



```
1. root@granite2v8:~# gdbserver localhost:2345 ./jbigtest
2. Process ./jbigtest created; pid = 351
3. gdbserver: Unable to determine the number of hardware watchpoints available.
4. gdbserver: Unable to determine the number of hardware breakpoints available.
5. Listening on port 2345
6. Remote debugging from host 10.38.52.106
7.
8. jbig_platform_get_config codec 0
9. UIO_LIB: uio_lib_init
10. posix: posix_create_thread start=0x7fb7e96fdc stack=0x7fb7eae150 stack_size=
    131072
11. UIO_LIB: Waiting for UIO interrupt event
12. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-core-uio0, num = 0x
    4, ver=devicetree
13. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio4/maps/map0
14. UIO_LIB: num = 0
15. UIO_LIB: name = /jbig-codec-core-uio0
16. UIO_LIB: phy_addr = 0xf9000000
17. UIO_LIB: size = 4096
18. UIO_LIB: Opened device jbig-codec-core-uio0
19. dev_jbig 0x416160 map: 0x4161f0, map->virt_addr 0xb7ffa000
20. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-idma-uio0, num = 0x
    5, ver=devicetree
21. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio5/maps/map0
22. UIO_LIB: num = 0
23. UIO_LIB: name = /jbig-codec-idma-uio0
24. UIO_LIB: phy_addr = 0xf9001000
25. UIO_LIB: size = 4096
26. UIO_LIB: Opened device jbig-codec-idma-uio0
27. dev_idma 0x416280 map: 0x416310, map->virt_addr 0xb7ff9000
28. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-uio0, num = 0x
    6, ver=devicetree
29. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio6/maps/map0
30. UIO_LIB: num = 0
31. UIO_LIB: name = /jbig-codec-odma-uio0
32. UIO_LIB: phy_addr = 0xf9002000
33. UIO_LIB: size = 4096
34. UIO_LIB: Opened device jbig-codec-odma-uio0
35. dev_odma 0x4163a0 map: 0x416430, map->virt_addr 0xb7ff8000
36. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-core-uio0, num
    = 0x7, ver=devicetree
37. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio7/maps/map0
38. UIO_LIB: num = 0
39. UIO_LIB: name = /jbig-codec-odma-core-uio0
40. UIO_LIB: phy_addr = 0xf9002800
41. UIO_LIB: size = 4096
42. UIO_LIB: Opened device jbig-codec-odma-core-uio0
43. dev_odma_core 0x4164c0 map: 0x416540, map->virt_addr 0xb7ff7800
44.
45. codec 1
46. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-core-uio1, num = 0x
    8, ver=devicetree
```

```
47. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio8/maps/map0
48. UIO_LIB: num = 0
49. UIO_LIB: name = /jbig-codec-core-uio1
50. UIO_LIB: phy_addr = 0xf9008000
51. UIO_LIB: size = 4096
52. UIO_LIB: Opened device jbig-codec-core-uio1
53. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-idma-uio1, num = 0x
9, ver=devicetree
54. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio9/maps/map0
55. UIO_LIB: num = 0
56. UIO_LIB: name = /jbig-codec-idma-uio1
57. UIO_LIB: phy_addr = 0xf9009000
58. UIO_LIB: size = 4096
59. UIO_LIB: Opened device jbig-codec-idma-uio1
60. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-uio1, num = 0x
a, ver=devicetree
61. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio10/maps/map0
62. UIO_LIB: num = 0
63. UIO_LIB: name = /jbig-codec-odma-uio1
64. UIO_LIB: phy_addr = 0xf900a000
65. UIO_LIB: size = 4096
66. UIO_LIB: Opened device jbig-codec-odma-uio1
67. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-core-uio1, num
= 0xb, ver=devicetree
68. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio11/maps/map0
69. UIO_LIB: num = 0
70. UIO_LIB: name = /jbig-codec-odma-core-uio1
71. UIO_LIB: phy_addr = 0xf900a800
