

去掉miniloader, ThreadX and u-boot的优化编译选项

1. miniloader

in tools.makedefs.mk

```
CROSS_CFLAGS = $(CROSS_CMN_FLAGS) -Os -Wall $(CDEFS) $(INCFLAGS)
```

==>

```
CROSS_CFLAGS = $(CROSS_CMN_FLAGS) -O1 -Wall $(CDEFS) $(INCFLAGS)
```

如果不带-O1，则生成的code无法载入SoC的 LCM。

报错如下：

```
| LD      bin/MiniLoader.elf
```

```
| arm-poky-linux-gnueabi-ld --sysroot=/home/walterzh/gerrit/tmp/sysroots/granite2 -  
L/home/walterzh/gerrit/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-  
gnueabi/./lib/gcc/arm-poky-linux-gnueabi/4.8.1 -EL -Map /home/walterzh/gerrit/tmp/work/armv7a-  
vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-  
r0/git/bin/MiniLoader.map --gc-sections --script /home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-  
poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/tools/MiniLoader.ld  
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-  
loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/entry.o  
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-  
loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/fwVersion.o  
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-  
loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/main_entry.o  
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-  
loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/code_images.o  
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-  
loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/divmod.o
```

/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/minPrintf.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/uart.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/utils.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/assert.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/memset.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/memcpy.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/crc32.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/hex_dump.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/ID_utils.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/memcmp.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/mpu.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/cache.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/avs_api.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/dro.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/UH_AVS.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/UH_I2CReg88PG870.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/CortexR4.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/crypto_callback.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/validate_code.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/ddr.o

/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/ddr_leveling.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/initMC5.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/spd_rom.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/flash_callback.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/i2c.o /home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/i2c_api.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/UTF_PinConfig.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/pll.o /home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/gr2_cfgs.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/SSPLL3P0G_core.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/pmu.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/lpp_apmu.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/UH_PMU_Dividers.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/UH_PMU_Device.o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/objs/dwApbUart.o -o
/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/bin/MiniLoader.elf

| arm-poky-linux-gnueabi-ld: /home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/bin/MiniLoader.elf section
`.text.APMU_set_clk_div' will not fit in region `lcm'

| arm-poky-linux-gnueabi-ld: region `lcm' overflowed by 25212 bytes

| make: *** [/home/walterzh/gerrit/tmp/work/armv7a-vfp-neon-poky-linux-gnueabi/g2-loader/v2014.01+gitAUTOINC+80aa8ef8b0-r0/git/bin/MiniLoader.elf] Error 1

| ERROR: oe_runmake failed

2. ThreadX

in buildenv.mk

```
1. ifeq ($(TYPE), debug)
2.     MACROS += DEBUG
3.     MACROS += HAVE_DBG_PRINTF
4.     CCPARAM = -O0 -Wno-unknown-pragmas -fno-builtin-memcpy
5. else ifeq ($(TYPE), release)
6.     CCPARAM = -Os -fno-strict-aliasing -fno-builtin-memcpy
7. else
8.     $(error Unsupported build type '$(TYPE)' - supported build types are: 'debug' 'release')
9. endif
```

编译开关TYPE=debug即可。

in meta-granite/recipes-bsp/g2-r4/g2-r4.bb

```
#TYPE = "release"
```

```
TYPE = "debug"
```

已经默认设为"debug"

3. u-boot

in meta-granite/recipes-bsp/u-boot/u-boot.inc

```
1. # Don't want to use -O2
2. #EXTRA_OEMAKE = "CROSS_COMPILE=${TARGET_PREFIX} OPTFLAGS='-Os'"
3. EXTRA_OEMAKE = "CROSS_COMPILE=${TARGET_PREFIX} OPTFLAGS='-O0'"
```

把'-Os' 变为 '-O0'

