

1. before relocating u-boot

----- (1G boundary)

| |

| |

| |

| |

| |

| |

|-----|

| |

| |

| u-boot |

| |

|-----|<----- 0x0800,0000 (miniloaders load u-boot to here)

| GENERATED_GBL_DATA_SIZE |

|-----|<----- 0x07ff,ff50

| Global data (0xa8 bytes) |

|-----|<----- 0x07ff,fea8 register volatile gd_t *gd asm ("r9"), gd_t *gd所指向的
space

| |

| stack |

| |

| |

| |

Question:

这里GENERATED_GBL_DATA_SIZE region好像完全没必要。它的大小就是sizeof(gd_t) (取整后)。之所以有它，是因为

in include/configs/pegmatite.h

```
#define CONFIG_SYS_INIT_SP_ADDR (CONFIG_SYS_BASE_ADDR +  
CONFIG_SYS_INIT_SP_SIZE - GENERATED_GBL_DATA_SIZE)
```

我觉得好像这里减去GENERATED_GBL_DATA_SIZE是没必要的，正确的应该应该是

```
#define CONFIG_SYS_INIT_SP_ADDR (CONFIG_SYS_BASE_ADDR +  
CONFIG_SYS_INIT_SP_SIZE)
```

因为gd_t structure space在crt0.S中已经预留了.

```
1.  #if defined(CONFIG_SPL_BUILD) && defined(CONFIG_SPL_STACK)  
2.      ldr    sp, =(CONFIG_SPL_STACK)  
3.  #else  
4.      ldr    sp, =(CONFIG_SYS_INIT_SP_ADDR)  
5.  #endif  
6.      bic    sp, sp, #7      /* 8-byte alignment for ABI compliance */  
7.      sub    sp, sp, #GD_SIZE /* allocate one GD above SP */  
8.      bic    sp, sp, #7      /* 8-byte alignment for ABI compliance */  
9.      mov    r9, sp          /* GD is above SP */  
10.     mov    r0, #0  
11.     bl     board_init_f
```

尝试一下修改code，看看u-boot运行是否出错。

2. after relocating u-boot

----- (1G boundary)

| |

| 48K (null) |

|-----|<----- 0x3ff,4000

| 16K TLB |

|-----|<----- 0x3ff,0000

| |

| |

| LCD frame buffer |

| (1200K) |

| |

| |

|-----|<----- 0x3fec,4000

||-----||

|| u-boot data (about 4k) ||

||-----||

|| || 256K

|| ||

|| u-boot code (about 252k) ||

|| ||

||-----||

|-----|<----- 0x3fe8,4000

| |

| |

| malloc heap | malloc()申请的空间从这里分配

| |

| 1200K |

| |

|-----|<----- 0x3f68,2000

| Board Info (56 bytes) |

|-----|<----- 0x3f68,1fc8

| Global Data (0xa8 bytes) |

|-----|<----- 0x3f68,1f20, register volatile gd_t *gd asm ("r9"), gd_t *gd所指向的
space

