  
void wait_ul_buffer (shared_buffers *s, int subframe)  
void signal_ul_buffer (shared_buffers *s, int subframe)
```

Function Documentation

◆ **init_buffers()**

```
void init_buffers ( shared_buffers * s )
```

Here is the call graph for this function:



Here is the caller graph for this function:



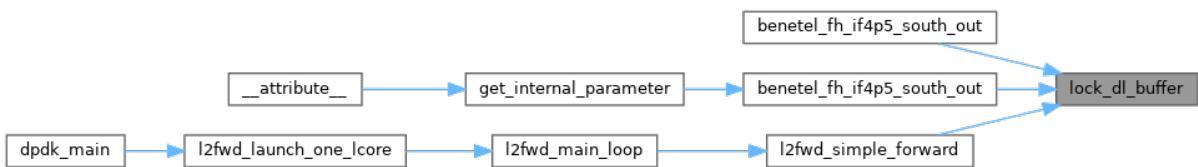
◆ lock_dl_buffer()

```
void lock_dl_buffer ( shared_buffers * s, int subframe )
```

Here is the call graph for this function:



Here is the caller graph for this function:



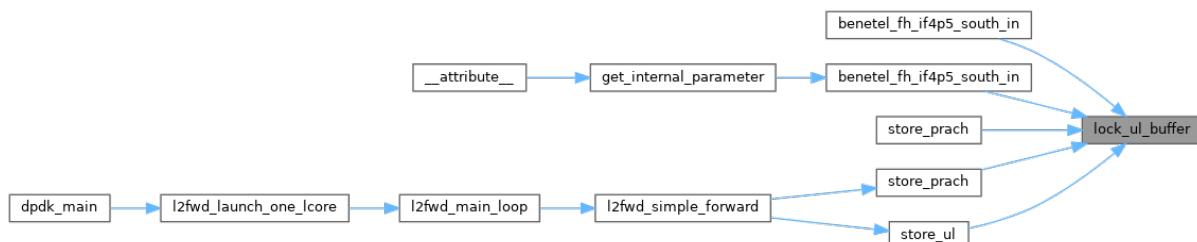
◆ lock_ul_buffer()

void lock_ul_buffer	(shared_buffers *	s,
	int		subframe)

Here is the call graph for this function:



Here is the caller graph for this function:



◆ signal_dl_buffer()

void signal_dl_buffer	(shared_buffers *	s,
	int		subframe)

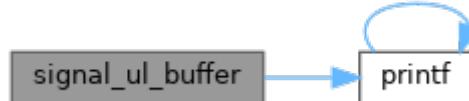
Here is the call graph for this function:



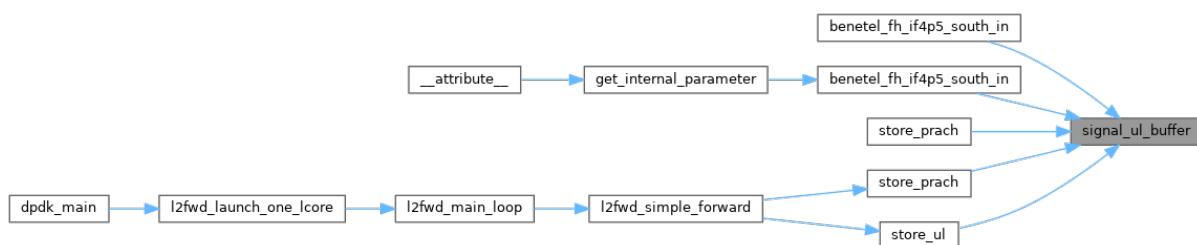
◆ signal_ul_buffer()

```
void signal_ul_buffer ( shared_buffers * s,
int subframe )
```

Here is the call graph for this function:



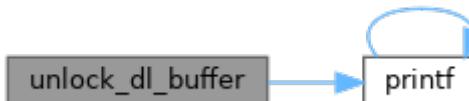
Here is the caller graph for this function:



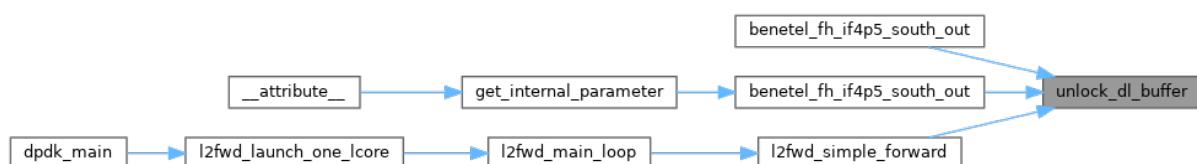
◆ unlock_dl_buffer()

```
void unlock_dl_buffer ( shared_buffers * s,
int subframe )
```

Here is the call graph for this function:



Here is the caller graph for this function:



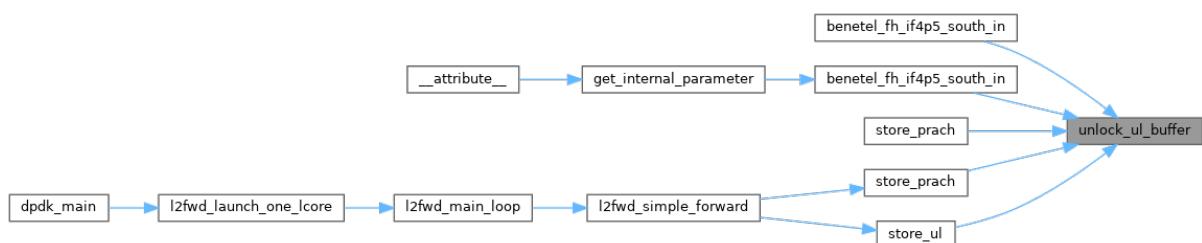
◆ unlock_ul_buffer()

```
void unlock_ul_buffer ( shared_buffers * s,
                      int subframe )
```

Here is the call graph for this function:



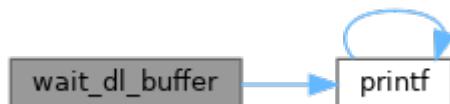
Here is the caller graph for this function:



◆ wait_dl_buffer()

```
void wait_dl_buffer ( shared_buffers * s,
                      int subframe )
```

Here is the call graph for this function:



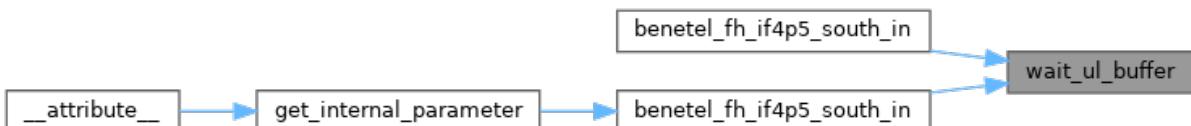
◆ wait_ul_buffer()

