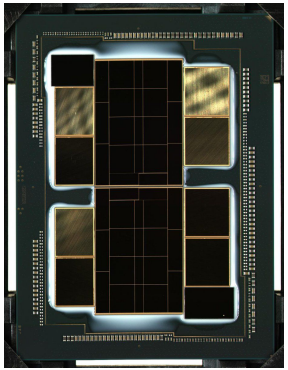
Software-defined design for systems of chiplets

Dr. Duncan Haldane
JITX
June 19th, 2022

Chiplet design work is snowballing

DISAGGREGATION ENABLES

- Increased yields
- 3D stacking
- Reduced time to market
- Mass customizability

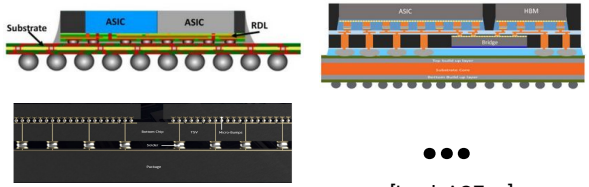


[Intel 2021]

IFF WE CAN SOLVE

- Unfamiliar optimization problems
- Extra power and area for D2D interfaces
- Packaging cost and NRE
- Verification complications

WHILE USING NEW



[Intel, ASE, ...]

Advanced packages

OPEN Compute Project

UCIe Universal Chiplet Interconnect Express

D2D interface standards

Chiplet design work is snowballing

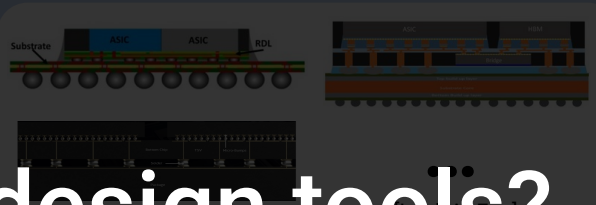
DISAGGREGATION ENABLES

- Increased yields
- 3D stacking
- Reduced time to market
- Heterogeneity

IFF WE CAN SOLVE

- Unfamiliar optimization problems
- Extra power and area for D2D interfaces
- Packaging cost and NRE
- Verification complications

WHILE USING NEW



[Intel 2021]

Advanced packages

OPEN Compute Project

UCIe Universal Chiplet Interconnect Express

D2D interface standards

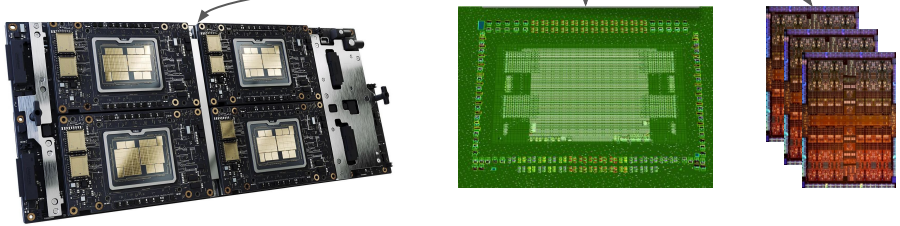
What should we ask of our design tools?

[Intel 2021]

What should we ask of our design tools?

1. Rigorous design optimization integrating:
   1.1. Si design co-optimization
   1.2. Package design co-optimization
   1.3. Board design co-optimization
2. Automated design verification
3. Design reuse across package technologies
4. Enable what-if? analysis

```
$ git clone silicon-ip
$ git clone package-design-kits
$ git clone board-ip
$ make hpc42
```



[Sources Intel, ASE]

jitX

Team

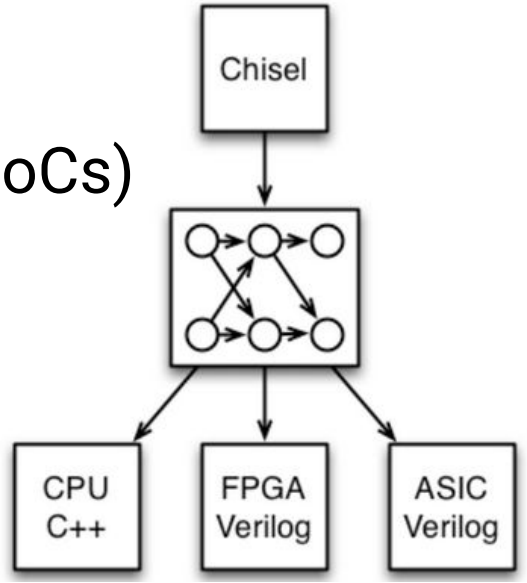


3 PhD founders from UC Berkeley

Prior work

CHISEL

(10x design productivity for SoCs)



