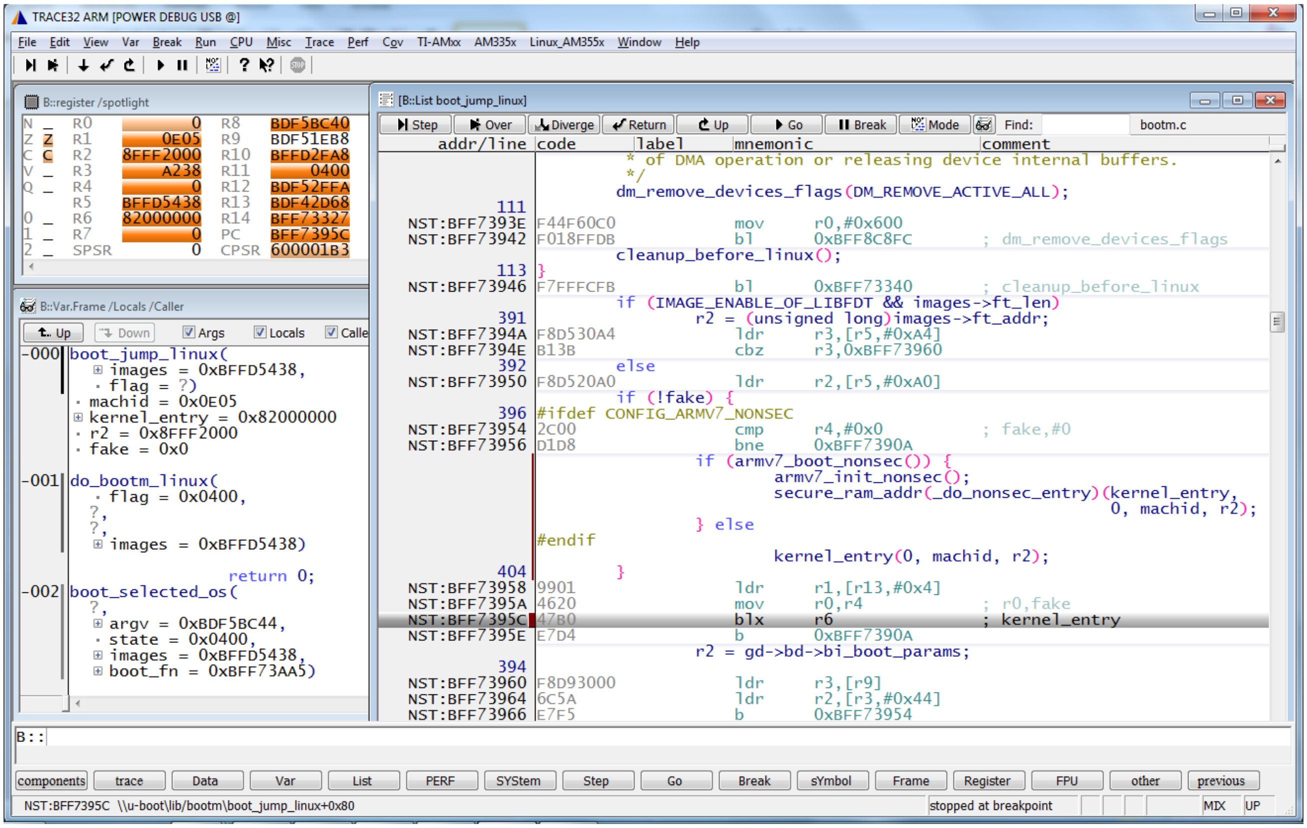
U-boot to Kernel Start

Test is run with SD booting using SDK6.3 on AM335x GP EVM

1. **Stop @u-boot prompt, and setup u-boot breakpoint as shown below in "/arch/arm/lib/bootm.c"**
   - R6=0x82000000 matching "loadaddr=0x82000000" from running "printenv" @u-boot prompt.