72. UIO_LIB: size = 4096
73. UIO_LIB: Opened device jbig-codec-odma-core-uio1
74. dev_odma_core 0x416dc0 map: 0x416e10, map->virt_addr 0xb7ff0800
75.
76. codec 2
77. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-core-uio2, num = 0x
c, ver=devicetree
78. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio12/maps/map0
79. UIO_LIB: num = 0
80. UIO_LIB: name = /jbig-codec-core-uio2
81. UIO_LIB: phy_addr = 0xf9010000
82. UIO_LIB: size = 4096
83. UIO_LIB: Opened device jbig-codec-core-uio2
84. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-idma-uio2, num = 0x
d, ver=devicetree
85. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio13/maps/map0
86. UIO_LIB: num = 0
87. UIO_LIB: name = /jbig-codec-idma-uio2
88. UIO_LIB: phy_addr = 0xf9011000
89. UIO_LIB: size = 4096
90. UIO_LIB: Opened device jbig-codec-idma-uio2
91. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-uio2, num = 0x
e, ver=devicetree
92. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio14/maps/map0
93. UIO_LIB: num = 0
94. UIO_LIB: name = /jbig-codec-odma-uio2
```

```

95. UIO_LIB: phy_addr = 0xf9012000
96. UIO_LIB: size = 4096
97. UIO_LIB: Opened device jbig-codec-odma-uio2
98. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-core-uio2, num
    = 0xf, ver=devicetree
99. UIO_LIB: uio_gBad mode in Error handler detected, code 0xbf000002 -- SError
100. CPU: 0 PID: 351 Comm: jbigtest Tainted: G          0      4.2.8-yocto-standar
    d #1
101. Hardware name: mv6220 TurnOn Card (DT)
102. task: ffffffff039c13f00 ti: ffffffff039e24000 task.ti: ffffffff039e24000
103. PC is at 0x7fb7e578f4
104. LR is at 0x7fb7e57a30
105. pc : [<0000007fb7e578f4>] lr : [<0000007fb7e57a30>] pstate: 80000000
106. sp : ffffffff039e27ff0
107. x29: 0000007fffffffbb10 x28: 0000000000000000
108. x27: 0000000000000000 x26: 0000000000000000
109. x25: 0000000000000000 x24: 0000000000000000
110. x23: 0000000000000000 x22: 0000000000000000
111. x21: 0000000000000000 x20: 0000000000000000
112. x19: 0000000000403ea8 x18: 0000000000000000
113. x17: 0000007fb7e6e460 x16: 0000007fb7e578d0
114. x15: 00000000000057d8 x14: 0000007fb7e552f0
115. x13: 0000007fb7ffecb8 x12: 000000000000002d
116. x11: 0101010101010101 x10: 0000000000000000
117. x9 : 0000000000000004 x8 : 0000007fb7e8e680
118. x7 : 0000000000000000 x6 : 000000000000003f
119. x5 : 0000000000000040 x4 : 0000000000000000
120. x3 : 0000007fb7e8e540 x2 : 0000007fb7e8e540
121. x1 : 000000000000003f x0 : 0000007fb7ff8000
122.
123. et_device_map: map for /sys/class/uio/uio15/maps/map0
124. UIO_LIB: num = 0
125. UIO_LIB: name = /jbig-codec-odma-core-uio2
126. UIO_LIB: phy_addr = 0xf9012800
127. UIO_LIB: size = 4096
128. UIO_LIB: Opened device jbig-codec-odma-core-uio2
129. dev_odma_core 0x416820 map: 0x416870, map->virt_addr 0xb7cf9800
130.
131. codec 3
132. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-core-uio3, num = 0x
    10, ver=devicetree
133. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio16/maps/map0
134. UIO_LIB: num = 0
135. UIO_LIB: name = /jbig-codec-core-uio3
136. UIO_LIB: phy_addr = 0xf9018000
137. UIO_LIB: size = 4096
138. UIO_LIB: Opened device jbig-codec-core-uio3
139. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-idma-uio3, num = 0x
    11, ver=devicetree
140. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio17/maps/map0
141. UIO_LIB: num = 0
142. UIO_LIB: name = /jbig-codec-idma-uio3
143. UIO_LIB: phy_addr = 0xf9019000
144. UIO_LIB: size = 4096