设置miniloader breakpoint

Security core R4是加电就运行的core (SoC的bootrom运行在R4上), bootrom载入miniloader。为了让xdb有机会调试miniloader,对source/main/entry.S作如下修改：

```
diff --git a/source/main/entry.S b/source/main/entry.S
```

```
index 942b60b..59ce760 100644
```

```
--- a/source/main/entry.S
```

```
+++ b/source/main/entry.S
```

```
@@ -49,8 +49,8 @@ LABEL(_SPloaded)
```

```
;/# Debugger foo. Kill this later!! Hold here for a bit so XDB can get its breakpoints re-init'ed.
```

```
-#if 0
```

```
-    mov r0, #0x100000
```

```
+#if 1
```

```
+    mov r0, #0xf00000
```

```
1: nop
```

```
    nop
```

```
    nop
```

```

1.  ;# Debugger foo. Kill this later!! Hold here for a bit so XDB can get its break
    points re-init'ed.
2.  #if 1
3.      mov r0, #0xf00000
4.  1:  nop
5.      nop
6.      nop
7.      nop
8.      nop
9.      subs r0,r0,#1
10.     bne 1b
11. #endif
12.     nop
13.     bl _set_up_bss_vars
14.     ;# call the code to authenticate this
15.     blx main_entry
16.     ;# should never return from this.
17. LABEL(_hello)
18.     b _hello

```

====>

XDB可以成功attach R4 core，并stop在上面的无限循环，但XDB无法set breakpoint(无论是soft breakpoint还是hard breakpoint),XDB报miniload code所在LCM不可读！

在XDB中查看code所在LCM的内容，无法读出！！！这样soft breakpoint自然无法设置（因为XDB会在修改code）

同时使用下面的命令设置hard breakpoint也不行，XDB竟然报告对应的hardware breakpoint register已被使用(used) --- 完全无厘头的原因。

set breakpoint at 0xD0F08838 hard 0

怀疑是否R4 core(security core)的anti-debug functionality启动了！是否与efuse中相关设置有关？！有待进一步调查。

Alternative method:

在miniloader退出以前，载入ThreadX(on R4)和u-boot(on A53)之前加入"wait code"。

```
$ git diff
```

```
index be24a71..aa50ecc 100644
```

```
--- a/source/main/main_entry.c
```

```
+++ b/source/main/main_entry.c
```

```
@@ -304,6 +304,10 @@ int main_entry()
```

```
    if (valid_code_images == true)
```

```
    {
```

```
        // x Execute loaded images, DOES NOT RETURN!!
```

```
+
```

```
+        int wait_xdb = 1;
```

```
+        while(wait_xdb);
```

```
+
```

```
        launch_code_images(rom_type, board_type, AP_loaded_code_launch_addr,  
R4_loaded_code_launch_addr);
```

```
        checkpoint('x');
```

```
    } else
```

但XDB attach到R4都有问题！

设置ThreadX breakpoint on R4

add code to make ThreadX wait on R4 core.

```
$ git diff
```

```
diff --git a/oem/marvell/88pa6220_r4/init/src/main.c b/oem/marvell/88pa6220_r4/init/src/main.c
```

index fba3075..7128d2f 100755

--- a/oem/marvell/88pa6220_r4/init/src/main.c

+++ b/oem/marvell/88pa6220_r4/init/src/main.c

```
@@ -73,6 +73,10 @@ extern func_ptr __CTOR_LIST__[];    /* C++ static/global constructor list
*/
```

```
*/
```

```
int main( void )
```

```
{
```

```
+   int wait_xdb = 1;
```

```
+
```

```
+   while(wait_xdb);
```

```
+
```

```
    unsigned long nptrs = (unsigned long) __CTOR_LIST__[0];
```

```
    unsigned i;
```

当XDB attach到R4 core以后，在R4 core上载入symbol，并可以设置软件断点，可以debug。那为什么miniloader不可以呢?奇怪！

设置u-boot breakpoint on A53

add waiting code in arch/arm/cpu/armv7/start.S

```
$ git diff
```

```
diff --git a/arch/arm/cpu/armv7/start.S b/arch/arm/cpu/armv7/start.S
```

index 6d9ba05..a693278 100755

--- a/arch/arm/cpu/armv7/start.S

+++ b/arch/arm/cpu/armv7/start.S

@@ -204,6 +204,15 @@ reset:

bl cpu_init_crit

#endif

+ mov r6, #0x1

+1: nop

+ nop

+ nop

+ nop

+ cmp r6, #0x0

+ bne 1b

+ nop

+

bl _main

当XDB attach到A53 core后，载入u-boot symbol，可以设置软件断点，可以debug。

