The Three Body Problem
There’s more to building Silicon than EDA currently helps

Peter Birch (Vypercore) & Ben Marshall (PQShield)
What’s the problem?

- ARCHITECTURE
- DESIGN
- VERIFICATION
- PHYSICAL
- MANUFACTURE
What’s the problem?
What’s the problem?

ARCHITECTURE  DESIGN  VERIFICATION  PHYSICAL  MANUFACTURE

PROVENANCE
The Three Bodies

- The ‘big three’ EDA vendors don’t make this easy
- Tools lack common interfaces
- Proprietary inputs and outputs
- Results vary run-to-run
- NDAs stop flows being shared
So what do we do?
SUCCESSFUL

Tenets of a Silicon Engineering Flow

TRACEABLE

TRACEABLE
SUCCESSFUL Tenets of a Silicon Engineering Flow

TRACEABLE

MODULAR
SUCCESSFUL
Tenets of a Silicon Engineering Flow

TRACEABLE

MODULAR

REPRODUCIBLE
Pure Functions

(files, pass) = synthesis(files, switches, environment)
Flows are Graphs
Transformations

INPUTS

SIMULATE

SPECIALISE

INVOKE TOOL

NORMALISE

OUTPUTS
Interfaces

INPUTS

SIMULATE

SPECIALISE

INVOKE TOOL

NORMALISE

OUTPUTS
Interfaces

INPUTS ➔ JSON
Interfaces

INPUTS → JSON

→ RELEASES
→ PARALLEL BUILDS
→ CHECKPOINTS
Tracking References
Makefiles are not enough

- EDA flows are very complex
- Many inputs and outputs
- Deeply branched trees of transformations
- Software build tools are not well suited
So we need something better…
...we have an answer...
…and it’s open source 😤
Blockwork

An opinionated build environment for EDA projects.

Languages
Blockwork

• Tool-agnostic framework
• Written in Python
• Uses Docker to isolate each step
• You bring the tool wrappers, transforms, and flows
• Under active and rapid development by VyperCore
• Inspired by Edalize and FuseSoC
Blockwork

!Testbench
design: !SvMod :vc_fifo
python: !PyMod
top: project.tech.fifo.verif.tb
modules:
  - verif/tb
needs:
  - !PyLib tech/io:verif/stream
  - !PyLib tech/verif/forastero

$> bw wf sim --target tech/fifo --project gen1
[17:02:56] INFO  Running transform: PacktypeTransformSv
[17:02:57] INFO  Running transform: PacktypeTransformPy
[17:02:58] INFO  Running transform: PacktypeTransformPy
[17:02:59] INFO  Running transform: PacktypeTransformSv
[17:03:11] INFO  Running transform: TbCompileTransform

[17:03:11] INFO  Running transform: SimAggregationTransform

<table>
<thead>
<tr>
<th>TEST</th>
<th>SEED</th>
<th>REAL_TIME(s)</th>
<th>SIM_TIME(ns)</th>
<th>STATUS</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>1234</td>
<td>0.0380602</td>
<td>282.001</td>
<td>FAIL</td>
<td>sim.log</td>
</tr>
</tbody>
</table>
Runs here, there, or anywhere...

DEBUG @ 25,000 FT
Blockwork

Please come join in the fun!

github.com/blockwork-eda
In Summary

- A good silicon flow is traceable, modular, and reproducible
- We need better tools!
- Blockwork is tool-agonistic and open source
- Vendors - please help us:
  - Relax NDAs around flows
  - Standardised interfaces
  - Machine readable inputs and outputs
  - Fully reproducible transformations