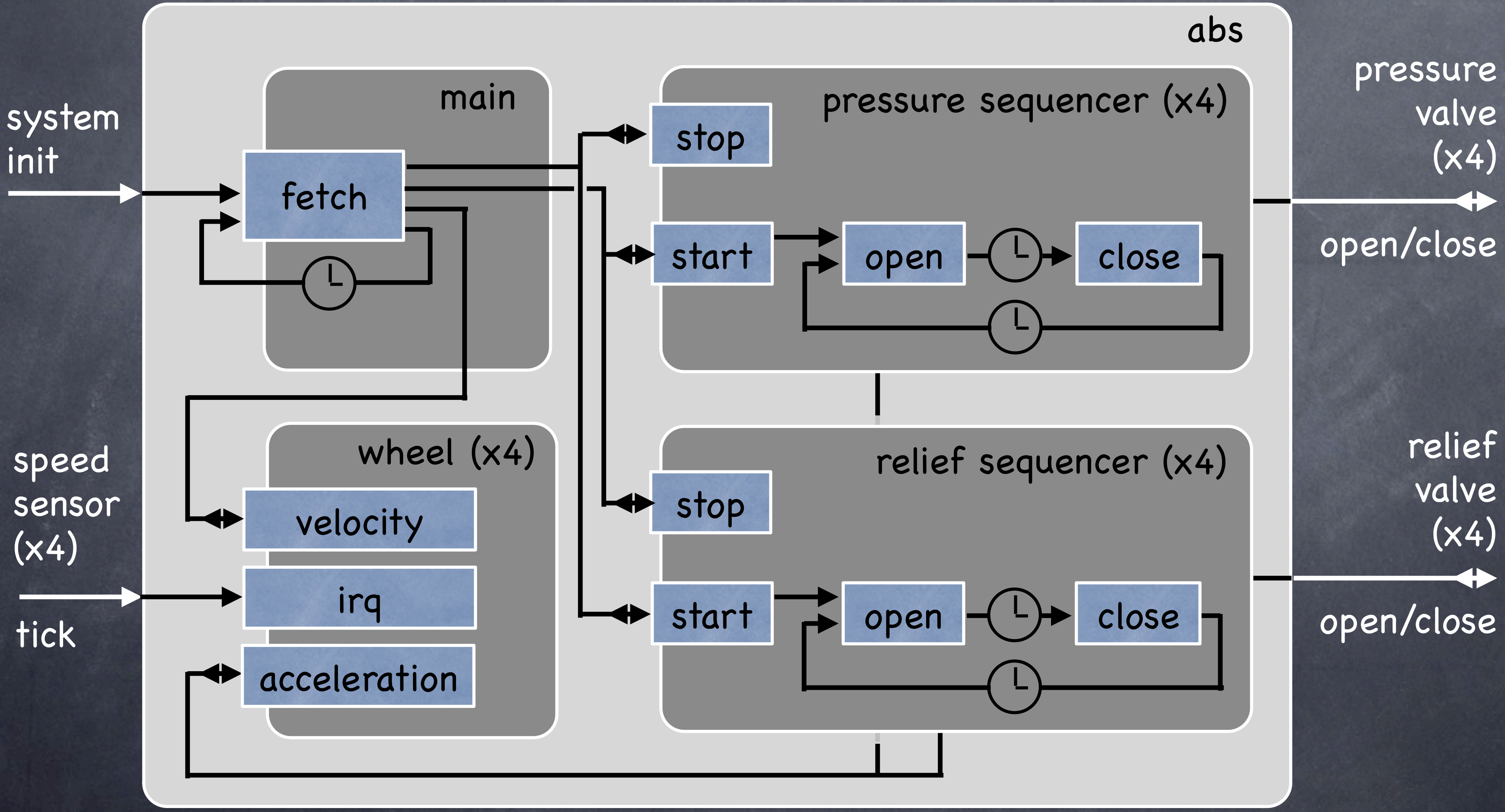
An ABS in Timber



An ABS in Timber

```
relief wheel valve = class
    msg := null
    stop = request
        abort msg
        valve.write cCLOSED
    start = request
        abort msg
        msg := send open
    open = action
        a <- wheel.acceleration
        if a < 0 then
            valve.write cOPEN
            msg := send after (millisec (a*10)) close
        else
            msg := send after (millisec 5) open
    close = action
        valve.write cCLOSED
        msg := send after (millisec 5) open
    result Sequencer {..}
```