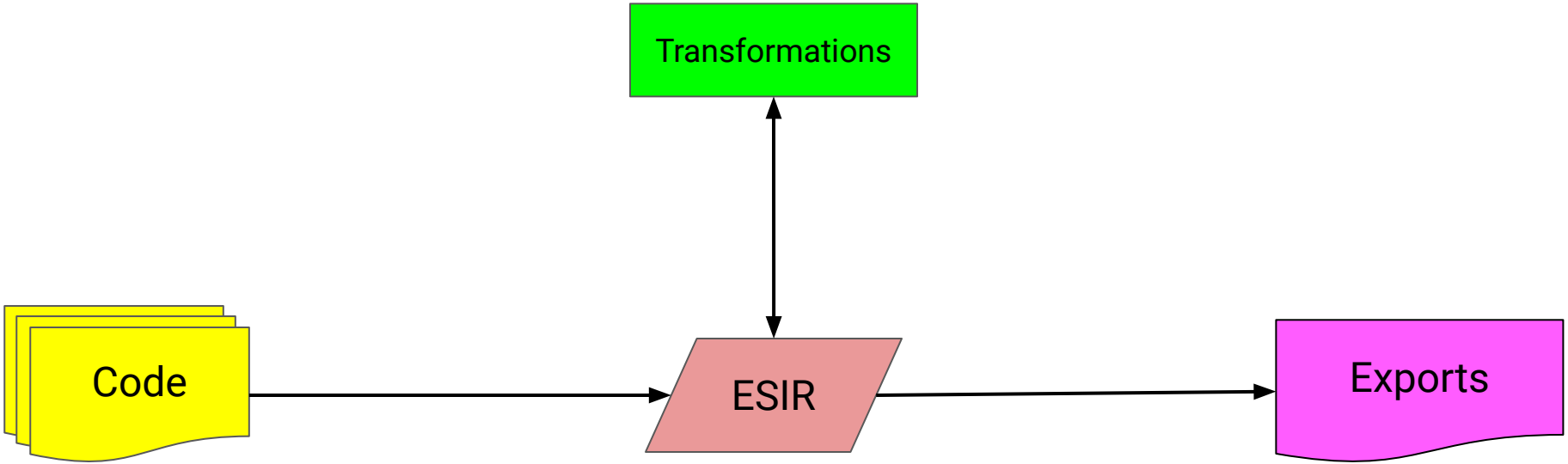
[Berkeley 2015]