```
N _   R0          0   R8   BDF5BC40
Z Z   R1       0E05   R9   BDF51EB8
C C   R2   8FFF2000   R10  BFFD2FA8
V _   R3       A238   R11      0400
Q _   R4          0   R12  BDF52FFA
      R5   BFFD5438   R13  BDF42D68
0 _   R6   82000000   R14  BFF73327
1 _   R7          0   PC   BFF7395C
2 _   SPSR        0   CPSR 600001B3
```

```
-000 boot_jump_linux(
        images = 0xBFFD5438,
        flag = ?)
     machid = 0x0E05
     kernel_entry = 0x82000000
     r2 = 0x8FFF2000
     fake = 0x0

-001 do_bootm_linux(
        flag = 0x0400,
       ?,
       ?,
        images = 0xBFFD5438)

                      return 0;
-002 boot_selected_os(
       ?,
        argv = 0xBDF5BC44,
        state = 0x0400,
        images = 0xBFFD5438,
        boot_fn = 0xBFF73AA5)
```

```
      addr/line code       label    mnemonic                 comment
                              * of DMA operation or releasing device internal buffers.
                              */
                             dm_remove_devices_flags(DM_REMOVE_ACTIVE_ALL);
             111
   NST:BFF7393E F44F60C0            mov     r0,#0x600
   NST:BFF73942 F018FFDB            bl      0xBFF8C8FC       ; dm_remove_devices_flags
                             cleanup_before_linux();
             113 }
   NST:BFF73946 F7FFFCFB            bl      0xBFF73340       ; cleanup_before_linux
                             if (IMAGE_ENABLE_OF_LIBFDT && images->ft_len)
                                     r2 = (unsigned long)images->ft_addr;
             391
   NST:BFF7394A F8D530A4            ldr     r3,[r5,#0xA4]
   NST:BFF7394E B13B                cbz     r3,0xBFF73960
             392             else
   NST:BFF73950 F8D520A0            ldr     r2,[r5,#0xA0]
                             if (!fake) {
             396 #ifdef CONFIG_ARMV7_NONSEC
   NST:BFF73954 2C00                cmp     r4,#0x0          ; fake,#0
   NST:BFF73956 D1D8                bne     0xBFF7390A
                                     if (armv7_boot_nonsec()) {
                                             armv7_init_nonsec();
                                             secure_ram_addr(_do_nonsec_entry)(kernel_entry,
                                                                               0, machid, r2);
                                     } else
                 #endif
                                             kernel_entry(0, machid, r2);
             404             }
   NST:BFF73958 9901                ldr     r1,[r13,#0x4]