```
void wait_ul_buffer ( shared_buffers * s,  
                      int subframe )
```

Here is the call graph for this function:



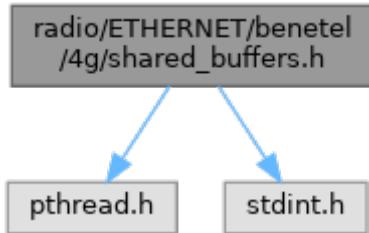
Here is the caller graph for this function:



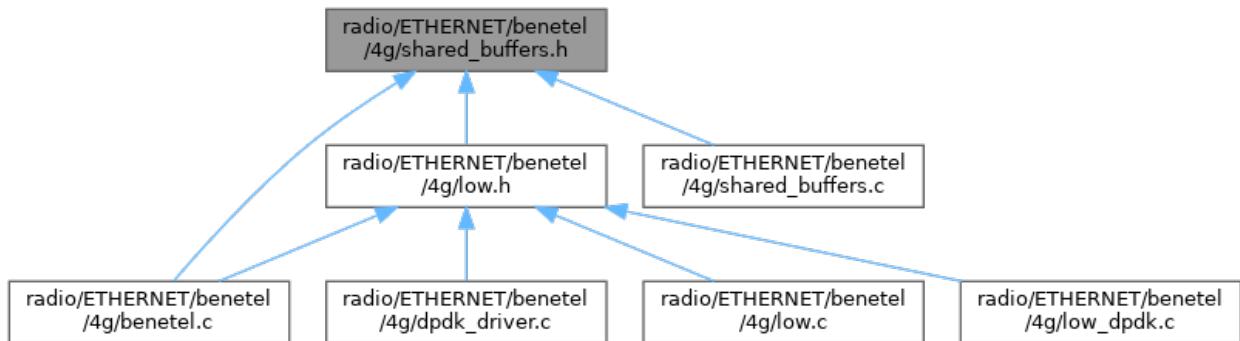
shared_buffers.h File Reference

```
#include <pthread.h>
#include <stdint.h>
```

Include dependency graph for shared_buffers.h:



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



Go to the source code of this file.

Classes

struct **shared_buffers**

Functions

```
void init_buffers (shared_buffers *s)
void lock_dl_buffer (shared_buffers *s, int subframe)
void unlock_dl_buffer (shared_buffers *s, int subframe)
void wait_dl_buffer (shared_buffers *s, int subframe)
void signal_dl_buffer (shared_buffers *s, int subframe)
void lock_ul_buffer (shared_buffers *s, int subframe)
void unlock_ul_buffer (shared_buffers *s, int subframe)
void wait_ul_buffer (shared_buffers *s, int subframe)
void signal_ul_buffer (shared_buffers *s, int subframe)
```

Function Documentation

◆ init_buffers()

```
void init_buffers ( shared_buffers * s )
```

◆ lock_dl_buffer()

```
void lock_dl_buffer ( shared_buffers * s,
                      int subframe )
```

◆ lock_ul_buffer()

```
void lock_ul_buffer ( shared_buffers * s,  
                      int subframe )
```

◆ signal_dl_buffer()

```
void signal_dl_buffer ( shared_buffers * s,  
                      int subframe )
```

◆ signal_ul_buffer()

```
void signal_ul_buffer ( shared_buffers * s,  
                      int subframe )
```

◆ unlock_dl_buffer()

```
void unlock_dl_buffer ( shared_buffers * s,  
                      int subframe )
```

◆ unlock_ul_buffer()

```
void unlock_ul_buffer ( shared_buffers * s,  
                      int subframe )
```

◆ wait_dl_buffer()

```
void wait_dl_buffer ( shared_buffers * s,  
                      int subframe )
```

◆ wait_ul_buffer()

```
void wait_ul_buffer ( shared_buffers * s,  
                      int subframe )
```

shared_buffers.h

Go to the documentation of this file.

```
1  /*
2   * Licensed to the OpenAirInterface (OAI) Software Alliance under one or more
3   * contributor license agreements. See the NOTICE file distributed with
4   * this work for additional information regarding copyright ownership.
5   * The OpenAirInterface Software Alliance licenses this file to You under
6   * the OAI Public License, Version 1.1 (the "License"); you may not use this file
7   * except in compliance with the License.
8   * You may obtain a copy of the License at
9   *
10  *     http://www.openairinterface.org/?page_id=698
11  *
12  * Unless required by applicable law or agreed to in writing, software
13  * distributed under the License is distributed on an "AS IS" BASIS,
14  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15  * See the License for the specific language governing permissions and
16  * limitations under the License.
17  *-----
18  * For more information about the OpenAirInterface (OAI) Software Alliance:
19  *     contact@openairinterface.org
20  */
21
22 #ifndef _BENETEL_4G_SHARED_BUFFERS_H_
23 #define _BENETEL_4G_SHARED_BUFFERS_H_
24
25 #include <pthread.h>
26 #include <stdint.h>
27
28 typedef struct {
29     /* [2] is for two antennas */
30     unsigned char dl[2][10][14*1200*4];
31     unsigned char ul[2][10][14*1200*4];
32     uint16_t dl_busy[2][10];
33     uint16_t ul_busy[2][10];
34
35     pthread_mutex_t m_ul[10];
36     pthread_cond_t c_ul[10];
37
38     pthread_mutex_t m_dl[10];
39     pthread_cond_t c_dl[10];
40
41     unsigned char prach[10][849*4];
42     unsigned char prach_busy[10];
43
44     /* statistics/error counting */
45     int ul_overflow;
46     int dl_underflow;
47 } shared_buffers;
48
49 void init_buffers(shared_buffers *s);
50
51 void lock_dl_buffer(shared_buffers *s, int subframe);
52 void unlock_dl_buffer(shared_buffers *s, int subframe);
53 void wait_dl_buffer(shared_buffers *s, int subframe);
54 void signal_dl_buffer(shared_buffers *s, int subframe);
55
```

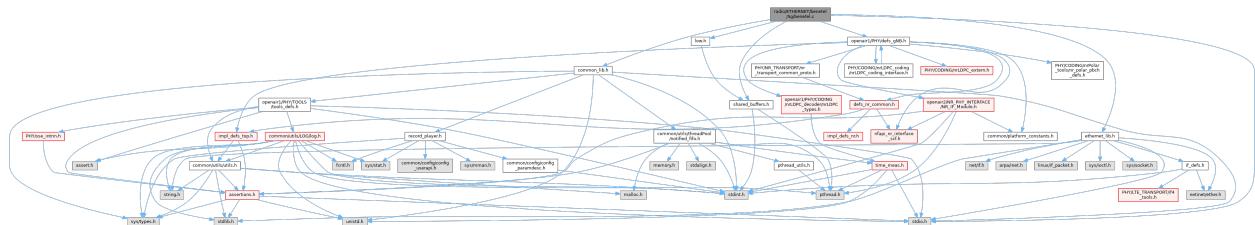
```
56 void lock_ul_buffer(shared_buffers *s, int subframe);
57 void unlock_ul_buffer(shared_buffers *s, int subframe);
58 void wait_ul_buffer(shared_buffers *s, int subframe);
59 void signal_ul_buffer(shared_buffers *s, int subframe);
60
61 #endif /* _BENETEL_4G_SHARED_BUFFERS_H_ */
```

doc/5Gnas.md File Reference

benetel.c File Reference