Encapsulated state variable

Synchronous method

Asynchronous call

Mutation

Asynchronous method

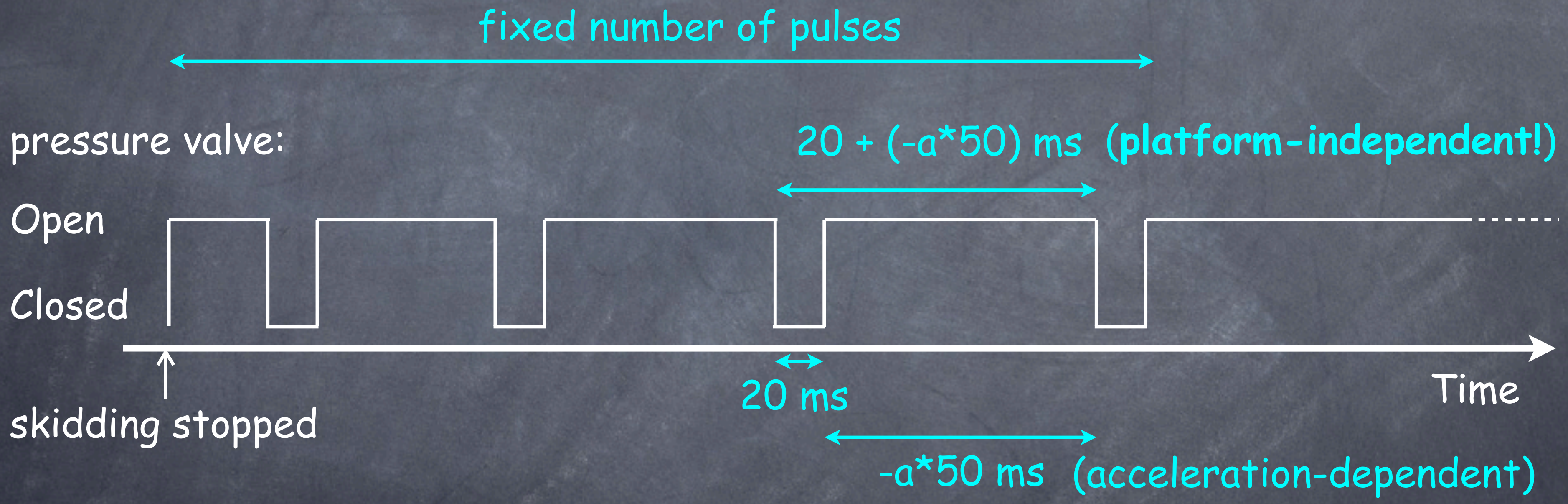
Synchronous call

Delayed asynchronous call

Returned object interface

An ABS in Timber

Reapplying brake pressure



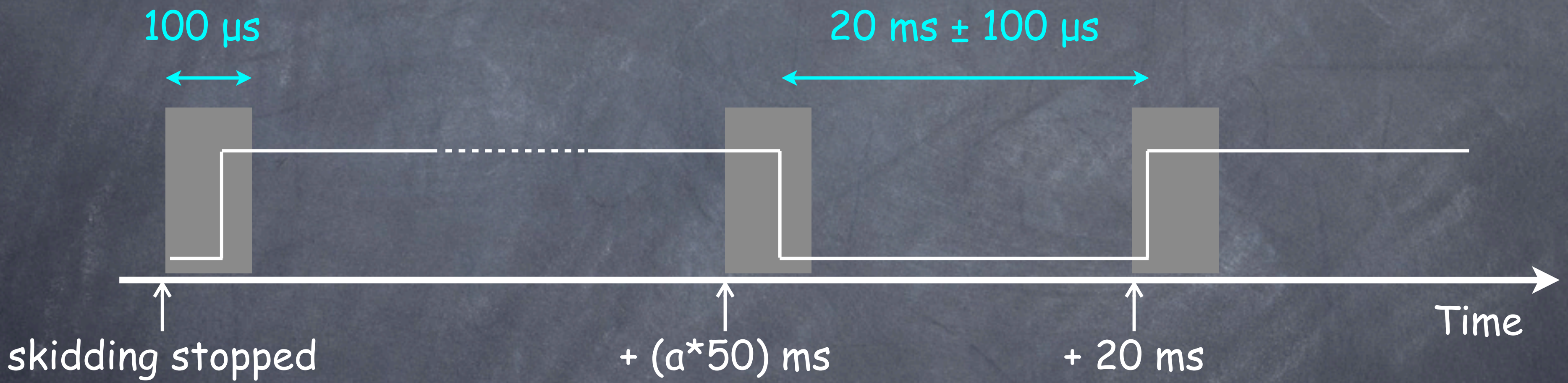
fixed number of pulses
pressure valve:
20 + (-a*50) ms (**platform-independent!**)
Open
Closed
skidding stopped
20 ms
Time
-a*50 ms (acceleration-dependent)

```
after (millisec 20) (open n)

    a <- wheel.acceleration
    after (millisec (-a*50)) (close (n-1))
```

An ABS in Timber

Reapplying brake pressure



open = **before** microsec 100 **action** ...

close = **before** microsec 100 **action** ...

open = **before** microsec 100 **action** ...