   NST:BFF7395A 4620                mov     r0,r4            ; r0,fake
   NST:BFF7395C 47B0                blx     r6               ; kernel_entry
   NST:BFF7395E E7D4                b       0xBFF7390A
                                     r2 = gd->bd->bi_boot_params;
             394
   NST:BFF73960 F8D93000            ldr     r3,[r9]
   NST:BFF73964 6C5A                ldr     r2,[r3,#0x44]
   NST:BFF73966 E7F5                b       0xBFF73954
```

```
B::
```

```
NST:BFF7395C  \\u-boot\lib/bootm\boot_jump_linux+0x80        stopped at breakpoint
```

2. **Run "boot" cmd @u-boot prompt, and serial log is shown below**
   Where kernel zImage, DTB are loaded from SD to DDR

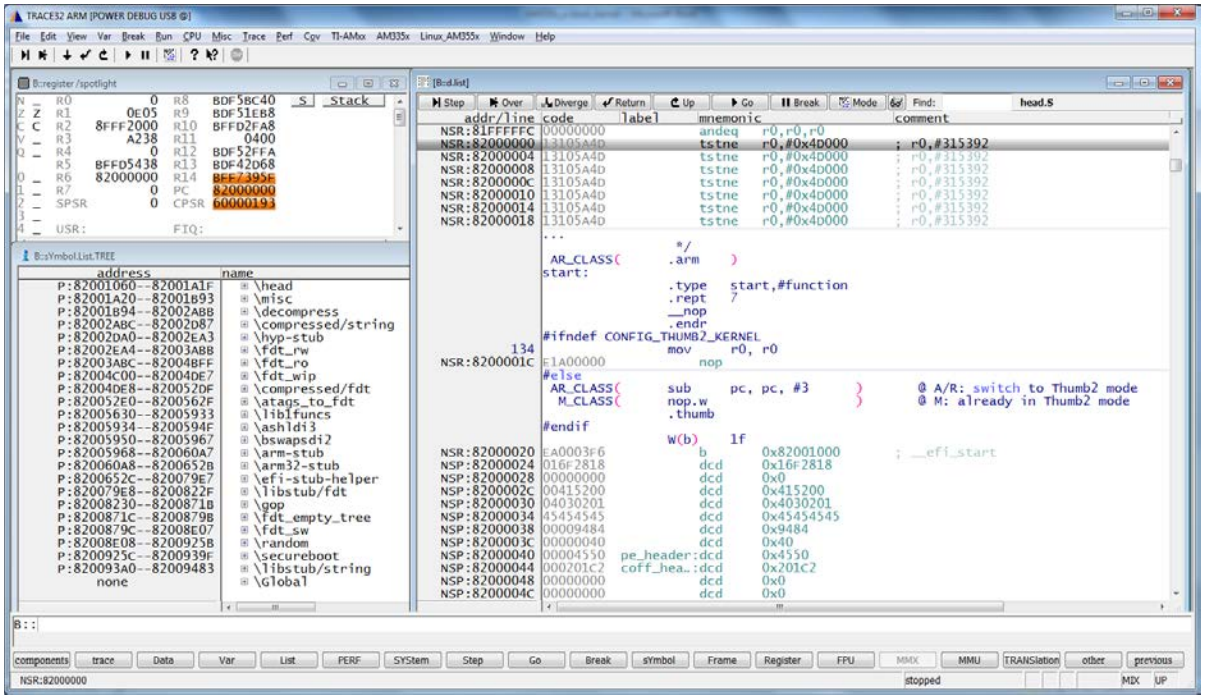
```
=> boot
switch to partitions #0, OK
mmc0 is current device
SD/MMC found on device 0
switch to partitions #0, OK
mmc0 is current device
Scanning mmc 0:1...
switch to partitions #0, OK
mmc0 is current device
SD/MMC found on device 0
4280832 bytes read in 283 ms (14.4 MiB/s)
41528 bytes read in 4 ms (9.9 MiB/s)
## Flattened Device Tree blob at 88000000
   Booting using the fdt blob at 0x88000000
   Loading Device Tree to 8fff2000, end 8ffff237 ... OK

Starting kernel ...
```

3. The call flow on bootz cmd for u-boot loading zImage from SD to DDR

```
B::Var.Frame_/Locals_/Caller
-000|boot_jump_linux(
    |   images = 0xBFFD5438,
    |   flag = ?)
    | machid = 0x0E05
    | kernel_entry = 0x82000000
    | r2 = 0x8FFF2000
    | fake = 0x0
    |
-001|do_bootm_linux(
    |   flag = 0x0400,
    | ?,
    | ?,
    |   images = 0xBFFD5438)
    |
    |           return 0;
-002|boot_selected_os(
    | ?,
    |   argv = 0xBDF5BC44,
    |   state = 0x0400,
    |   images = 0xBFFD5438,
    |   boot_fn = 0xBFF73AA5)
    |
    |     boot_fn(state, argc, argv, images);
-003|do_bootm_states(
    |   cmdtp = 0xBFFD2FA8,
    |   flag = 0x0,
    |   argc = 0x3,
    |   argv = 0xBDF5BC44,
    |   states = 0x0710,
    |   images = 0xBFFD5438,
    |   boot_progress = 0x1)
    | boot_fn = 0xBFF73AA5
    |
    |
-004|do_bootz(
    |   cmdtp = 0xBFFD2FA8,
    |   flag = 0x0,
    |   argc = 0x3,
    |   argv = 0xBDF5BC44)
    | ret = 0x0
    |
    |     ret = do_bootm_states(cmdtp, flag, argc, argv,
-005|cmd_call(inline)
-005|cmd_process(
```

4. **Run one more stepout of the previously set breakpoint <BLX R6>**
   - The first instruction @0x82000000 is “arch/arm/boot/compressed/head.S” for decompressing kernel image.