```
#include <stdio.h>
#include "common_lib.h"
#include "ethernet_lib.h"
#include "shared_buffers.h"
#include "low.h"
#include "openair1/PHY/defs_gNB.h"
```

Include dependency graph for benetel.c:



Classes

struct **benetel_eth_state_t**

Functions

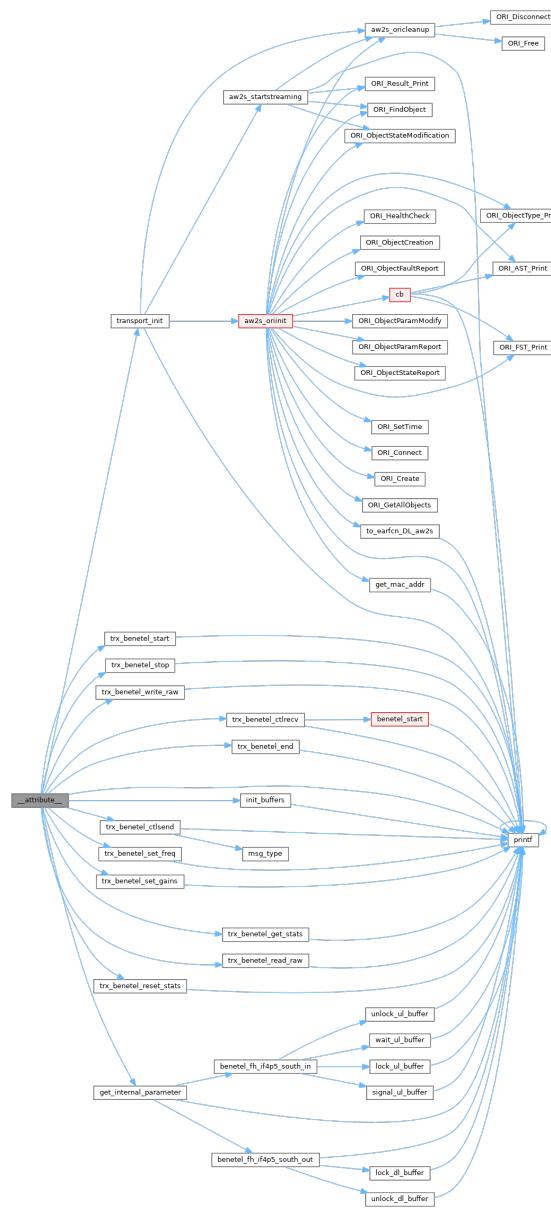
```
int trx_benetel_start (openair0_device *device)
void trx_benetel_end (openair0_device *device)
int trx_benetel_stop (openair0_device *device)
int trx_benetel_set_freq (openair0_device *device, openair0_config_t
*openair0_cfg)
int trx_benetel_set_gains (openair0_device *device, openair0_config_t
*openair0_cfg)
int trx_benetel_get_stats (openair0_device *device)
int trx_benetel_reset_stats (openair0_device *device)
int ethernet_tune (openair0_device *device, unsigned int option, int value)
    this function allows you to configure certain ethernet parameters in socket or
    device level
int trx_benetel_write_raw (openair0_device *device, openair0_timestamp
    timestamp, void **buff, int nsamps, int cc, int flags)
int trx_benetel_read_raw (openair0_device *device, openair0_timestamp
    *timestamp, void **buff, int nsamps, int cc)
char * msg_type (int t)
int trx_benetel_ctlsend (openair0_device *device, void *msg, ssize_t msg_len)
int trx_benetel_ctlrecv (openair0_device *device, void *msg, ssize_t msg_len)
void benetel_fh_if4p5_south_in (RU_t *ru, int *frame, int *slot)
void benetel_fh_if4p5_south_out (RU_t *ru, int frame, int slot, uint64_t
    timestamp)
void * get_internal_parameter (char *name)
    _attribute_ ((_visibility_("default")))
```

Function Documentation

◆ **_attribute_()**

```
_attribute_ ( __visibility__("default") )
```

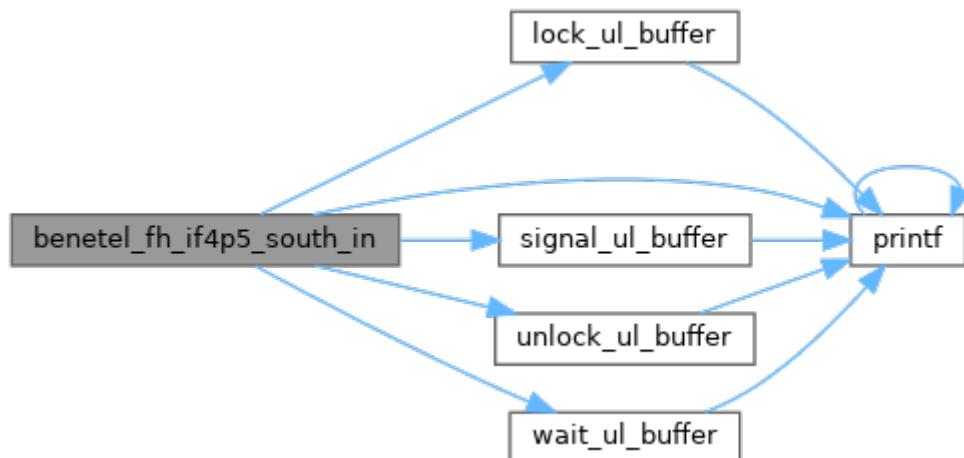
Here is the call graph for this function:



◆ **benetel_fh_if4p5_south_in()**

```
void benetel_fh_if4p5_south_in ( RU_t * ru,
                                 int * frame,
                                 int * slot )
```

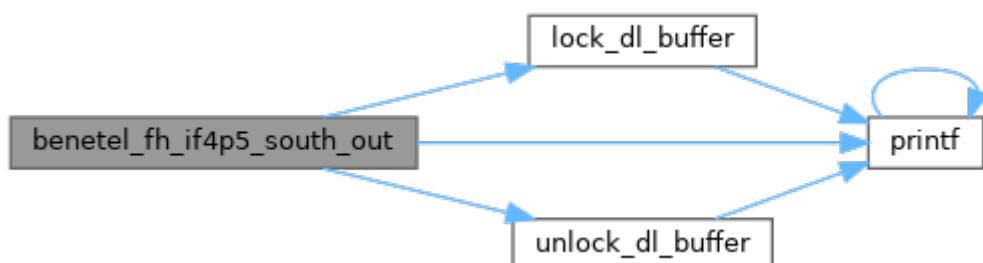
Here is the call graph for this function:



◆ benetel_fh_if4p5_south_out()

