
SourceForge
Search
Browse Enterprise Blog Deals Help
Log In or Join

Solution Centers

Go Parallel
Resources Newsletters

- Home
- Browse
- OpenOCD - Open On-Chip Debugger
- Mailing Lists

OpenOCD - Open On-Chip Debugger

The Open On-Chip Debugger

Brought to you by: dail, gowinex, ntfreak

- Summary
- Files
- Reviews
- Support
- News
- Donate
- Mailing Lists
- Tickets
- Code
- Gerrit Review

Search Mailing Lists

- openocd-commit
- openocd-devel
- openocd-user

Re: [OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?

Re: [OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?

From: Justin Drake <drakefjustin@gm...> - 2012-03-23 10:00:09

```
Thanks Andreas, that was helpful. But I do not understand how this is
harmless. If GDB does not perform exception return then I cannot
follow the execution of my program by doing the command ni (next
instruction) continuously.

On Thu, Mar 22, 2012 at 6:12 PM, Andreas Fritiofson
<andreas.fritiofson@...> wrote:
> On Thu, Mar 22, 2012 at 5:06 PM, Justin Drake <drakefjustin@...> wrote:
>> I am getting the following error on the OpenOCD terminal:
>>
>> Error: address + size wrapped(0xfffffffe, 0x00000004)
>>
>> What address is referred to? What do the two addresses 0xfffffffe and
>> 0x00000004 correspond to?
>
> In ARMv7-M, 0xfffffffX is a special value for the return address that
> is placed in the link register when an exception is taken. When the
> exception handler returns and the PC gets loaded with this special
> value, the core knows that it should perform exception return instead
> of a normal branch. This is one of the nifty features that makes it
> possible to write exception handlers as regular C functions.
>
```

```
> However, when GDB follows the stack trace from inside an exception
> handler, it ignores the fact that 0xfffffffX is a special value and
> tries to load a word (0x00000004 bytes) from that address as if it was
> a regular return address. OpenOCD warns that the read wraps around the
> 32-bit address space, and also emits some MEM_AP errors from trying to
> access these locations. Upon receiving the errors, GDB assumes it has
> reached the start of the stack trace and stops.
>
> The error is completely harmless but could be avoided if GDB was made
> aware of the special value and how to get the true return address.
>
> /Andreas
```

Thread view

[OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?

From: Justin Drake <drakefjustin@gm...> - 2012-03-22 16:07:33

```
I am getting the following error on the OpenOCD terminal:

Error: address + size wrapped(0xfffffffe, 0x00000004)

What address is referred to? What do the two addresses 0xfffffffe and
0x00000004 correspond to?
```

Re: [OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?

From: simonqian.openocd <simonqian.openocd@gm...> - 2012-03-22 17:41:26

Attachments: Message as HTML

```
I guess you are trying to access 0xFFFFFFFF as a 32-bit instruction or data, so the debugger will access 0xFFFFFFFE(0xFFFFFFFF & ~3).
You should provide more debug message.




simonqian.openocd

From: Justin Drake
Date: 2012-03-23 00:06
To: openocd-devel
Subject: [OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?
I am getting the following error on the OpenOCD terminal:

Error: address + size wrapped(0xfffffffe, 0x00000004)

What address is referred to? What do the two addresses 0xfffffffe and
0x00000004 correspond to?

------------------------------------------------------------------------------
This SF email is sponsosred by:
Try Windows Azure free for 90 days Click Here
http://p.sf.net/sfu/sfd2d-msazure
_______________________________________________
OpenOCD-devel mailing list
OpenOCD-devel@...
https://lists.sourceforge.net/lists/listinfo/openocd-devel
```

Re: [OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?