   - The decompression code is under “arch/arm/boot/compressed”, and piggy-backed into zImage.
   - The decompressed kernel physical address (0x80008000 below) is determined in this file “arch/arm/boot/compressed/head.S”.



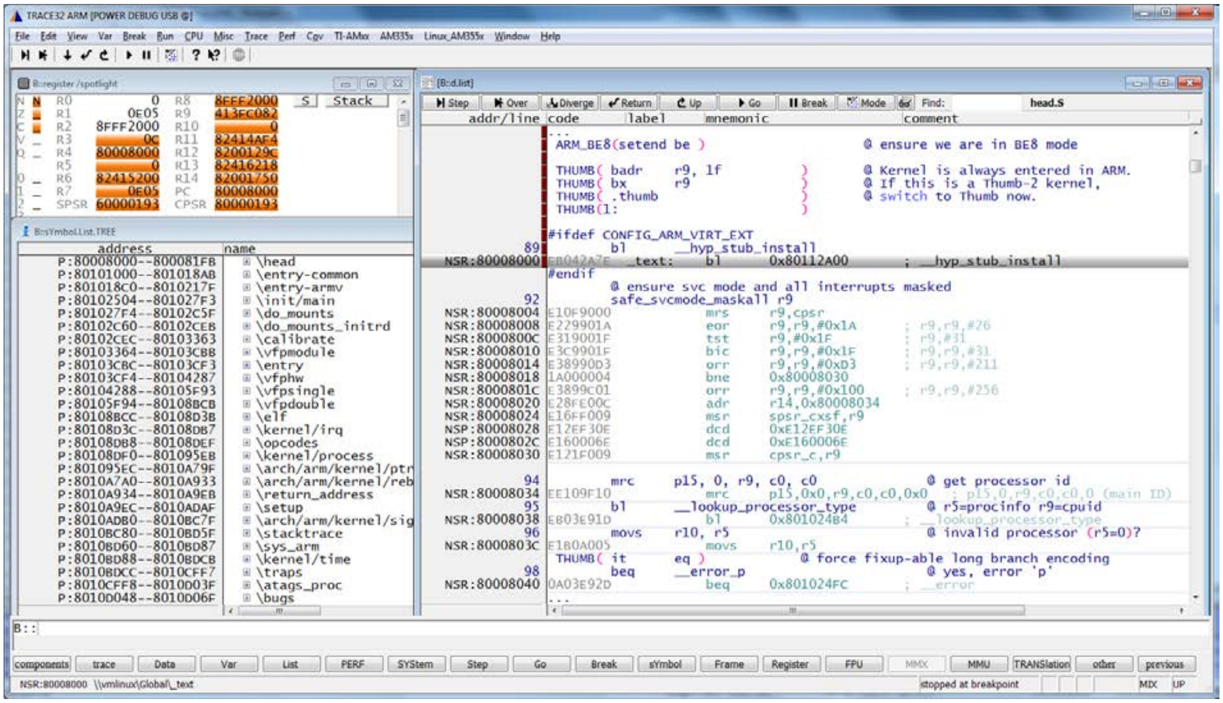
TRACE32 ARM [POWER DEBUG USB @]
File Edit View Var Break Run CPU Misc Trace Perf Cov TI-AMxx AM335x Linux_AM355x Window Help
B::register /spotlight
B::sYmbol.List.TREE
[B::d.list]
Step Over Diverge Return Up Go Break Mode Find: head.S
components trace Data Var List PERF SYStem Step Go Break sYmbol Frame Register FPU MMX MMU TRANSlation other previous
NSR:82000000
stopped
MIX UP

```
N _  R0          0  R8    BDF5BC40   S   Stack
Z Z  R1       0E05  R9    BDF51EB8
C C  R2   8FFF2000  R10   BFFD2FA8
V _  R3       A238  R11       0400
Q _  R4          0  R12   BDF52FFA
     R5   BFFD5438  R13   BDF42D68
0 _  R6   82000000  R14   BFF7395F
1 _  R7          0  PC    82000000
2 _  SPSR        0  CPSR  60000193
3 _
4 _  USR:           FIQ:
```

```
          address              name
P:82001060--82001A1F    \head
P:82001A20--82001B93    \misc
P:82001B94--82002ABB    \decompress
P:82002ABC--82002D87    \compressed/string
P:82002DA0--82002EA3    \hyp-stub
P:82002EA4--82003ABB    \fdt_rw
P:82003ABC--82004BFF    \fdt_ro
P:82004C00--82004DE7    \fdt_wip
P:82004DE8--820052DF    \compressed/fdt
P:820052E0--8200562F    \atags_to_fdt
P:82005630--82005933    \lib1funcs
P:82005934--8200594F    \ashldi3
P:82005950--82005967    \bswapsdi2
P:82005968--820060A7    \arm-stub
P:820060A8--8200652B    \arm32-stub
P:8200652C--820079E7    \efi-stub-helper
P:820079E8--8200822F    \libstub/fdt
P:82008230--8200871B    \gop
P:8200871C--8200879B    \fdt_empty_tree
P:8200879C--82008E07    \fdt_sw
P:82008E08--8200925B    \random
P:8200925C--8200939F    \secureboot
P:820093A0--82009483    \libstub/string
      none              \Global
B::
```

```
   addr/line code      label      mnemonic                     comment
NSR:81FFFFFC 00000000             andeq    r0,r0,r0
NSR:82000000 13105A4D             tstne    r0,#0x4D000         ; r0,#315392
NSR:82000004 13105A4D             tstne    r0,#0x4D000         ; r0,#315392
NSR:82000008 13105A4D             tstne    r0,#0x4D000         ; r0,#315392
NSR:8200000C 13105A4D             tstne    r0,#0x4D000         ; r0,#315392
NSR:82000010 13105A4D             tstne    r0,#0x4D000         ; r0,#315392
NSR:82000014 13105A4D             tstne    r0,#0x4D000         ; r0,#315392
NSR:82000018 13105A4D             tstne    r0,#0x4D000         ; r0,#315392
             ...
                               */
              AR_CLASS(        .arm      )
             start:
                               .type    start,#function