```
void benetel_fh_if4p5_south_out ( RU_t * ru,
                                   int frame,
                                   int slot,
                                   uint64_t timestamp )
```

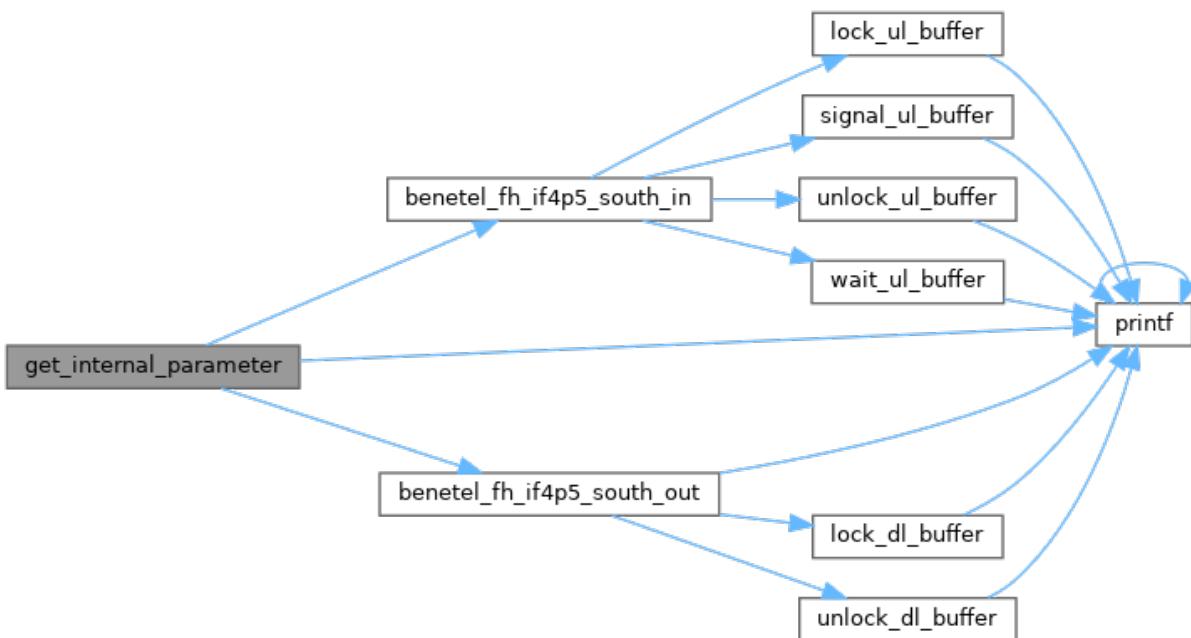
Here is the call graph for this function:



◆ get_internal_parameter()

```
void * get_internal_parameter ( char * name )
```

Here is the call graph for this function:



◆ msg_type()

```
char * msg_type ( int t )
```

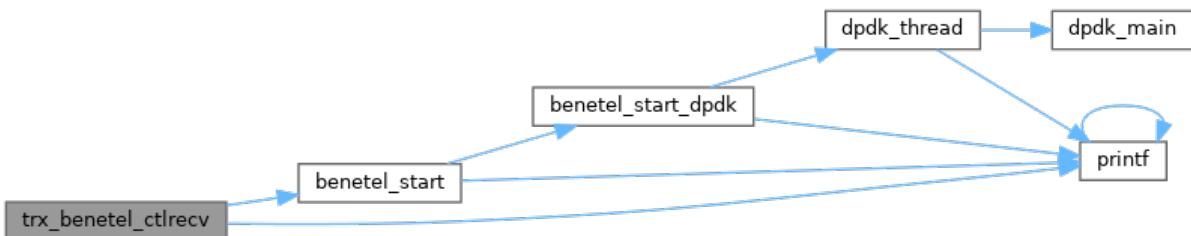
◆ trx_benetel_ctlrecv()

```

int trx_benetel_ctlrecv ( openair0_device * device,
                          void * msg,
                          ssize_t msg_len )

```

Here is the call graph for this function:



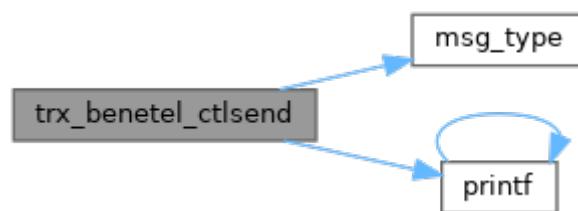
◆ `trx_benetel_ctlsend()`

```

int trx_benetel_ctlsend ( openair0_device * device,
                          void * msg,
                          ssize_t msg_len )

```

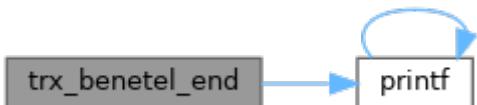
Here is the call graph for this function:



◆ `trx_benetel_end()`

```
void trx_benetel_end ( openair0_device * device )
```

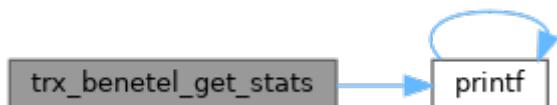
Here is the call graph for this function:



◆ `trx_benetel_get_stats()`

```
int trx_benetel_get_stats ( openair0_device * device )
```

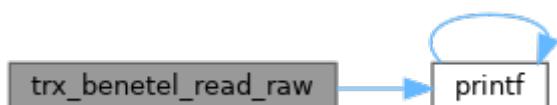
Here is the call graph for this function:



◆ `trx_benetel_read_raw()`

```
int trx_benetel_read_raw ( openair0_device * device,
                           openair0_timestamp * timestamp,
                           void ** buff,
                           int nsamps,
                           int cc )
```

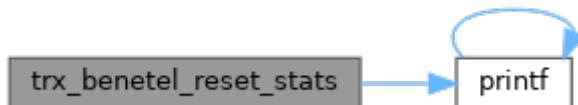
Here is the call graph for this function:



◆ `trx_benetel_reset_stats()`

```
int trx_benetel_reset_stats ( openair0_device * device )
```

Here is the call graph for this function:



◆ `trx_benetel_set_freq()`

```
int trx_benetel_set_freq ( openair0_device * device,  
                           openair0_config_t * openair0_cfg )
```

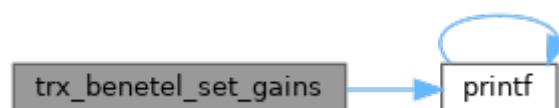
Here is the call graph for this function:



◆ `trx_benetel_set_gains()`

```
int trx_benetel_set_gains ( openair0_device * device,  
                           openair0_config_t * openair0_cfg )
```

Here is the call graph for this function:



◆ `trx_benetel_start()`

```
int trx_benetel_start ( openair0_device * device )
```

Here is the call graph for this function:



◆ **trx_benetel_stop()**