| 16 bytes (gap) |

|-----|<----- 0x3f68,1f10

| |

| stack |

| |

| |

| |

| |

| |

| |

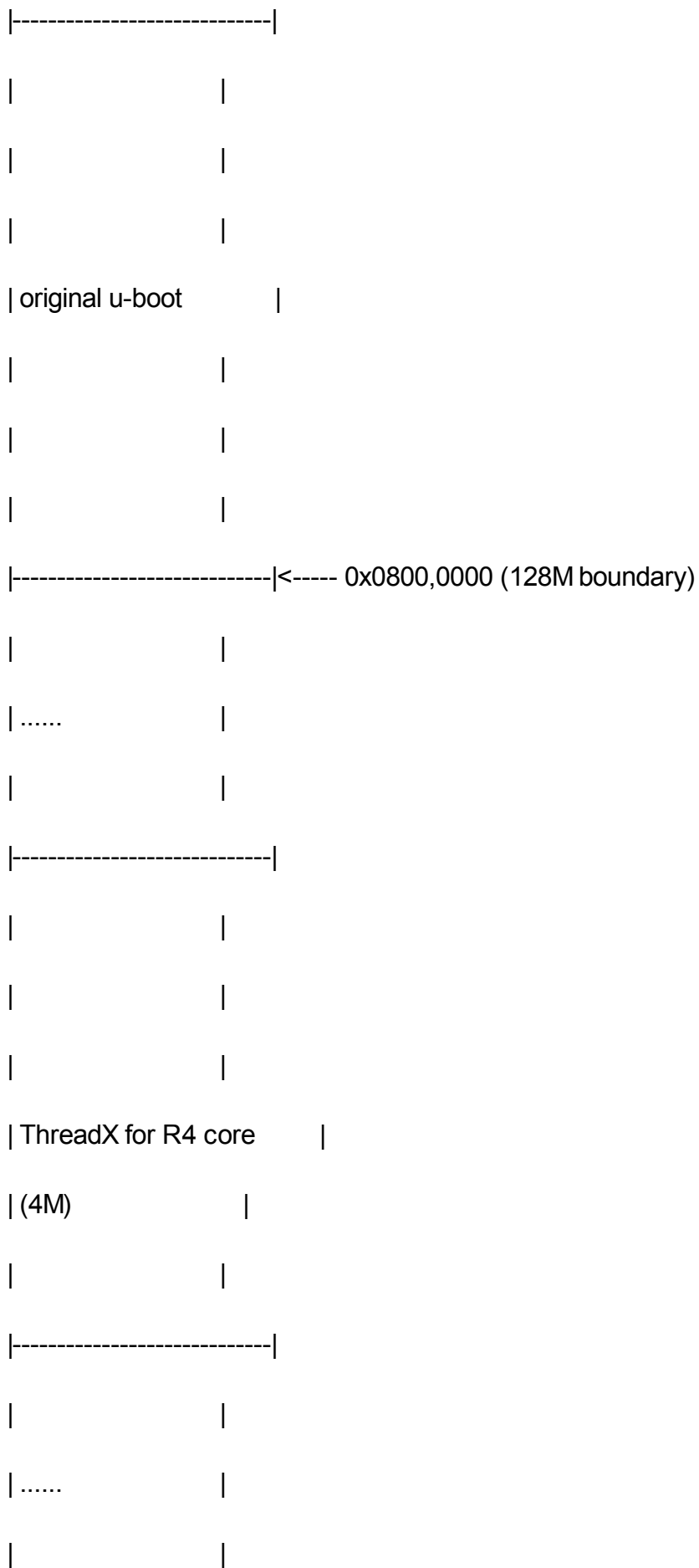
| |

| |

| |

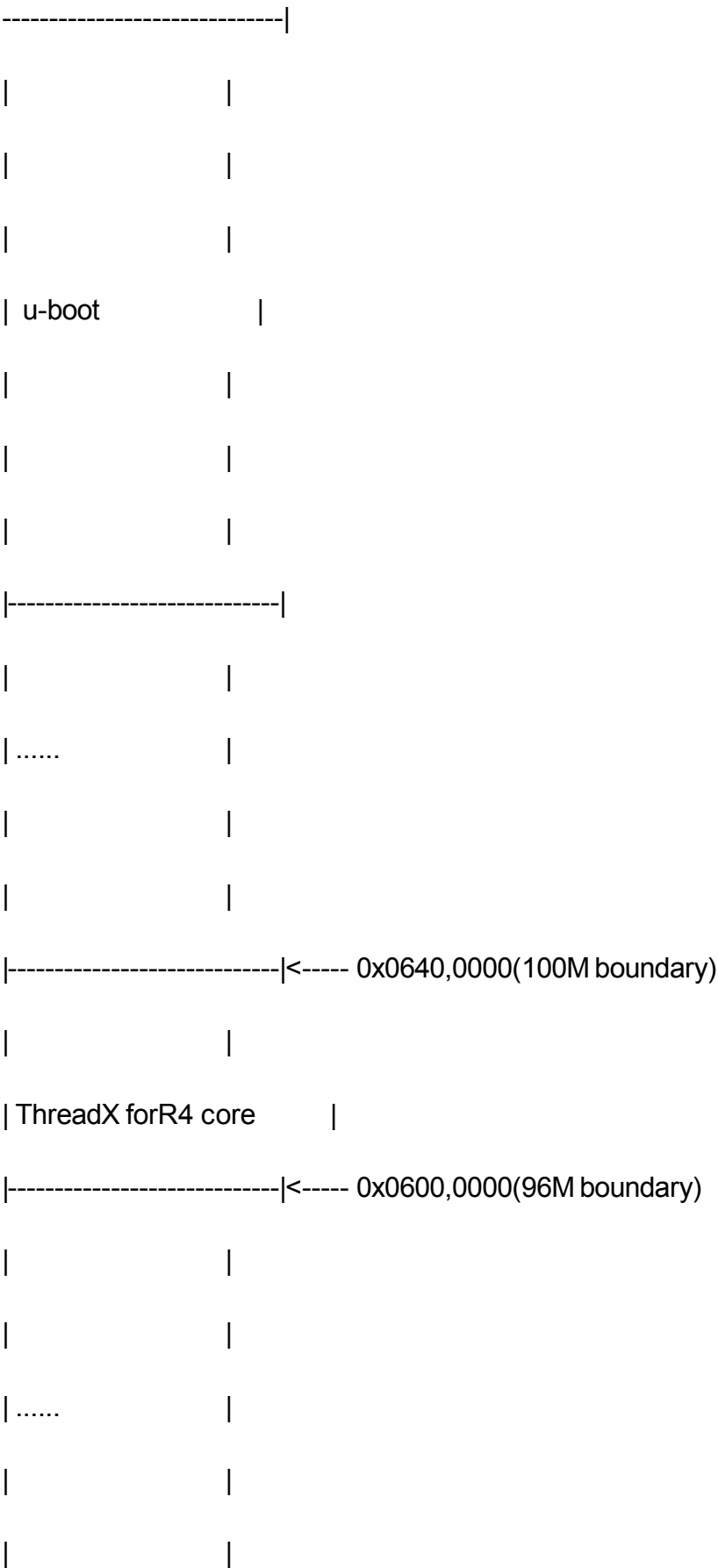
| |

| |



整个u-boot memory layout (after relocating)的地址计算在board_init_f() in arch/arm/lib/board.c。

3. u-boot load kernel and device tree blob (dtb)



|-----|

| |

| device tree blob |

| |

|-----|<----- 0x00f0,0000 (15M boundary)

| |

| |

| |

| |

|-----|

| | 0x00f0,0000 - 0x8000 > uncompressed kernel

| | 目前kernel在解压后远小于15M,所以预留的空间是足够的

| ulmage |

| |

| |

| |

-----|<----- 0x0000,8000 (32K boundary)

| |

| |

| 32K |

| |

|-----|<----- 0x0000,0000