                               .rept    7
                               __nop
                               .endr
             #ifndef CONFIG_THUMB2_KERNEL
         134                   mov      r0, r0
NSR:8200001C E1A00000             nop
             #else
              AR_CLASS(        sub      pc, pc, #3       )     @ A/R: switch to Thumb2 mode
               M_CLASS(        nop.w                     )     @ M: already in Thumb2 mode
                               .thumb
             #endif
                               W(b)     1f
NSR:82000020 EA0003F6             b        0x82001000          ; __efi_start
NSP:82000024 016F2818             dcd      0x16F2818
NSP:82000028 00000000             dcd      0x0
NSP:8200002C 00415200             dcd      0x415200
NSP:82000030 04030201             dcd      0x4030201
NSP:82000034 45454545             dcd      0x45454545
NSP:82000038 00009484             dcd      0x9484
NSP:8200003C 00000040             dcd      0x40
NSP:82000040 00004550 pe_header:dcd      0x4550
NSP:82000044 000201C2 coff_hea..:dcd      0x201C2
NSP:82000048 00000000             dcd      0x0
NSP:8200004C 00000000             dcd      0x0
```

5. **Setup breakpoint @0x80008000, and run**
   - Decompression would take time to finish before hitting bkpt @0x80008000
   - The first instruction @0x80008000 is “arch/arm/kernel/head.S”, the kernel entry code before MMU is ON



TRACE32 ARM [POWER DEBUG USB @]
File Edit View Var Break Run CPU Misc Trace Perf Cov TI-AMxx AM335x Linux_AM355x Window Help
B::register /spotlight
B::sYmbol.List.TREE
[B::d.list]
Step Over Diverge Return Up Go Break Mode Find: head.S
components trace Data Var List PERF SYStem Step Go Break sYmbol Frame Register FPU MMX MMU TRANSlation other previous
NSR:80008000 \\vmlinux\Global_text
stopped at breakpoint
MIX UP

```
N N  R0          0  R8    8FFF2000   S   Stack
Z _  R1       0E05  R9    413FC082
C _  R2   8FFF2000  R10          0
V _  R3         0C  R11   82414AF4
Q _  R4   80008000  R12   8200129C
     R5          0  R13   82416218
0 _  R6   82415200  R14   82001750
1 _  R7       0E05  PC    80008000
2 _  SPSR 60000193  CPSR  80000193
```

```
          address              name
P:80008000--800081FB    \head
P:80101000--801018AB    \entry-common
P:801018C0--8010217F    \entry-armv
P:80102504--801027F3    \init/main
P:801027F4--80102C5F    \do_mounts
P:80102C60--80102CEB    \do_mounts_initrd
P:80102CEC--80103363    \calibrate
P:80103364--80103CBB    \vfpmodule
P:80103CBC--80103CF3    \entry
P:80103CF4--80104287    \vfphw
P:80104288--80105F93    \vfpsingle
P:80105F94--80108BCB    \vfpdouble
P:80108BCC--80108D3B    \elf
P:80108D3C--80108DB7    \kernel/irq
P:80108DB8--80108DEF    \opcodes