```
int trx_benetel_stop ( openair0_device * device )
```

Here is the call graph for this function:



◆ **trx_benetel_write_raw()**

int trx_benetel_write_raw	(openair0_device *	device,
		openair0_timestamp	timestamp,
		void **	buff,
		int	nsamps,
		int	cc,
		int	flags)

Here is the call graph for this function:

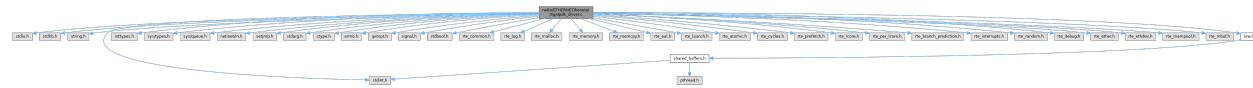


dpdk_driver.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <inttypes.h>
#include <sys/types.h>
#include <sys/queue.h>
#include <netinet/in.h>
#include <setjmp.h>
#include <stdarg.h>
#include <ctype.h>
#include <errno.h>
#include <getopt.h>
#include <signal.h>
#include <stdbool.h>
#include <rte_common.h>
#include <rte_log.h>
#include <rte_malloc.h>
#include <rte_memory.h>
#include <rte_memcpy.h>
#include <rte_eal.h>
#include <rte_launch.h>
#include <rte_atomic.h>
#include <rte_cycles.h>
#include <rte_prefetch.h>
#include <rte_lcore.h>
#include <rte_per_lcore.h>
#include <rte_branch_prediction.h>
#include <rte_interrupts.h>
#include <rte_random.h>
#include <rte_debug.h>
#include <rte_ether.h>
```

```
#include <rte_ethdev.h>  
#include <rte_mempool.h>  
#include <rte_mbuf.h>  
#include "low.h"
```

Include dependency graph for dpdk_driver.c:



Classes

```
struct lcore_queue_conf  
struct l2fwd_port_statistics
```

Macros

```
#define ANT_NUM buf[23]
#define PAYLOAD_1 buf[20]
#define PAYLOAD_2 buf[21]
#define ETH_TYPE buf[17]
#define SYMBOL buf[29]
#define SUBFRAME buf[28]
#define FRAME buf[27]
#define RTE_LOGTYPE_L2FWD RTE_LOGTYPE_USER1
#define MAX_PKT_BURST 32
#define BURST_TX_DRAIN_US 5 /* TX drain every ~100us */
#define MEMPOOL_CACHE_SIZE 256
#define RTE_TEST_RX_DESC_DEFAULT 1024
#define RTE_TEST_TX_DESC_DEFAULT 1024
#define MAX_RX_QUEUE_PER_LCORE 16
#define MAX_TX_QUEUE_PER_PORT 16
#define MAX_TIMER_PERIOD 86400 /* 1 day max */
#define CMD_LINE_OPT_MAC_UPDATING "mac-updating"
#define CMD_LINE_OPT_NO_MAC_UPDATING "no-mac-updating"
#define CHECK_INTERVAL 100 /* 100ms */
#define MAX_CHECK_TIME 90 /* 9s (90 * 100ms) in total */
```

Enumerations

```
enum { CMD_LINE_OPT_MIN_NUM = 256 }
```

Functions

```
static void print_stats (void)
static void I2fwd_simple_forward (struct rte_mbuf *m, unsigned portid,
                                  benetel_t *bs)
static void I2fwd_main_loop (benetel_t *bs)
static int I2fwd_launch_one_lcore (void *bs)
static void I2fwd_usage (const char *prgname)
static int I2fwd_parse_portmask (const char *portmask)
static unsigned int I2fwd_parse_nqueue (const char *q_arg)
static int I2fwd_parse_timer_period (const char *q_arg)
static int I2fwd_parse_args (int argc, char **argv)
static void check_all_ports_link_status (uint32_t port_mask)
static void signal_handler (int signum)
int dpdk_main (int argc, char **argv, benetel_t *bs)
```

Variables

```
static volatile bool force_quit
unsigned short iq [712320]
unsigned char iq_swap [1424640]
unsigned short * iq_ptr [14]
unsigned int dl_start = 0
unsigned int slot_id_ctrl = 0
unsigned int count_symbol = 0
unsigned int sf = 0x10
static int mac_updating = 1
static uint16_t nb_rxd = RTE_TEST_RX_DESC_DEFAULT
static uint16_t nb_txd = RTE_TEST_TX_DESC_DEFAULT
static struct ether_addr I2fwd_ports_eth_addr [RTE_MAX_ETHPORTS]
static uint32_t I2fwd_enabled_port_mask = 0
static uint32_t I2fwd_dst_ports [RTE_MAX_ETHPORTS]
static unsigned int I2fwd_rx_queue_per_lcore = 1
struct lcore_queue_conf _rte_cache_aligned
struct lcore_queue_conf lcore_queue_conf [RTE_MAX_LCORE]
static struct rte_eth_dev_tx_buffer * tx_buffer [RTE_MAX_ETHPORTS]
static struct rte_eth_conf port_conf
struct rte_mempool * I2fwd_pktmbuf_pool = NULL
struct I2fwd_port_statistics port_statistics [RTE_MAX_ETHPORTS]
static uint64_t timer_period = 10
static const char short_options []
static const struct option lgopts []
```

Macro Definition Documentation

◆ ANT_NUM

```
#define ANT_NUM buf[23]
```

◆ BURST_TX_DRAIN_US

```
#define BURST_TX_DRAIN_US 5 /* TX drain every ~100us */
```

◆ CHECK_INTERVAL

```
#define CHECK_INTERVAL 100 /* 100ms */
```

◆ CMD_LINE_OPT_MAC_UPDATING

```
#define CMD_LINE_OPT_MAC_UPDATING "mac-updating"
```

◆ CMD_LINE_OPT_NO_MAC_UPDATING

```
#define CMD_LINE_OPT_NO_MAC_UPDATING "no-mac-updating"
```

◆ ETH_TYPE

```
#define ETH_TYPE buf[17]
```

◆ FRAME

```
#define FRAME buf[27]
```

◆ MAX_CHECK_TIME

```
#define MAX_CHECK_TIME 90 /* 9s (90 * 100ms) in total */
```

◆ MAX_PKT_BURST

```
#define MAX_PKT_BURST 32
```

◆ MAX_RX_QUEUE_PER_LCORE

```
#define MAX_RX_QUEUE_PER_LCORE 16
```

◆ MAX_TIMER_PERIOD

```
#define MAX_TIMER_PERIOD 86400 /* 1 day max */
```

◆ MAX_TX_QUEUE_PER_PORT

```
#define MAX_TX_QUEUE_PER_PORT 16
```

◆ MEMPOOL_CACHE_SIZE

```
#define MEMPOOL_CACHE_SIZE 256
```

◆ PAYLOAD_1

```
#define PAYLOAD_1 buf[20]
```

◆ PAYLOAD_2

```
#define PAYLOAD_2 buf[21]
```

◆ RTE_LOGTYPE_L2FWD

```
#define RTE_LOGTYPE_L2FWD RTE_LOGTYPE_USER1
```

◆ RTE_TEST_RX_DESC_DEFAULT

```
#define RTE_TEST_RX_DESC_DEFAULT 1024
```

◆ RTE_TEST_TX_DESC_DEFAULT

```
#define RTE_TEST_TX_DESC_DEFAULT 1024
```

◆ SUBFRAME

```
#define SUBFRAME buf[28]
```

◆ SYMBOL

```
#define SYMBOL buf[29]
```

Enumeration Type Documentation

◆ anonymous enum

anonymous enum

Enumerator

CMD_LINE_OPT_MIN_NUM

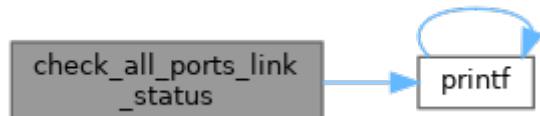
Function Documentation

◆ check_all_ports_link_status()

```
static void check_all_ports_link_status ( uint32_t port_mask )
```

static

Here is the call graph for this function:



