```
#include <common.h>
```

- assert(x) macro

  > assert (buf->data != NULL);

```
#define assert(x) \
    ({ if (!(x) && _DEBUG) \
        __assert_fail(#x, __FILE__, __LINE__, __func__); })
```

- error(fmt, args…) macro

  > error("failed to write LDO control register: %d", err);

```
#define error(fmt, args...) do {                    \
        printf("ERROR: " pr_fmt(fmt) "\nat %s:%d/%s()\n",    \
            ##args, __FILE__, __LINE__, __func__);      \
} while (0)
```

- BUG() macro

  > BUG();

```
#define BUG() do { \
    printf("BUG: failure at %s:%d/%s()!\n", __FILE__, __LINE__, __FUNCTION__
); \
    panic("BUG!"); \
} while (0)
```

- BUG_ON(condition) macro

  > BUG_ON(ident >= MAX_RESOURCES);

```
#define BUG_ON(condition) do { if (unlikely((condition)!=0)) BUG(); } while(
0)
```

- BUILD_BUG_ON(condition)

  > BUILD_BUG_ON(sizeof(struct ubi_ec_hdr) != 64);

```
#define BUILD_BUG_ON(condition) ((void)sizeof(char[1 - 2*!!(condition)]))
```

- debug(fmt, args…) macro

  > debug("magic: 0x%x\n", sparse_header->magic);

```
#define debug(fmt, args...)          \
    debug_cond(_DEBUG, fmt, ##args)
```