From: Andreas Fritiofson <andreas.fritiofson@gm...> - 2012-03-22 18:40:25

```
On Thu, Mar 22, 2012 at 5:06 PM, Justin Drake <drakefjustin@...> wrote:
> I am getting the following error on the OpenOCD terminal:
>
> Error: address + size wrapped(0xfffffffe, 0x00000004)
>
> What address is referred to? What do the two addresses 0xfffffffe and
> 0x00000004 correspond to?

In ARMv7-M, 0xfffffffX is a special value for the return address that
is placed in the link register when an exception is taken. When the
exception handler returns and the PC gets loaded with this special
value, the core knows that it should perform exception return instead
of a normal branch. This is one of the nifty features that makes it
possible to write exception handlers as regular C functions.

However, when GDB follows the stack trace from inside an exception
handler, it ignores the fact that 0xfffffffX is a special value and
tries to load a word (0x00000004 bytes) from that address as if it was
a regular return address. OpenOCD warns that the read wraps around the
32-bit address space, and also emits some MEM_AP errors from trying to
access these locations. Upon receiving the errors, GDB assumes it has
```

reached the start of the stack trace and stops.

The error is completely harmless but could be avoided if GDB was made aware of the special value and how to get the true return address.

/Andreas

Re: [OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?

From: Justin Drake <drakefjustin@gm...> - 2012-03-23 10:00:09

Thanks Andreas, that was helpful. But I do not understand how this is harmless. If GDB does not perform exception return then I cannot follow the execution of my program by doing the command ni (next instruction) continuously.

On Thu, Mar 22, 2012 at 6:12 PM, Andreas Fritiofson
<andreas.fritiofson@...> wrote:

> On Thu, Mar 22, 2012 at 5:06 PM, Justin Drake <drakefjustin@...> wrote:
>> I am getting the following error on the OpenOCD terminal:
>>
>> Error: address + size wrapped(0xfffffffe, 0x00000004)
>>
>> What address is referred to? What do the two addresses 0xfffffffe and 0x00000004 correspond to?
>
> In ARMv7-M, 0xfffffffX is a special value for the return address that is placed in the link register when an exception is taken. When the exception handler returns and the PC gets loaded with this special value, the core knows that it should perform exception return instead of a normal branch. This is one of the nifty features that makes it possible to write exception handlers as regular C functions.
>
> However, when GDB follows the stack trace from inside an exception handler, it ignores the fact that 0xfffffffX is a special value and tries to load a word (0x00000004 bytes) from that address as if it was a regular return address. OpenOCD warns that the read wraps around the 32-bit address space, and also emits some MEM_AP errors from trying to access these locations. Upon receiving the errors, GDB assumes it has reached the start of the stack trace and stops.
>
> The error is completely harmless but could be avoided if GDB was made aware of the special value and how to get the true return address.
>
> /Andreas

Re: [OpenOCD-devel] What type of memory is referred to in "Error: address + size wrapped"?

From: Andreas Fritiofson <andreas.fritiofson@gm...> - 2012-03-23 10:45:10

On Fri, Mar 23, 2012 at 10:59 AM, Justin Drake <drakefjustin@...> wrote:

> Thanks Andreas, that was helpful. But I do not understand how this is harmless. If GDB does not perform exception return then I cannot follow the execution of my program by doing the command ni (next instruction) continuously.

GDB never performs exception return, the core does it. GDB is just not aware beforehand where the PC will end up after stepping the 'bx lr' instruction in an exception handler. The core happily executes the instruction though, performs exception unstacking and so on, and afterwards GDB finds out what really happened.

The only limitation I'm aware of is that GDB can't show the stack trace properly, it doesn't show the stack frames below an exception frame. Which also means GDB can't step out of an exception handler (finish) because it doesn't know where to put the breakpoint. Single stepping works fine though. These limitations are in GDB, not OpenOCD, which is why I said that OpenOCD's "error" message really is harmless.

Of course, we could probably fake the reply to GDB when we detect it's tracing the stack and fixup the values so GDB can make sense of it, but that would be outright lying.

/Andreas

SourceForge

About Site Status @sfnet_ops

Powered by

Apache Allura™

Find and Develop Software

Create a Project Software Directory Top Downloaded Projects

Community

Blog @sourceforge Resources

Help

Site Documentation Support Request Real-Time Support