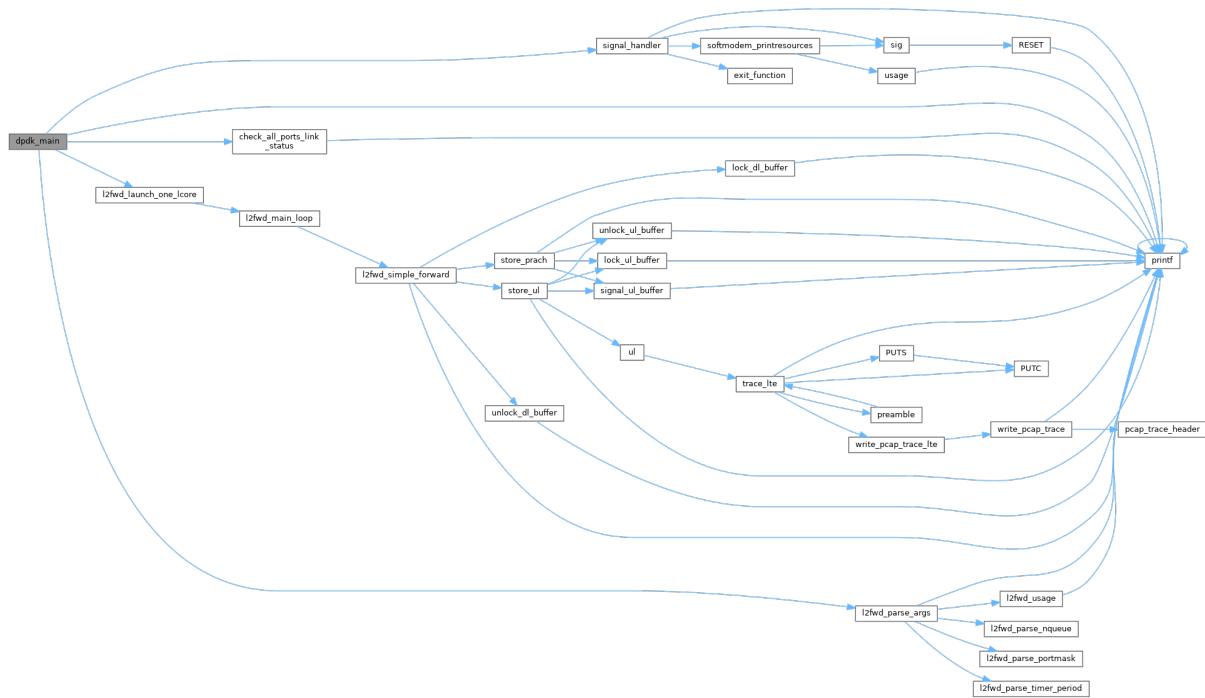
◆ dpdk_main()

```

int dpdk_main
(
    int argc,
    char **argv,
    benetel_t *bs
)

```

Here is the call graph for this function:



Here is the caller graph for this function:



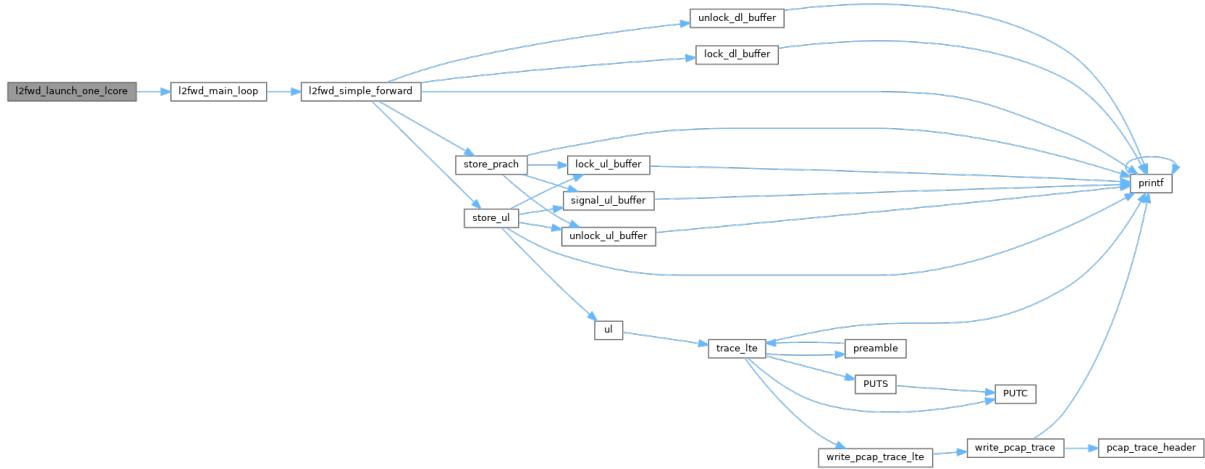
◆ l2fwd_launch_one_lcore()

```
static int l2fwd_launch_one_lcore
```

```
( void * bs )
```

```
static
```

Here is the call graph for this function:



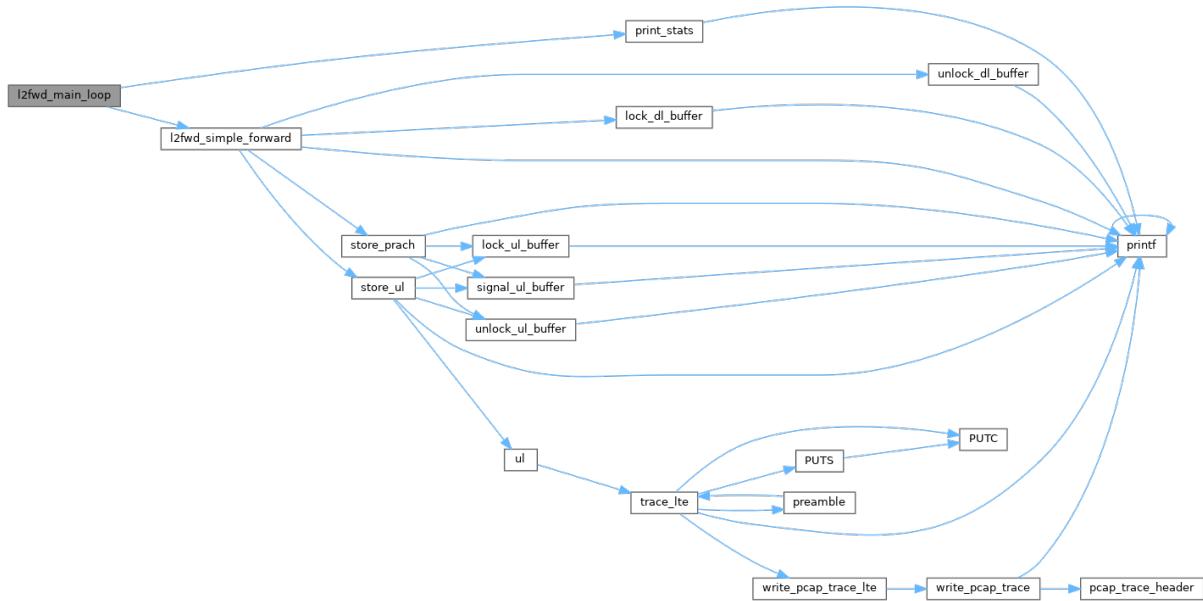
◆ l2fwd_main_loop()

```
static void l2fwd_main_loop
```

```
( benetel_t * bs )
```

```
static
```

