The COBRA Device Driver

- Host-to-target communications protocol
- PLX-9054 virtual serial driver
- COBRA device driver
PLX-9054 Messaging Unit

Host  
PCI bus

MBOX0
write
read

MBOX1
write
read

P2LDBELL
write-1 to set
read, write-1 to clear

L2PDBELL
write-1 to set
read, write-1 to clear

Target
Local bus

PLX-9054
INTA#
write-1 to clear
Host-to-target transmission:

- **Host**
  - Write data to MBOX0, and write RX_READY to P2LDBELL
  - "Transmitter Empty ISR"
  - ISR: read and clear L2PDBELL

- **Target**
  - ISR: read MBOX0, read and clear P2LDBELL
  - "Receiver Ready ISR"
  - Write TX_EMPTY to L2PDBELL
Simultaneous transmissions:

- **Host**
  - send DATA(H2T)
  - read MBOX1
  - send TX_EMPTY
  - read and clear L2PDBELL
  - "Receiver Ready ISR"

- **Target**
  - send DATA(T2H)
  - read MBOX0, read and clear P2LDBELL
  - "Receiver Ready ISR"

- **INTA#**
  - ISR

- **MBOX0**
  - DATA (H2T)

- **P2LDBELL**
  - RX_READY
  - TX_EMPTY

- **LINT#**
  - ISR

- **MBOX1**
  - DATA (T2H)

- **L2PDBELL**
  - RX_READY
  - TX_EMPTY

- "Transmitter Empty ISR"
  - read and clear L2PDBELL

- "Receiver Ready ISR"
  - read and clear P2LDBELL
Simultaneous transmissions (slow target):

Host

INTA#

MBOX0

P2LDBELL

LINT#

MBOX1

L2PDBELL

Target

send DATA

ISR

"Receiver Ready ISR"

read MBOX1, read and clear L2PDBELL

send TX_EMPTY

"Transmitter Empty ISR"

read and clear L2PDBELL

"Receiver Ready and Transmitter Empty ISR"
The COBRA driver

User-space applications
write()
read()
mmap()

/dev/cobra_control#
write()
read()
mmap()

/dev/cobra_stdio#
write()
read()

/dev/cobra_plx#
write()
read()
mmap()

/dev/cobra_download#
write_buffer
write()

/read_buffer
read()

/read_buffer
read()

/read_buffer
read()

//cli_stdio#
write_buffer
write()

/stdin_buffer
read()

/stdout_buffer
read()

/stderr_buffer
read()

/monitor_buffer
read()

/data_buffer
read()
Device SDRAM regions
Write life-cycle:

1. alloc
2. add
3. alloc request
4. free entry
5. remove
6. free request
7. schedule work: write_request
8. send RX_READY
9. schedule work: write_ready
10. schedule work: write_done
11. receive TX_EMPTY or TX_EMPTY timeout
12. schedule work: transaction_done
13. schedule work: transaction_start
14. transfer to SDRAM
15. schedule work: transaction_start
16. add
17. alloc request
18. remove
19. schedule work: write_request
20. schedule work: write_done
21. schedule work: write_error
22. schedule work: transaction_start
23. transaction request list
24. transaction request cache
25. write buffer list
26. buffer entry cache
27. kernel page allocator
Read life-cycle:

read() → copy_to_user

buffer entry cache

free

kernel page allocator

alloc

read list

read list was full

read buffer list

remove

transaction request cache

alloc

schedule work: read_request

RX_READY

transfer from SDRAM

add

transaction request list

remove

call-back

schedule work: read_done

transfer done

send TX_EMPTY

transaction_start

schedule work: transaction_start

transaction_done

schedule work: transaction_done
Host-to-target download (write):

```
dd if=test.bin of=/dev/cobra_download15 bs=128 count=1
```

The target (the PCI bridge) delivers data in 4 x 8-word (32-byte) bursts
Host-to-target upload (read):

```
    dd if=/dev/cobra_download15 of=junk.bin bs=128 count=1
```

The target (the PCI bridge) accepts data in single 32-word (128-byte) burst.