```

```

145. UIO_LIB: Opened device jbig-codec-idma-uio3
146. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-uio3, num = 0x
12, ver=devicetree
147. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio18/maps/map0
148. UIO_LIB: num = 0
149. UIO_LIB: name = /jbig-codec-odma-uio3
150. UIO_LIB: phy_addr = 0xf901a000
151. UIO_LIB: size = 4096
152. UIO_LIB: Opened device jbig-codec-odma-uio3
153. UIO_LIB: uio_locate_dev_by_number: found dev: jbig-codec-odma-core-uio3, num
= 0x13, ver=devicetree
154. UIO_LIB: uio_get_device_map: map for /sys/class/uio/uio19/maps/map0
155. UIO_LIB: num = 0
156. UIO_LIB: name = /jbig-codec-odma-core-uio3
157. UIO_LIB: phy_addr = 0xf901a800
158. UIO_LIB: size = 4096
159. UIO_LIB: Opened device jbig-codec-odma-core-uio3
160. dev_odma_core 0x4174b0 map: 0x4173c0, map->virt_addr 0xb7cf5800
161. Killing all inferiors

```

出现了如下trap

```

1. UIO_LIB: uio_gBad mode in Error handler detected, code 0xbf000002 -- SError
2. CPU: 0 PID: 351 Comm: jbigtest Tainted: G          0 4.2.8-yocto-standar
d #1
3. Hardware name: mv6220 TurnOn Card (DT)
4. task: ffffffff039c13f00 ti: ffffffff039e24000 task.ti: ffffffff039e24000
5. PC is at 0x7fb7e578f4
6. LR is at 0x7fb7e57a30
7. pc : [<0000007fb7e578f4>] lr : [<0000007fb7e57a30>] pstate: 80000000
8. sp : ffffffff039e27ff0
9. x29: 0000007fffffff10 x28: 0000000000000000
10. x27: 0000000000000000 x26: 0000000000000000
11. x25: 0000000000000000 x24: 0000000000000000
12. x23: 0000000000000000 x22: 0000000000000000
13. x21: 0000000000000000 x20: 0000000000000000
14. x19: 0000000000403ea8 x18: 0000000000000000
15. x17: 0000007fb7e6e460 x16: 0000007fb7e578d0
16. x15: 000000000000057d8 x14: 0000007fb7e552f0
17. x13: 0000007fb7ffecb8 x12: 000000000000002d
18. x11: 0101010101010101 x10: 0000000000000000
19. x9 : 0000000000000004 x8 : 0000007fb7e8e680
20. x7 : 0000000000000000 x6 : 000000000000003f
21. x5 : 0000000000000040 x4 : 0000000000000000
22. x3 : 0000007fb7e8e540 x2 : 0000007fb7e8e540
23. x1 : 000000000000003f x0 : 0000007fb7ff8000

```

出错时的PC为

PC is at 0x7fb7e578f4

查看jbig在gdb server控制下的memory map如下

```

1. root@granite2v8:~# cat /proc/351/maps
2. 00400000-00405000 r-xp 00000000 b3:22 6205 /ho
me/root/jbittest
3. 00414000-00415000 rw-p 00004000 b3:22 6205 /ho
me/root/jbittest
4. 00415000-00416000 rw-p 00000000 00:00 0 [he
ap]
5. 7fb7d0c000-7fb7e3c000 r-xp 00000000 b3:22 23 /li
b64/libc-2.21.so
6. 7fb7e3c000-7fb7e4b000 ---p 00130000 b3:22 23 /li
b64/libc-2.21.so
7. 7fb7e4b000-7fb7e4f000 r--p 0012f000 b3:22 23 /li
b64/libc-2.21.so
8. 7fb7e4f000-7fb7e51000 rw-p 00133000 b3:22 23 /li
b64/libc-2.21.so
9. 7fb7e51000-7fb7e55000 rw-p 00000000 00:00 0
10. 7fb7e55000-7fb7e5e000 r-xp 00000000 b3:22 8034 /us
r/lib64/libjbig.so.1.0
11. 7fb7e5e000-7fb7e6e000 ---p 00009000 b3:22 8034 /us
r/lib64/libjbig.so.1.0
12. 7fb7e6e000-7fb7e6f000 rw-p 00009000 b3:22 8034 /us
r/lib64/libjbig.so.1.0
13. 7fb7e6f000-7fb7e8f000 rw-p 00000000 00:00 0
14. 7fb7e8f000-7fb7e9e000 r-xp 00000000 b3:22 3424 /us
r/lib64/libdmaalloc.so.1.0
15. 7fb7e9e000-7fb7ead000 ---p 0000f000 b3:22 3424 /us
r/lib64/libdmaalloc.so.1.0
16. 7fb7ead000-7fb7eaf000 rw-p 0000e000 b3:22 3424 /us
r/lib64/libdmaalloc.so.1.0
17. 7fb7eaf000-7fb7eee000 rw-p 00000000 00:00 0
18. 7fb7eee000-7fb7f7f000 r-xp 00000000 b3:22 86 /li
b64/libm-2.21.so
19. 7fb7f7f000-7fb7f8f000 ---p 00091000 b3:22 86 /li
b64/libm-2.21.so
20. 7fb7f8f000-7fb7f90000 r--p 00091000 b3:22 86 /li
b64/libm-2.21.so
21. 7fb7f90000-7fb7f91000 rw-p 00092000 b3:22 86 /li
b64/libm-2.21.so
22. 7fb7f91000-7fb7f97000 r-xp 00000000 b3:22 67 /li
b64/librt-2.21.so
23. 7fb7f97000-7fb7fa6000 ---p 00006000 b3:22 67 /li
b64/librt-2.21.so
24. 7fb7fa6000-7fb7fa7000 r--p 00005000 b3:22 67 /li
b64/librt-2.21.so
25. 7fb7fa7000-7fb7fa8000 rw-p 00006000 b3:22 67 /li
b64/librt-2.21.so
26. 7fb7fa8000-7fb7fbe000 r-xp 00000000 b3:22 90 /li
b64/libpthread-2.21.so
27. 7fb7fbe000-7fb7fcd000 ---p 00016000 b3:22 90 /li
b64/libpthread-2.21.so
28. 7fb7fcd000-7fb7fce000 r--p 00015000 b3:22 90 /li
b64/libpthread-2.21.so
29. 7fb7fce000-7fb7fcf000 rw-p 00016000 b3:22 90 /li

