```
Messages in this thread
```

```
First message in thread Greg Kroah-Hartman Greg Kro
                                                                                                                            Greg Kroah-Hartman <>
                                                                                         From
                                                                                         Date
                                                                                        From: wilkii.wii <wilkii.wii@hiiawei.com>
                                                                                                                  [ 1025.321302] Call trace:
            Greg Kroah-Hartman
            Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
                                                                                            static void dw_writer(struct dw_spi *dws)
           Greg Kroah-Hartman
           Greg Kroah-Hartman
                                                                                                                 u32 max = tx_max(dws);
u32 max;
u16 txw = 0;
           Greg Kroah-Hartman
                                                                                                                  spin_unlock(&dws->buf_lock);
           Greg Kroah-Hartman
           Greg Kroah-Hartman
           Greg Kroah-Hartman
                                                                                            static void dw_reader(struct dw_spi *dws)
           Greg Kroah-Hartman
           Greg Kroah-Hartman
                                                                                                                 u32 max = rx_max(dws);
u32 max;
u16 rxw;
           Greg Kroah-Hartman
           Greg Kroah-Hartman
         Greg Kroah-Hartman
Greg Kroah-Hartman
Greg Kroah-Hartman
Greg Kroah-Hartman
Jon Hunter
Naresh Kamboju
Guenter Roeck
                                                                                                                                           dws->rx += dws->n_bytes;
                                                                                                                   spin_unlock(&dws->buf_lock);
         Patch in this message

    Get diff 1

                                                                                                                  dws->dma_mapped = 0;
```

```
Subject [PATCH 4.14 23/89] spi: spi-dw: Add lock protect dw_spi rx/tx to prevent concurrent calls ©
                        Mon, 3 Feb 2020 16:19:08 +0000
 [ Upstream commit 19b61392c5a852b4e8a0bf35aecb969983c5932d ]
\label{local_dw_spi_irq} \mbox{dw\_spi\_transfer\_one concurrent calls.}
 I find a panic in dw_writer(): txw = *(u8 *)(dws->tx), when dw->tx==null, dw->len==4, and dw->tx_end==1.
When tpm driver's message overtime dw spi irq() and dw_spi transfer one may concurrent visit dw_spi, so I think dw_spi structure lack of protection.
Otherwise dw_spi_transfer_one set dw rx/tx buffer and then open irq, store dw rx/tx instructions and other cores handle irq load dw rx/tx instructions may out of order.
                     [ 1025.32131] cara kexec+0x98/0x148 [ 1025.321323] panic+0x17c/0x314 [ 1025.321323] die+0x29c/0x2e8 [ 1025.321331] die kernel fault+0x89/0xb0 [ 1025.321331] die kernel fault+0x99/0xb0 [ 1025.321341] do kernel fault+0x99/0xb0 [ 1025.321341] do translation fault+0x88/0x50 [ 1025.321349] do_mem_abort+0x68/0x18 [ 1025.321349] do_mem_abort+0x68/0x18 [ 1025.321351] ell da+0x20/0x80 [ 1025.321362] dw_writer+0x88/0x10 [ 1025.321364] inherrupt_transfer+0x60/0x110 [ 1025.321365] dw_spi_irq+0x48/0x70
 Signed-off-by: wuxu.wu <wuxu.wu@huawei.com>
Link: https://lore.kernel.org/r/1577849981-31489-1-git-send-email-wuxu.wu@huawei.com
Signed-off-by: Mark Brown <br/>Signed-off-by: Sashalewin <sashalekernel.org>
   drivers/spi/spi-dw.c | 15 ++++++++++--
drivers/spi/spi-dw.h | 1 +
2 files changed, 13 insertions(+), 3 deletions(-)
diff --git a/drivers/spi/spi-dw.c b/drivers/spi/spi-dw.c
index b217c22ff72fe..b461200871f89 100644
--- a/drivers/spi/spi-dw.c
+++ b/drivers/spi/spi-dw.c
@@ -180,9 +180,11 @@ static inline u32 rx_max(struct dw_spi *dws)
+ spin_lock(&dws->buf_lock);
+ max = rx max(dws);
while (max--) {
    rxw = dw read io reg(dws, DW SPI_DR);
    /* Care rx only if the transfer's original "rx" is not null */
@@ -212,6 +217,7 @@ static void dw_reader(struct dw_spi *dws)
   static void int error_stop(struct dw_spi *dws, const char *msg)
@ -284,18 +290,20 @@ static int dw_spi_transfer_one(struct spi_master *master,
                   struct dw spi *dws = spi_master get_devdata(master);
struct chip_data *chip = spi_get_ctldata(spi);
unsigned long flags;
u8 imask = 0;
u16 txlevel = 0;
u23 cr0;
int ret;
                   spin lock irgsave(&dws->buf lock, flags);
dws->tx = (void *)transfer->tx buf;
dws->tx end = dws->tx + transfer->len;
dws->rx = transfer->rx buf;
dws->rx end = dws->xx + transfer->len;
dws->len = transfer->len;
dws-len = transfer->len;
spin_unlock_irgrestore(&dws->buf_lock, flags);
                   spi enable chip(dws, 0);
@@ -486,6 +494,7 @@ int dw spi add host(struct device *dev, struct dw spi *dws)
    dws->type = SSI_MOTO_SFI;
    dws->dma_inited = 0;
    dws->dma_dadr = (dma_addr_t)(dws->paddr + DW_SPI_DR);
    spin_lock_init(&dws->buf_lock);
                   ret = request_irq(dws->irq, dw_spi_irq, IRQF_SHARED, dev_name(dev),
ret = request_irq(dws->irq, dw_spi_irq, IRQF_SHAF
diff --git a/drivers/spi/spi-dw.h b/drivers/spi/spi-dw.h
index 5c07cf8f19e00. 45fbf3ad59lcc 100644
--- a/drivers/spi/spi-dw.h
+++ b/drivers/spi/spi-dw.h
00 -117,6 +117,7 00 struct dw_spi {
                   size_t
void
                                                                           *tx;
*tx_end;
buf_lock;
*rx;
                    void
                    spinlock t
                    void
```

*rx; *rx_end; dma_mapped;

void int

2.20.1

(0.09

Last update: 2020-02-03 17:29 [W:0.014 / U:2.220 seconds] ©2003-2020 Jasper Spaans|hosted at Digital Ocean and TransIP|Read the blog|Advertise on this site