

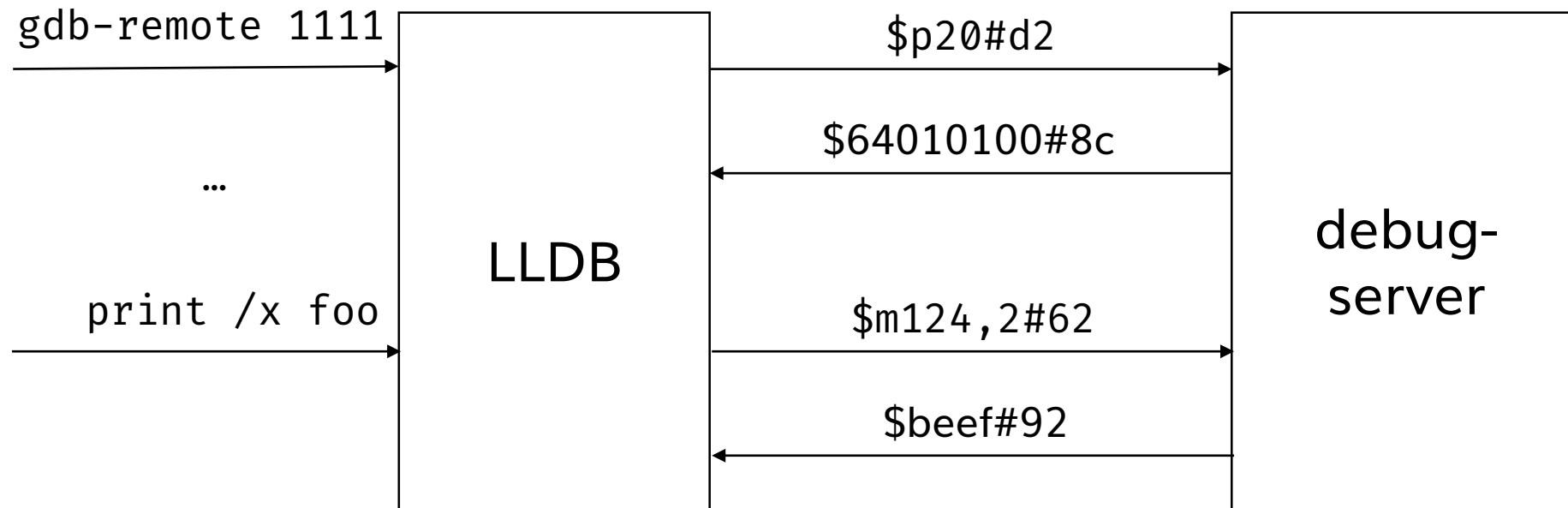


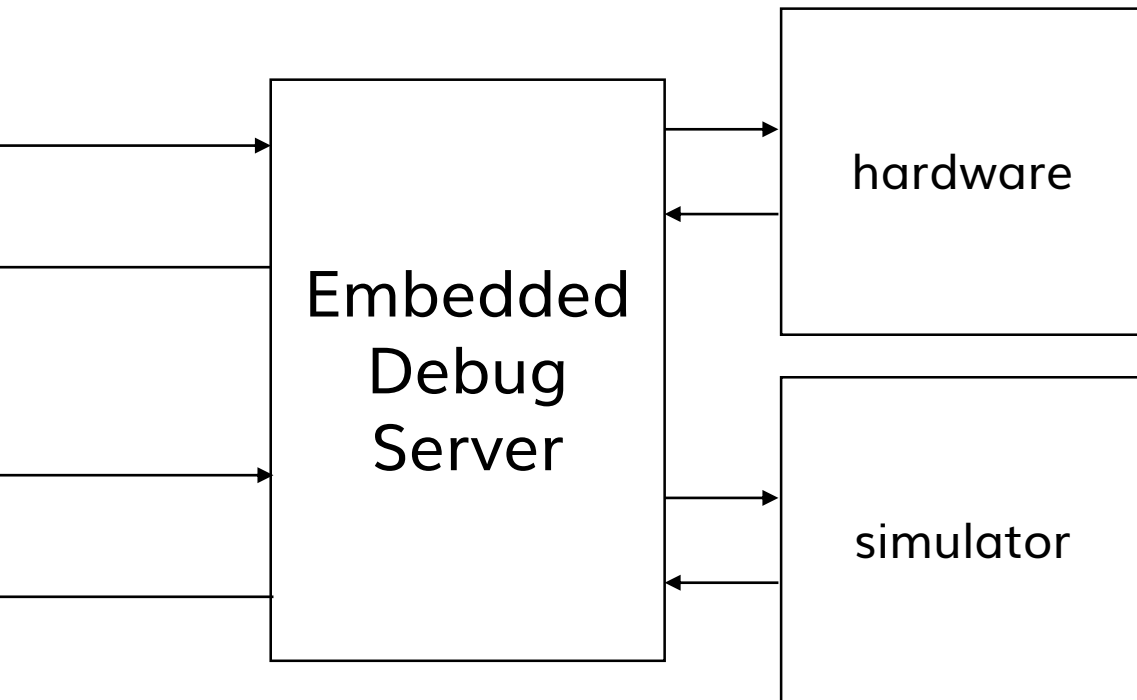
# A Unified Debug Server for Deeply Embedded Systems and LLDB

Simon Cook



Copyright © 2019 Embecosm.  
Freely available under a Creative Commons license.