```

```

b64/libpthread-2.21.so
30. 7fb7fcf000-7fb7fd3000 rw-p 00000000 00:00 0
31. 7fb7fd3000-7fb7fef000 r-xp 00000000 b3:22 108 /li
b64/ld-2.21.so
32. 7fb7ff4000-7fb7ff7000 rw-p 00000000 00:00 0
33. 7fb7ffb000-7fb7ffc000 rw-p 00000000 00:00 0
34. 7fb7ffc000-7fb7ffd000 r--p 00000000 00:00 0 [vv
ar]
35. 7fb7ffd000-7fb7ffe000 r-xp 00000000 00:00 0 [vd
so]
36. 7fb7ffe000-7fb7fff000 r--p 0001b000 b3:22 108 /li
b64/ld-2.21.so
37. 7fb7fff000-7fb8001000 rw-p 0001c000 b3:22 108 /li
b64/ld-2.21.so
38. 7fffffd000-8000000000 rw-p 00000000 00:00 0 [st
ack]
39. root@granite2v8:~#

```

0x7fb7e578f4落在libjbig.so.1.0 code中

```
7fb7e55000-7fb7e5e000 r-xp 00000000 b3:22 8034 /usr/lib64/libjbig.so.1.0
```

$0x7fb7e578f4 - 0x7fb7e55000 = 0x28f4$  (offset)

确定libjbig.so.1.0 code的大致偏移

```

1. walterzh@walterzh-Precision-T1650:~/work/current/ccsgit/driver/jbig-codec/jb
ig-codec-app$ /home/walterzh/work/2016-05-LSP/64-bit/tmp/sysroots/x86_64-lin
ux/usr/bin/aarch64-poky-linux/aarch64-poky-linux-readelf -l libjbig.so.1.0
2.
3. Elf file type is DYN (Shared object file)
4. Entry point 0x2770
5. There are 5 program headers, starting at offset 64
6.
7. Program Headers:
8.   Type                Offset                VirtAddr                PhysAddr
9.   FileSiz             MemSiz                Flags  Align
10.  LOAD                 0x0000000000000000 0x0000000000000000 0x0000000000000000
11.   0x00000000000086bc 0x00000000000086bc  R E    10000
12.  LOAD                 0x0000000000009000 0x0000000000019000 0x0000000000019000
13.   0x000000000000508 0x0000000000020b60  RW     10000
14.  DYNAMIC              0x0000000000009018 0x0000000000019018 0x0000000000019018
15.   0x0000000000001f0 0x0000000000001f0  RW      8
16.  NOTE                 0x000000000000158 0x000000000000158 0x000000000000158
17.   0x000000000000024 0x000000000000024  R       4
18.  GNU_STACK            0x0000000000000000 0x0000000000000000 0x0000000000000000
19.   0x0000000000000000 0x0000000000000000  RW     10
20.
21. Section to Segment mapping:
22. Segment Sections...
23. 00  .note.gnu.build-id .gnu.hash .dynsym .dynstr .gnu.version .gnu.ver
sion_r .rela.dyn .rela.plt .init .plt .text .fini .rodata .eh_frame
24. 01  .init_array .fini_array .jcr .dynamic .got .got.plt .data .bss
25. 02  .dynamic
26. 03  .note.gnu.build-id
27. 04