Here is the call graph for this function:

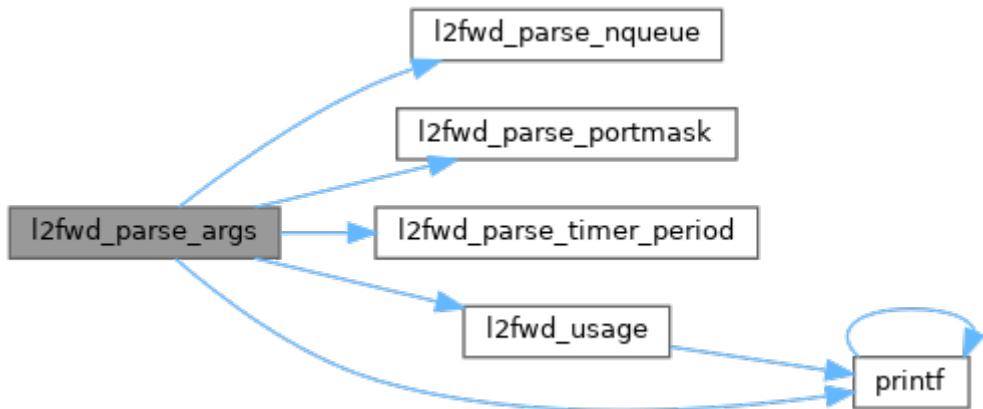


◆ l2fwd_parse_args()

```
static int l2fwd_parse_args ( int argc,  
                           char ** argv )
```

static

Here is the call graph for this function:



◆ l2fwd_parse_nqueue()

```
static unsigned int l2fwd_parse_nqueue ( const char * q_arg )
```

static

◆ l2fwd_parse_portmask()

```
static int l2fwd_parse_portmask ( const char * portmask )
```

static

◆ l2fwd_parse_timer_period()

```
static int l2fwd_parse_timer_period ( const char * q_arg )
```

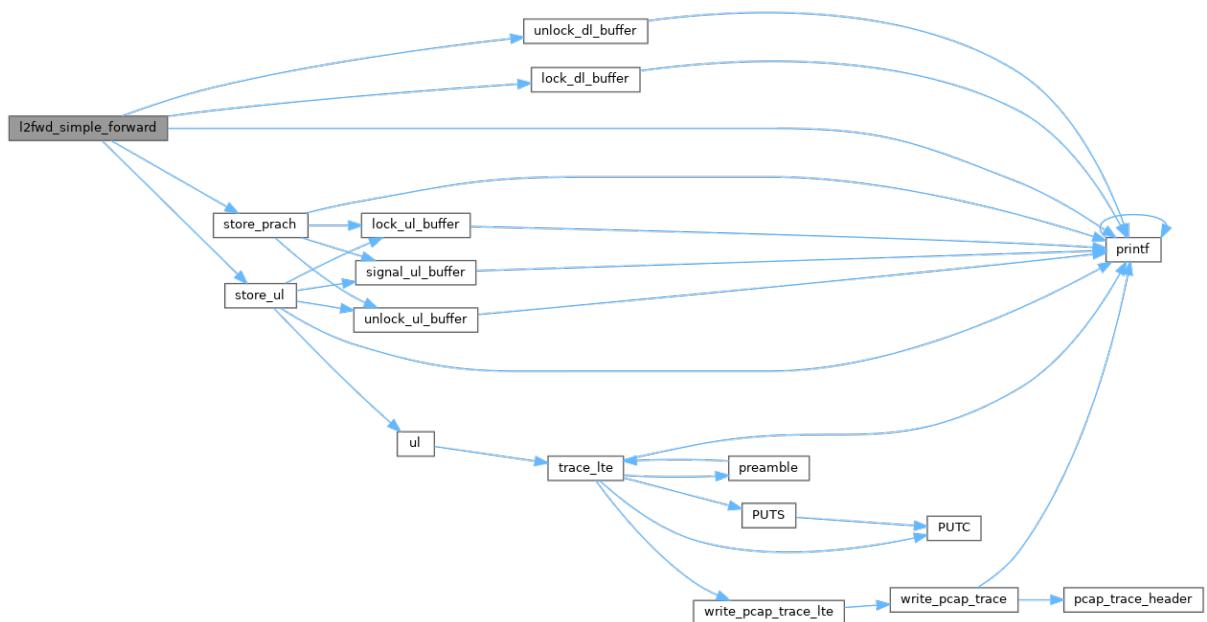
static

◆ l2fwd_simple_forward()

```
static void l2fwd_simple_forward ( struct rte_mbuf * m,  
                                  unsigned portid,  
                                  benetel_t * bs )
```

static

Here is the call graph for this function:



◆ l2fwd_usage()

```
static void l2fwd_usage ( const char * prgname )
```

static

Here is the call graph for this function:



◆ print_stats()

```
static void print_stats ( void )
```

static

Here is the call graph for this function:



◆ signal_handler()

```
static void signal_handler ( int signum )
```

static

Here is the call graph for this function:



Variable Documentation

◆ _rte_cache_aligned

```
struct l2fwd_port_statistics _rte_cache_aligned
```

◆ count_symbol

```
unsigned int count_symbol = 0
```

◆ dl_start

```
unsigned int dl_start = 0
```

◆ force_quit

volatile bool force_quit

static

◆ iq

unsigned short iq[712320]

◆ iq_ptr

unsigned short* iq_ptr[14]

◆ iq_swap

unsigned char iq_swap[1424640]

◆ l2fwd_dst_ports

uint32_t l2fwd_dst_ports[RTE_MAX_ETHPORTS]

static

◆ l2fwd_enabled_port_mask

```
uint32_t l2fwd_enabled_port_mask = 0
```

static

◆ l2fwd_pktmbuf_pool

```
struct rte_mempool* l2fwd_pktmbuf_pool = NULL
```

◆ l2fwd_ports_eth_addr

```
struct ether_addr l2fwd_ports_eth_addr[RTE_MAX_ETHPORTS]
```

static

◆ l2fwd_rx_queue_per_lcore

```
unsigned int l2fwd_rx_queue_per_lcore = 1
```

static

◆ lcore_queue_conf

```
struct lcore_queue_conf lcore_queue_conf[RTE_MAX_LCORE]
```

