Anvil *HDL*

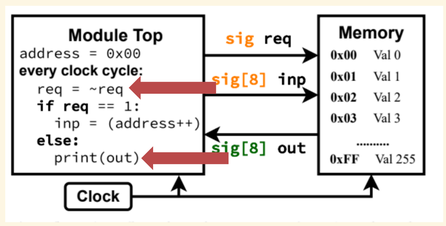
A General-Purpose Timing-Safe Hardware Description Language

Jason Zhijingcheng Yu*, Aditya Ranjan Jha*, Umang Mathur, Trevor E. Carlson, Prateek Saxena

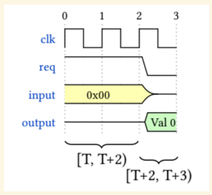
Department of Computer Science, National University of Singapore (*Equal contribution)

NUS National University of Singapore

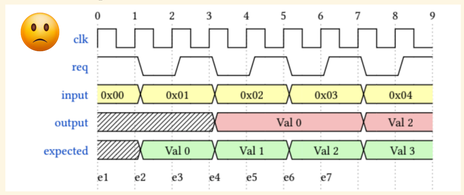
Motivation: Timing Hazards



Protocol



Reality



Scenarios of Timing Hazards

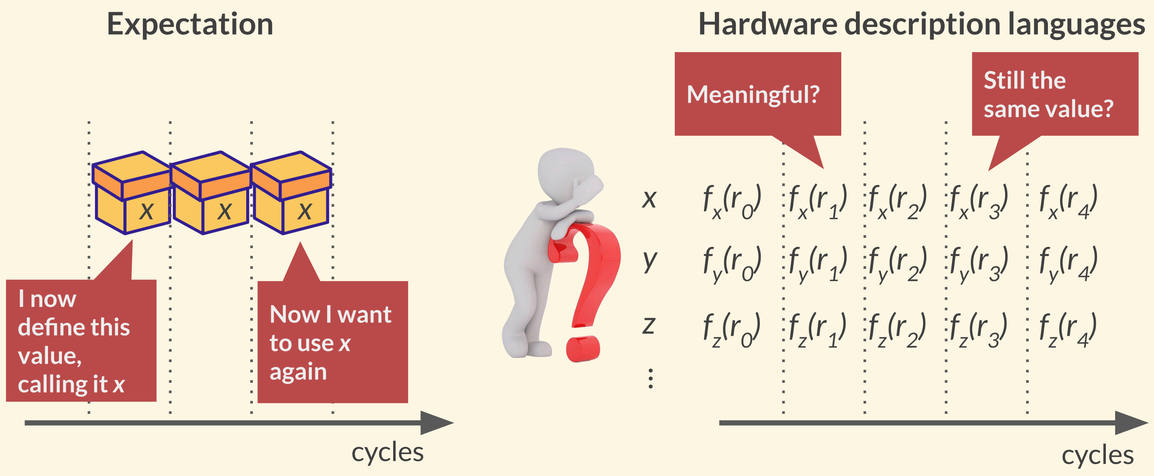
Scenario 1:

- **Expected:** Same value across cycles
- **Reality:** Value changes halfway

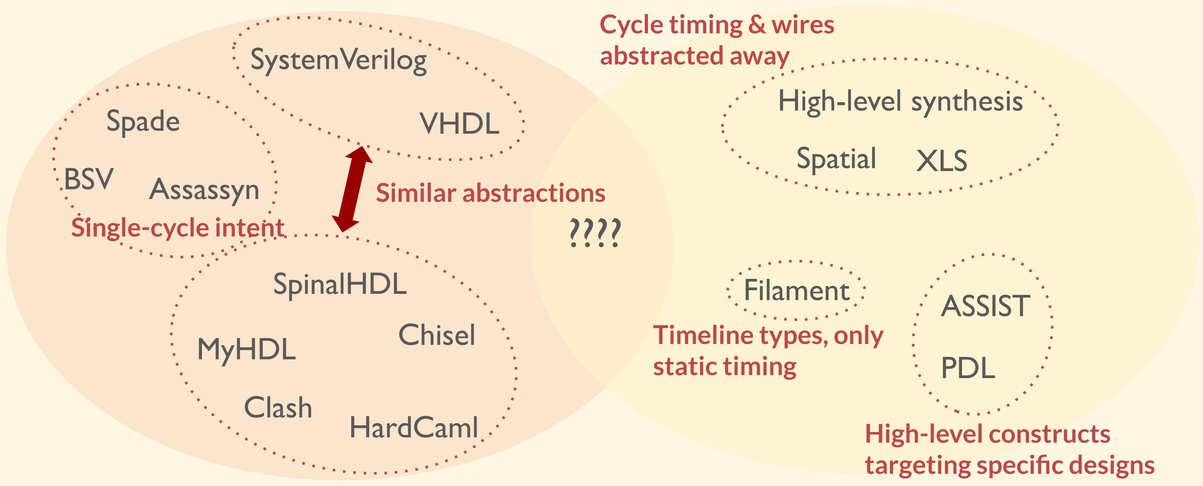
Scenario 2:

- **Expected:** Meaningful value is used
- **Reality:** Garbage value is used

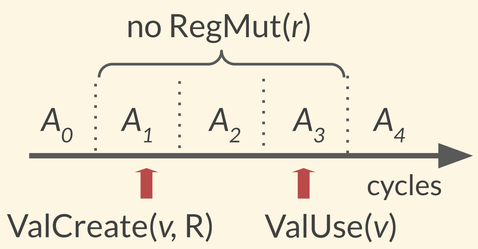
Root Cause: Lacking Abstractions in HDLs



Existing HDL Landscape



Desired Property: Timing Safety

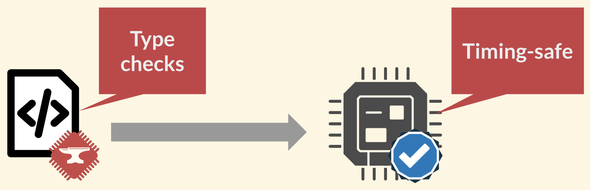


action ∈ A_i

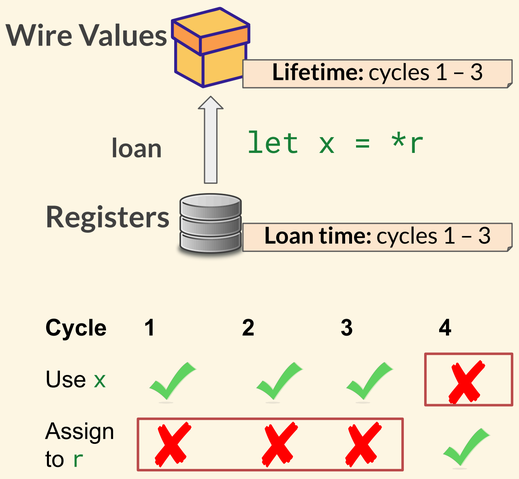
- ValCreate(*v*, R)
- ValUse(*v*)
- RegMut(*r*)

Anvil — A Timing-Safe and General-Purpose HDL

Research Question: Can we get both *timing safety* and *expressiveness for low-level control*?

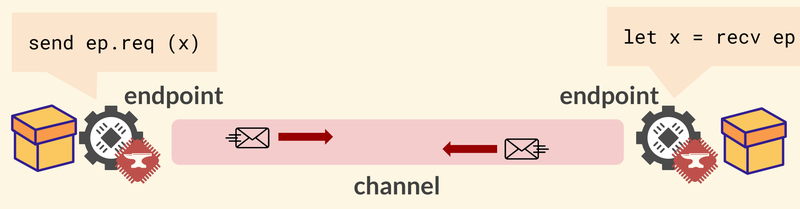


Core Idea: *Lifetime* and *Loan time*

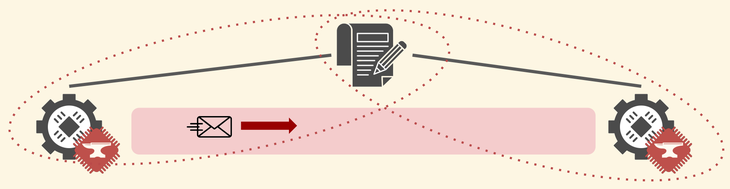



```
set r := 1'b1 >>
let v = *r;
dprint "Value of v is %d" (v);
set r := 1'b0 >>
dprint "Value of v is %d" (v);
dfinish
```

Message Passing Communication



Timing Contracts: Lifetime across Channels



Challenge: How to describe time in different modules?

