Debugger for Patmos

Special Course
Prof. Martin Schoeberl

Mariana Santos – s233360

What was the project about?

- Debuggers
- Patmos

What will I talk about?

- GDB Remote Serial Protocol
- Implementation
- Results
- Future work

GDB Remote Serial Protocol (RSP)

How to use it?

Remote debugging

Host machine → development machine

Target machine → patmos on the FPGA

How to start implementing?

Description of the target's architecture

Implement stub functions on the target

How is communication realised?

Communication medium

Serial communication over UART

Packet with ASCII characters

Program needs to wait

Handler for traps/exceptions
Will receive commands from GDB

Invoke a trap for initial communication

Implementation

Defining architecture

MIPS architecture was used

32-bit 32 general purpose registers
32-bit instructions

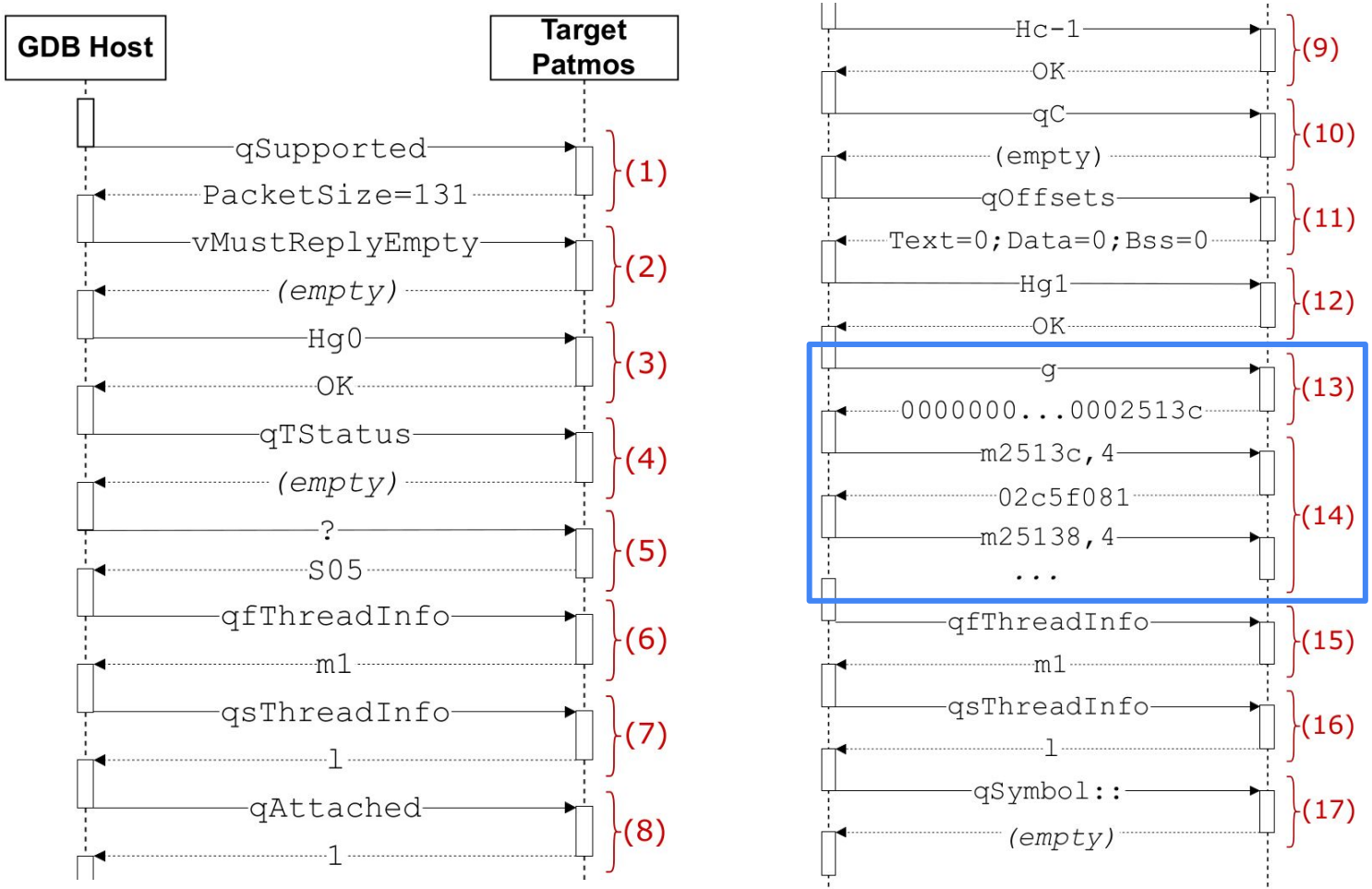
Not feasible to implement a proper one

What happens when an interrupt occurs?

An interrupt is triggered upon starting the program

Packets will be exchanged with GDB

Setup


GDB Host
Target Patmos
qSupported
PacketSize=131
(1)
vMustReplyEmpty
(empty)
(2)
Hg0
OK
(3)
qTStatus
(empty)
(4)
?
S05
(5)
qfThreadInfo
m1
(6)
qsThreadInfo
l
(7)
qAttached
1
(8)
Hc-1
OK
(9)
qC
(empty)
(10)
qOffsets
Text=0;Data=0;Bss=0
(11)
Hg1
OK
(12)
g
0000000...0002513c
m2513c,4
02c5f081
m25138,4
...
(13)
(14)
qfThreadInfo
m1
(15)
qsThreadInfo
l
(16)
qSymbol::
(empty)
(17)

What about the g and m packets?

g

Obtain register values

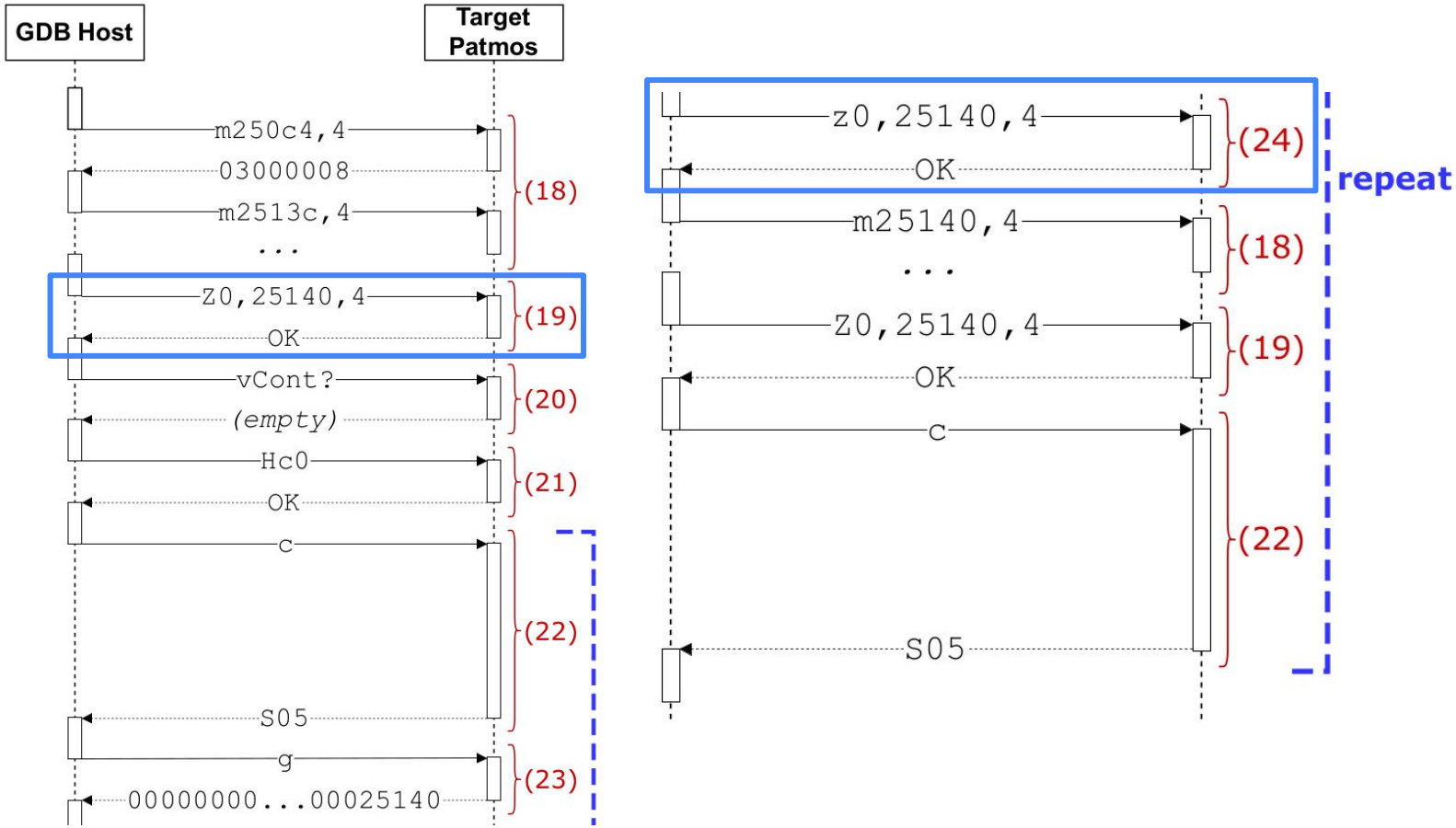
0 r0	1 r1	2 r2	3 r3	4 r4	5 r5	6 r6	7 r7
8 r8	9 r9	10 r10	11 r11	12 r12	13 r13	14 r14	15 r15
16 r16	17 r17	18 r18	19 r19	20 r20	21 r21	22 r22	23 r23
24 r24	25 r25	26 r26	27 r27	28 r28	29 r29	30 r30	31 r31
32 sr	33 lo	34 hi	35 bad	36 cause	37 pc		

m (m2513c,4)

Read from memory address

Addresses correspond to **instructions**

"next"


GDB Host
Target Patmos
m250c4,4
03000008
m2513c,4
...
(18)
Z0,25140,4
OK
(19)
vCont?
(empty)
(20)
Hc0
OK
(21)
c
(22)
S05
g
00000000...00025140
(23)
z0,25140,4
OK
(24)
repeat
m25140,4
...
(18)
Z0,25140,4
OK
(19)
c
(22)
S05

What about the Z0 and z0 packets?

Z0 (Z0,25140,4)

Replace instruction with a **trap**

Invalidate instruction cache

z0 (z0,25140,4)

Return original instruction

Invalidate instruction cache

Results

A few GDB commands...

```
break <line number>

break *<instruction address>

continue

step

next

info locals

info registers
```

Results overview

Some commands need debugging information

Breakpoints are set

continue, **next** and **step** execute until breakpoint

If there’s no breakpoint, doesn’t end

Breakpoint instructions are never executed

What about the Z0 and z0 packets?

Z0 (Z0,25140,4)

Replace instruction with a **trap**

If instruction is 64-bit, skip to the next instruction

z0 (z0,25140,4)

Return original instruction

Address will be the PC

When execution returns, PC+4 will be the first instruction to be executed

Conclusion & Future work

What needs to be done?

Describe Patmos to GDB

Handle the issue with the **z0** packet

Add debugger information in the compiled binaries

Conclusion

Currently not a usable debugging solution

However:

- Successful communication
- Setup debugging session
- Packet handling
- Setting breakpoints
- Information on register values