P:80108DF0--801095EB    \kernel/process
P:801095EC--8010A79F    \arch/arm/kernel/ptr
P:8010A7A0--8010A933    \arch/arm/kernel/reb
P:8010A934--8010A9EB    \return_address
P:8010A9EC--8010ADAF    \setup
P:8010ADB0--8010BC7F    \arch/arm/kernel/sig
P:8010BC80--8010BD5F    \stacktrace
P:8010BD60--8010BD87    \sys_arm
P:8010BD88--8010BDCB    \kernel/time
P:8010BDCC--8010CFF7    \traps
P:8010CFF8--8010D03F    \atags_proc
P:8010D048--8010D06F    \bugs
B::
```

```
   addr/line code      label      mnemonic                     comment
             ...
              ARM_BE8(setend be )                           @ ensure we are in BE8 mode

              THUMB( badr    r9, 1f             )           @ Kernel is always entered in ARM.
              THUMB( bx      r9                 )           @ If this is a Thumb-2 kernel,
              THUMB( .thumb                     )           @ switch to Thumb now.
              THUMB(1:                          )

             #ifdef CONFIG_ARM_VIRT_EXT
          89        bl      __hyp_stub_install
NSR:80008000 EB042A7E  _text:    bl       0x80112A00       ; __hyp_stub_install
             #endif
                    @ ensure svc mode and all interrupts masked
          92        safe_svcmode_maskall r9
NSR:80008004 E10F9000             mrs      r9,cpsr
NSR:80008008 E229901A             eor      r9,r9,#0x1A         ; r9,r9,#26
NSR:8000800C E319001F             tst      r9,#0x1F            ; r9,#31
NSR:80008010 E3C9901F             bic      r9,r9,#0x1F         ; r9,r9,#31
NSR:80008014 E38990D3             orr      r9,r9,#0xD3         ; r9,r9,#211
NSR:80008018 1A000004             bne      0x80008030
NSR:8000801C E3899C01             orr      r9,r9,#0x100        ; r9,r9,#256
NSR:80008020 E28FE00C             adr      r14,0x80008034
NSR:80008024 E16FF009             msr      spsr_cxsf,r9
NSP:80008028 E12EF30E             dcd      0xE12EF30E
NSP:8000802C E160006E             dcd      0xE160006E
NSR:80008030 E121F009             msr      cpsr_c,r9

          94        mrc     p15, 0, r9, c0, c0                @ get processor id
NSR:80008034 EE109F10             mrc      p15,0x0,r9,c0,c0,0x0   ; p15,0,r9,c0,c0,0 (main ID)
          95        bl      __lookup_processor_type           @ r5=procinfo r9=cpuid
NSR:80008038 EB03E91D             bl       0x801024B4       ; __lookup_processor_type
          96        movs    r10, r5                           @ invalid processor (r5=0)?
NSR:8000803C E1B0A005             movs     r10,r5
              THUMB( it     eq )                 @ force fixup-able long branch encoding
          98        beq     __error_p                         @ yes, error 'p'
NSR:80008040 0A03E92D             beq      0x801024FC       ; __error
             ...
```