◆ lgopts

```
const struct option lgopts[]
```

static

Initial value:

```
= {  
    { CMD_LINE_OPT_MAC_UPDATING, no_argument, &mac_updating, 1},  
    { CMD_LINE_OPT_NO_MAC_UPDATING, no_argument, &mac_updating, 0},  
    {NULL, 0, 0, 0}  
}
```

◆ mac_updating

```
int mac_updating = 1
```

static

◆ nb_rxd

```
uint16_t nb_rxd = RTE_TEST_RX_DESC_DEFAULT
```

static

◆ nb_txd

```
uint16_t nb_txd = RTE_TEST_TX_DESC_DEFAULT
```

static

◆ port_conf

```
struct rte_eth_conf port_conf
```

static

Initial value:

```
= {  
    .rxmode = {  
        .split_hdr_size = 0,  
        .offloads      = DEV_RX_OFFLOAD_JUMBO_FRAME,  
        .split_hdr_size = 0,  
        .max_rx_pkt_len = 9500,  
    },  
    .txmode = {  
        .mq_mode = ETH_MQ_TX_NONE,  
    },  
}
```

◆ port_statistics

```
struct l2fwd_port_statistics port_statistics[RTE_MAX_ETHPORTS]
```

◆ sf

```
unsigned int sf = 0x10
```

◆ short_options

```
const char short_options[]
```

static

Initial value:

```
=  
"p:"  
"q:"  
"T:"
```

◆ slot_id_ctrl

```
unsigned int slot_id_ctrl = 0
```

◆ timer_period

```
uint64_t timer_period = 10
```

static

◆ tx_buffer

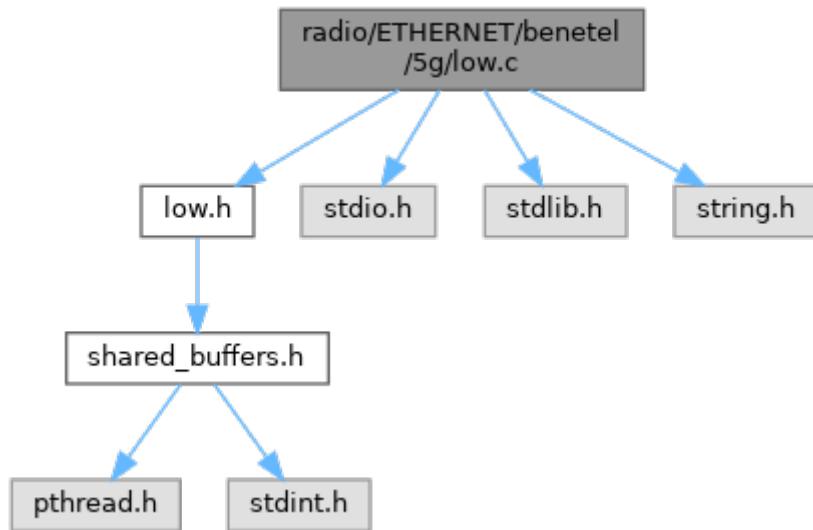
```
struct rte_eth_dev_tx_buffer* tx_buffer[RTE_MAX_ETHPORTS]
```

static

low.c File Reference

```
#include "low.h"  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>
```

Include dependency graph for low.c:



Functions

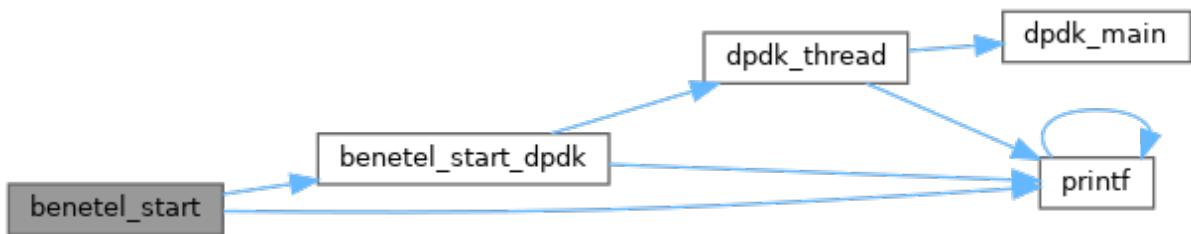
```
void store_ul (benetel_t *bs, ul_packet_t *ul)  
void store_prach (benetel_t *bs, int frame, int slot, void *data)  
void * benetel_start_dpdk (char *ifname, shared_buffers *buffers, char  
*dpdk_main_command_line)  
void * benetel_start (char *ifname, shared_buffers *buffers, char  
*dpdk_main_command_line)
```

Function Documentation

◆ **benetel_start()**

void * benetel_start	(char *	ifname,
		shared_buffers *	buffers,
		char *	dpdk_main_command_line)

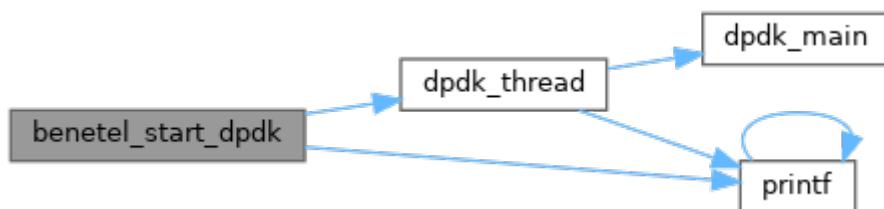
Here is the call graph for this function:



◆ benetel_start_dpdk()

void * benetel_start_dpdk	(char *	ifname,
		shared_buffers *	buffers,
		char *	dpdk_main_command_line)

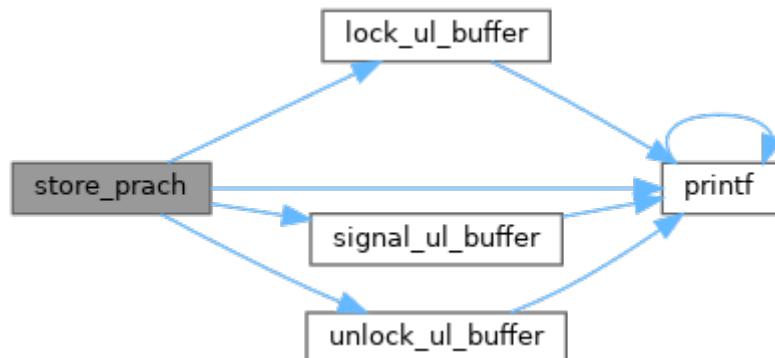
Here is the call graph for this function:



◆ store_prach()

```
void store_prach ( benetel_t *bs,
                    int frame,
                    int slot,
                    void *data )
```

Here is the call graph for this function:



◆ `store_ul()`

```
void store_ul ( benetel_t *bs,
                ul_packet_t *ul )
```

Here is the call graph for this function:

