

The 3 macros are from mv3dp LSP(2014). They make me don't worry about whether I should use printk or early_print when kernel initialize.

MY_BUG()

MY_BUG_ON(condition)

MY_DBG_MSG(msg)

MY_BUG() is like **BUG()**.

MY_BUG_ON is like **BUG_ON(condition)**

MY_DBG_MSG(msg) output msg string.

use MY_KERNEL_DEBUG and USE_EARLY_PRINT to control the 3 macros behaviour.

if MY_KERNEL_DEBUG and USE_EARLY_PRINT all defined

output function use early_print(), so the 3 macros could be used before printk could work.

if MY_KERNEL_DEBUG is defined and USE_EARLY_PRINT is not defined

output function use printk().

if MY_KERNEL_DEBUG is not defined.

The 3 macros are all dummy.

=====

```
#ifdef MY_KERNEL_DEBUG
```

```
#ifdef USE_EARLY_PRINT
```

```
#define MY_KERNEL_PRINTK    early_print
```

```

/*

* I borrow the function from arch/arm/kernel/setup.c

*/

extern void __init early_print(const char *str, ...);


#define MY_BUG() do { \

    early_print("BUG: failure at %s:%d/%s()!\n", __FILE__, __LINE__, __func__); \

    dump_stack(); \

    while(1); \

} while (0)


#define MY_BUG_ON(condition) do { if (unlikely(condition)) MY_BUG(); } while(0)


#define MY_DBG_MSG(msg) do { \

    early_print("%s: %s\n", __func__, msg); \

} while(0)


#else    //    !USE_EARLY_PRINT


#define MY_KERNEL_PRINTK    printk


#define MY_BUG() do { \

    printk("BUG: failure at %s:%d/%s()!\n", __FILE__, __LINE__, __func__); \

    dump_stack(); \

    while(1); \

} while (0)


#define MY_BUG_ON(condition) do { if (unlikely(condition)) MY_BUG(); } while(0)


#define MY_DBG_MSG(msg) do { \

```

```
    printk("%s: %s\n", __func__, msg); \
} while(0)
```

```
#endif    //    end USE_EARLY_PRINT
```

```
#else    //    !MY_KERNEL_DEBUG
```

```
#define MY_BUG()
```

```
#define MY_BUG_ON(condition)
```

```
#define MY_DBG_MSG(msg)
```

```
#define MY_KERNEL_PRINTK    dummy_print
```

```
// hope the gcc is smart enough, could make the dummy function null when optimization is enabled.
```

```
static void __init dummy_print(const char *str, ...) {}
```

```
#endif    //    end MY_KERNEL_DEBUG
```

```
#ifdef MY_KERNEL_DEBUG
```

```
#define MY_CODE_FOR_DEBUG(code)    code
```

```
#else
```

```
#define MY_CODE_FOR_DEBUG(code)
```

```
#endif
```