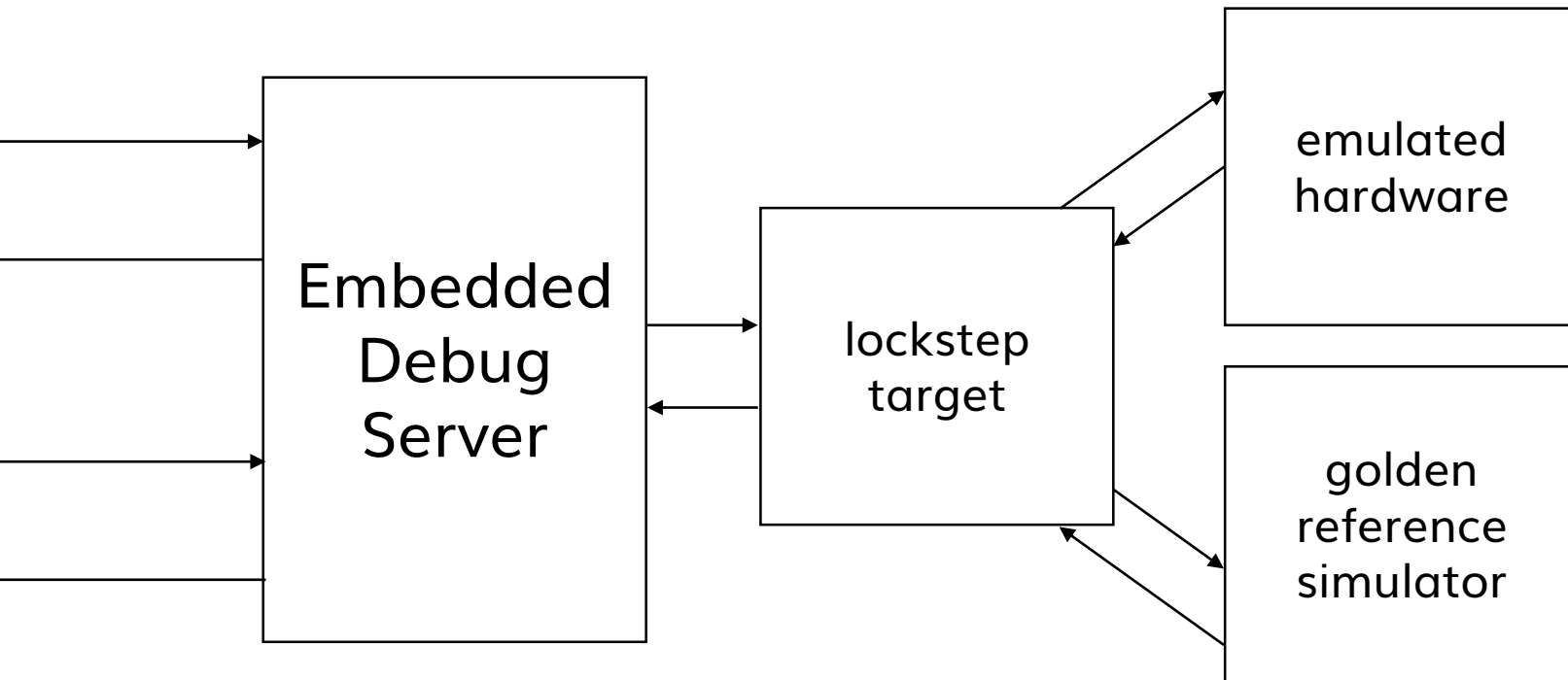


```

class ExampleTarget : public Target {
public:
    // Timers
    uint64_t getCycleCount() const;
    uint64_t getInstrCount() const;

    // Read-write memory and registers
    std::size_t readRegister(const int reg,
                             uint_reg_t &value);
    std::size_t writeRegister(const int reg,
                              const uint_reg_t value);
    std::size_t read(const uint_addr_t addr,
                    uint8_t *buffer,
                    const std::size_t size);
    std::size_t write(const uint_addr_t addr,
                     const uint8_t *buffer,
                     const std::size_t size);

    // Execution control
    bool prepare(const std::vector<ResumeType> &actions);
    bool resume(void) override;
    WaitRes wait(std::vector<ResumeRes> &results);
    ...
};
  
```



```
std::size_t
LockstepTarget::readRegister(const int reg,
                             uint_reg_t &value) {
    uint_reg_t vL;
    uint_reg_t vR;
    std::size_t rL = _l->readRegister(reg, vL);
    std::size_t rR = _r->readRegister(reg, vR);

    // Report inconsistency server side.
    if ((rL != rR) || (vL != vR))
        std::cerr << "Lockstep error: register"
                    << " inconsistency."
                    << std::endl;

    // TODO: Allow this to be configured.
    // For now we just choose the left value
    // since readRegister cannot fail.
    value = vL;
    return rL;
}
```