```

```

1.  LOAD                 0x0000000000000000 0x0000000000000000 0x0000000000000000
2.   0x00000000000086bc 0x00000000000086bc  R E    10000

```

即libjbig.so.1.0 载入内存后的偏移为0 - 0x86bc  
地址为零是因为是动态载入，静态期无法确定address.

反汇编libjbig.so.1.0 code to locate the bug code

```

/home/walterzh/work/2016-05-LSP/64-bit/tmp/sysroots/x86_64-linux/usr/bin/aarch64-
poky-linux/aarch64-poky-linux-objdump -dS libjbig.so.1.0

```



```

1. libjbig.so.1.0:      file format elf64-littleaarch64
2.
3.
4. Disassembly of section .init:
5.
6. 0000000000002200 <_init>:
7.     .section .init,"ax",%progbits
8.     .align 2
9.     .global _init
10.    .type _init, %function
11. _init:
12.     stp x29, x30, [sp, -16]!
13. 2200: a9bf7bfd     stp x29, x30, [sp,#-16]!
14.     mov x29, sp
15. 2204: 910003fd     mov x29, sp
16. #if PREINIT_FUNCTION_WEAK
17.     bl call_weak_fn
18. 2208: 9400015a     bl 2770 <call_weak_fn>
19.
20. /* crt0.S puts function epilogues in the .init and .fini sections
21.    corresponding to the prologues in crt0.S. */
22.
23.     .section .init,"ax",%progbits
24.     ldp x29, x30, [sp], 16
25. 220c: a8c17bfd     ldp x29, x30, [sp],#16
26.     RET
27. 2210: d65f03c0     ret
28.
29. Disassembly of section .plt:
30.
31. 0000000000002220 <memcpy@plt-0x20>:
32. 2220: a9bf7bfd     stp x16, x30, [sp,#-16]!
33. 2224: f00000b0     adrp x16, 19000 <__frame_dummy_init_array_entry>
34. 2228: f9413211     ldr x17, [x16,#608]
35. 222c: 91098210     add x16, x16, #0x260
36. 2230: d61f0220     br x17
37. 2234: d503201f     nop
38. 2238: d503201f     nop
39. 223c: d503201f     nop
40.
41. ....
42.
43. 00000000000028d0 <jbig_block_init>:
44. * @author (10/13/2011)
45. *
46. * @param jbig_block_interface
47. */
48. void jbig_block_init(const jbig_block_config_t* jbig_block_interface)
49. {
50.     28d0: d10043ff     sub sp, sp, #0x10
51.     28d4: f90007e0     str x0, [sp,#8]
52.     DBG_MEMLOG(LOG_INFO,"JBIG BLOCK INIT FROM POWER MANAGER!!!\n");
53.     //jbig_block_interface->jbig_regs->JCTL = 0;

```

```

54.      jbig_block_interface->jbig_idma_regs->int_c1 = 0x1FF;
55.      28d8: f94007e0 ldr x0, [sp,#8]
56.      28dc: f9400800 ldr x0, [x0,#16]
57.      28e0: 52803fe1 mov w1, #0x1ff // #511
58.      28e4: b9001401 str w1, [x0,#20]
59.      jbig_block_interface->jbig_odma_regs->UICR = JBIG_ODMA_UDMA_INT_ENABLE_M
      ASK;
60.      28e8: f94007e0 ldr x0, [sp,#8]
61.      28ec: f9400c00 ldr x0, [x0,#24]
62.      28f0: 528007e1 mov w1, #0x3f // #63
63.      28f4: b9001401 str w1, [x0,#20]

```

找到出错指令 28f4: b9001401 str w1, [x0,#20]

也就是

```

jbig_block_interface->jbig_odma_regs->UICR =
JBIG_ODMA_UDMA_INT_ENABLE_MASK;
引起trap.

```

出错的原因：

**jbig block依赖与pegmatite regulator supply power，但pegmatite-regulator.ko没有载入，所以jbig block没有上电！**