Static Lifetime Contracts

Observation:

- Message passing defines an agreed-on cycle
- Offset by N cycles is agreed on

```
chan ch {
    left req : (request@#1),
    right resp : (logic[16]@#3)
}
```

Dynamic Lifetime Contracts

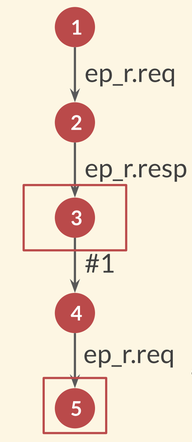
Observation:

- Multiple messages share the same channel

```
chan ch {
    left req : (request@#1),
    right resp : (logic[16]@req)
}
```

Reasoning about Abstract Time — Event Graph

```
chan ch {
    left req : (request@#1),
    right resp : (logic[16]@req)
}
1 send ep_r.req(...) >>
2 let res = recv ep_r.resp >>
3 dprint "result = %d" (res);
3 cycle 1 >>
4 send ep_r.req(...) >>
5 dprint "result = %d" (res);
  ...
```



Lifetime of `res`

e3 — e3 + req

Use of `res` at e5 requires:
e5 be within the lifetime

Implementation of the other module not needed

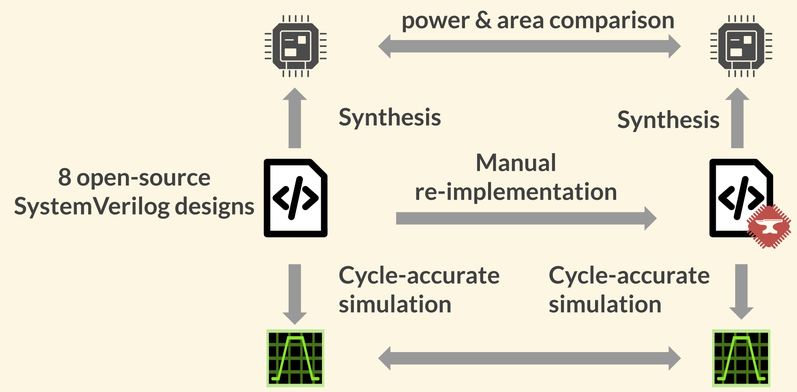
Evaluation

Evaluation Questions

1. **Safety** (via formal proofs)
2. **Expressiveness**
3. **Practicality**

Benchmark Selection

Targeting real-world, diverse, dynamic latency



Hardware Designs	Area (um²)		Power (mW)		f_{max} (MHz, ±50)		Latency (cycles)	
	Baseline	Anvil	Baseline	Anvil	Baseline	Anvil	Baseline	Overhead
FIFO Buffer (SV)	690	674 (−2%)	1.434	1.403 (−2%)	4062	4156	dyn	0
Spill Register (SV)	165	171 (3%)	0.459	0.469 (2%)	5187	5375	dyn	0
Passthrough Stream FIFO (SV)	679	679 (0%)	1.239	1.264 (2%)	4093	3625	1	0
CVA6 Translation Lookaside Buffer (SV)	5561	5611 (0%)	5.813	5.835 (0%)	2468	2406	dyn	0
CVA6 Page Table Walker (SV)	499	561 (12%)	0.649	0.676 (4%)	3531	3281	dyn	0
AES Cipher Core (SV)	9096	9090 (0%)	0.793	0.972 (22%)	781	1229	dyn	0
AXI Lite Demux Router (SV)	1318	1469 (11%)	1.351	1.385 (2%)	2437	2125	dyn	0
AXI Lite Mux Router (SV)	1448	1633 (12%)	1.336	1.324 (0%)	2406	2187	dyn	0
Average overhead compared with SystemVerilog baselines: Area = 4.50%, **Power** = 3.75%								
Pipelined ALU (Filament)	498	404 (−18%)	0.606	0.626 (3%)	3062	4675	1	0
Systolic Array (Filament)	2509	2425 (−3%)	2.493	2.764 (10%)	2400	2862	1	0
Average overhead compared with Filament baselines: Area = −10.5%, **Power** = 6.5%								

Takeaway:

- Cycle-accurate functional behaviour of real-world designs reproduced
- Practical synthesis overhead (vs SystemVerilog designs), averaging 4.50% area and 3.75% power

Beyond Theory: A Practical HDL

Current Capabilities:

- Sync Modes
- Recursive (Pipelining)
- Parametrised Data Types
- SystemVerilog FFI
- Language Server

Future Work:

- Liveness-by-Construction
- Asynchronous Events
- Expressive Lifetimes
- Extensible, Optimised Compiler
- ...

Anvil Playground

Jason

Acknowledgements:
NUS KISP Lab members, anonymous reviewers, Yaswanth Tavva, Sai Dhawal Phaye

Research partly funded by Singapore Ministry of Education Tier 2 grants MOE-T2EP20124-0007 and MOE-T2EP20222-0007