# Lockstep

```
$ lldb
(lldb) gdb-remote 51000
Process 1 stopped
* thread #1, stop reason = signal SIGTRAP
frame #0: 0x0001015c
-> 0x1015c: addi a5, zero, 5
0x10160: mv t6, a5
0x10164: mv a5, zero
0x10168: mv a0, a5
(lldb) si
Process 1 stopped
* thread #1, stop reason = instruction step into
frame #0: 0x00010160
-> 0x10160: mv t6, a5
0x10164: mv a5, zero
0x10168: mv a0, a5
0x1016c: lw s0, 12(sp)
(lldb) si
Process 1 stopped
* thread #1, stop reason = signal SIGSYS
frame #0: 0x00010164
-> 0x10164: mv a5, zero
0x10168: mv a0, a5
0x1016c: lw s0, 12(sp)
0x10170: addi sp, sp, 16
```

SIGSYS returned, indicating divergence

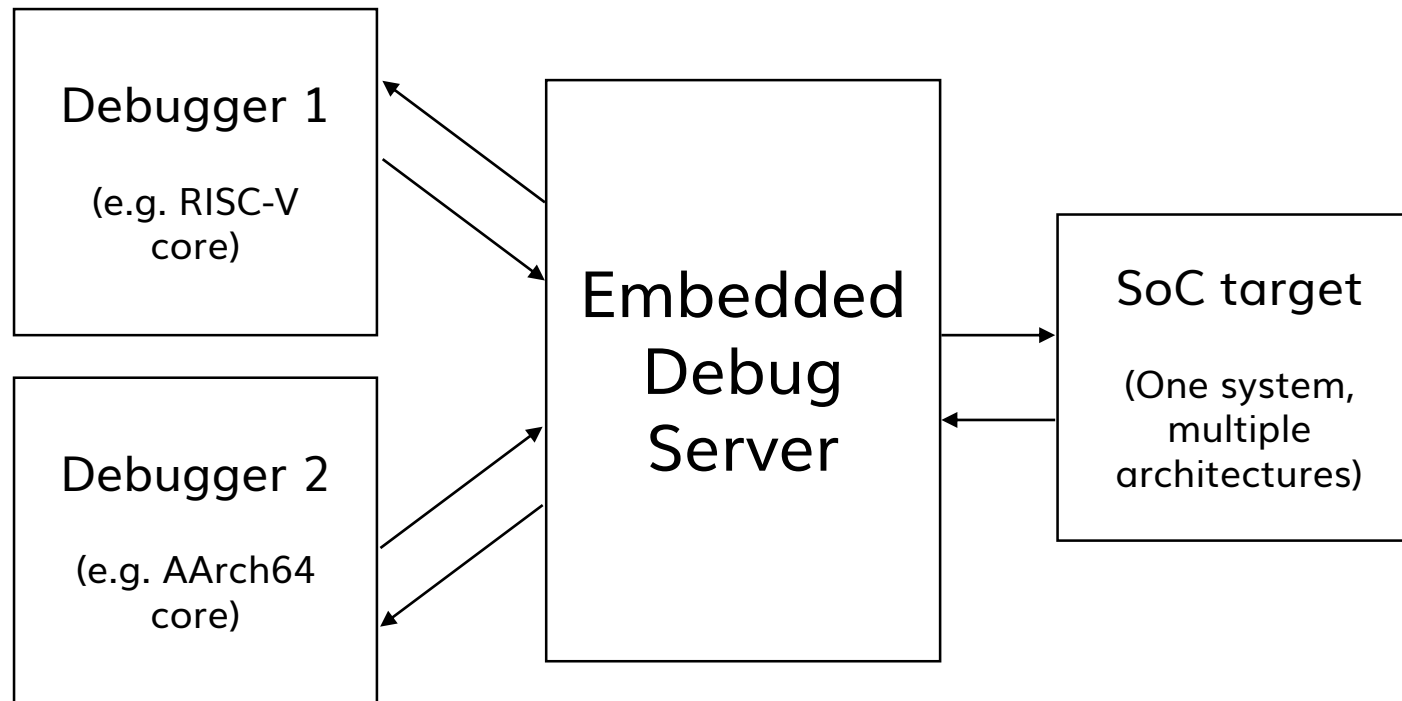
```
(lldb) register read --all
general:
x0 = 0x00000000
x1 = 0x000100d8
x2 = 0xffffffff0
x3 = 0x00011da8
< ... >
x28 = 0x00000000
x29 = 0x00000000
x30 = 0x00000000
pc = 0x00010164
1 registers were unavailable.
```

x31 not shown, since both targets disagree on value

```
lockstep-left:
x31-left = 0x00000005
32 registers were unavailable.
```

```
lockstep-right:
x31-right = 0x00000004
32 registers were unavailable.
```

both targets x31 shown, allowing investigation into divergence





# Questions?

[www.embecoscsm.com](http://www.embecoscsm.com)



Copyright © 2019 Embecoscsm.  
Freely available under a Creative Commons license.