Funding

DARPA

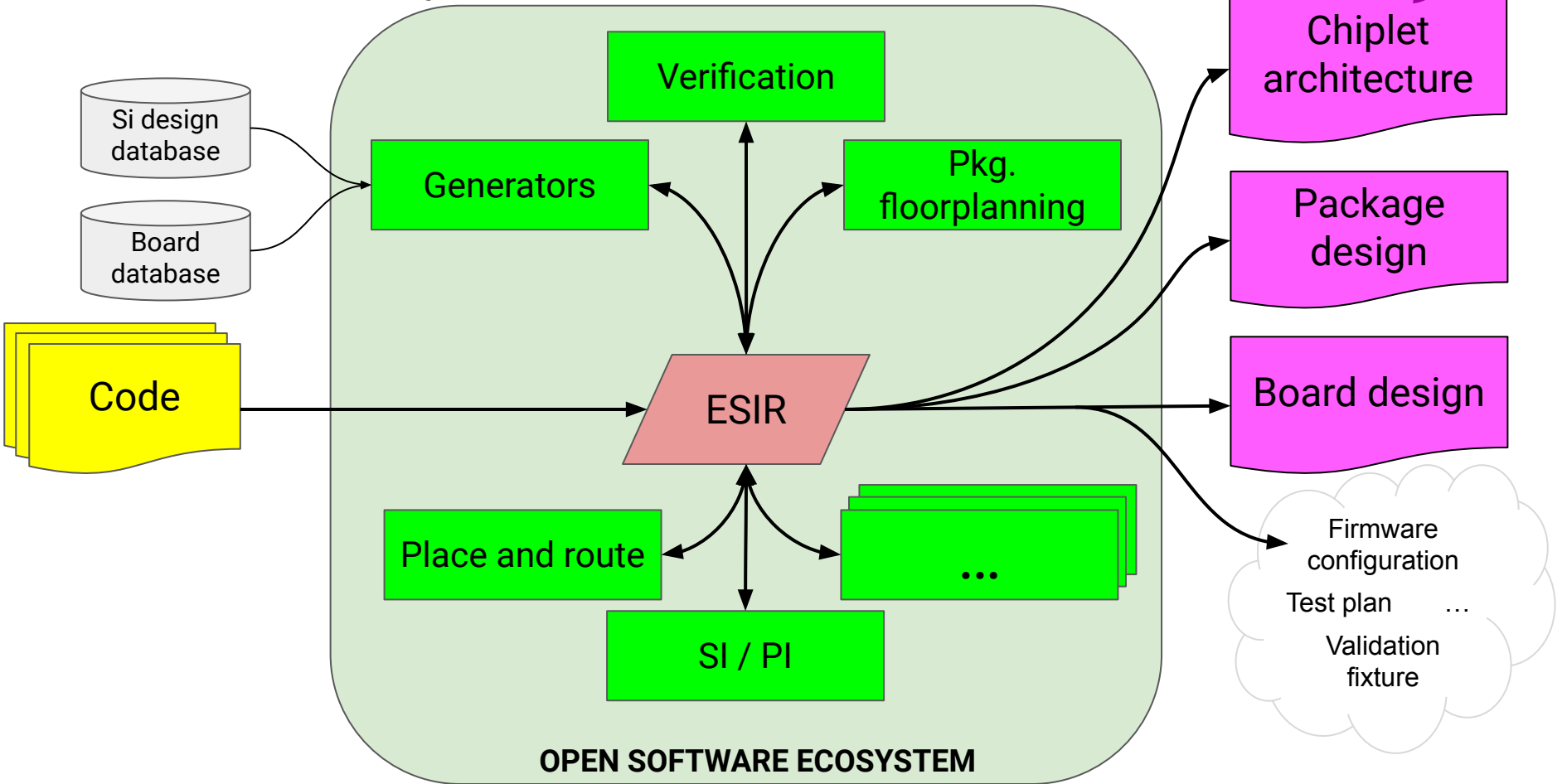
SEQUOIA

A compiler for systems of chiplets

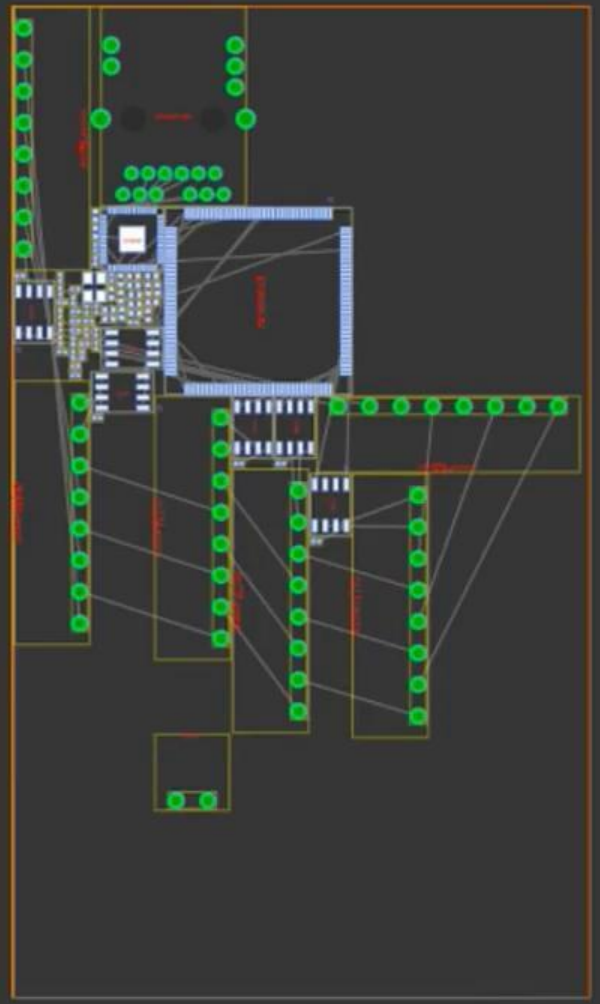


ESIR: Electronic Systems Intermediate Representation (open-source)

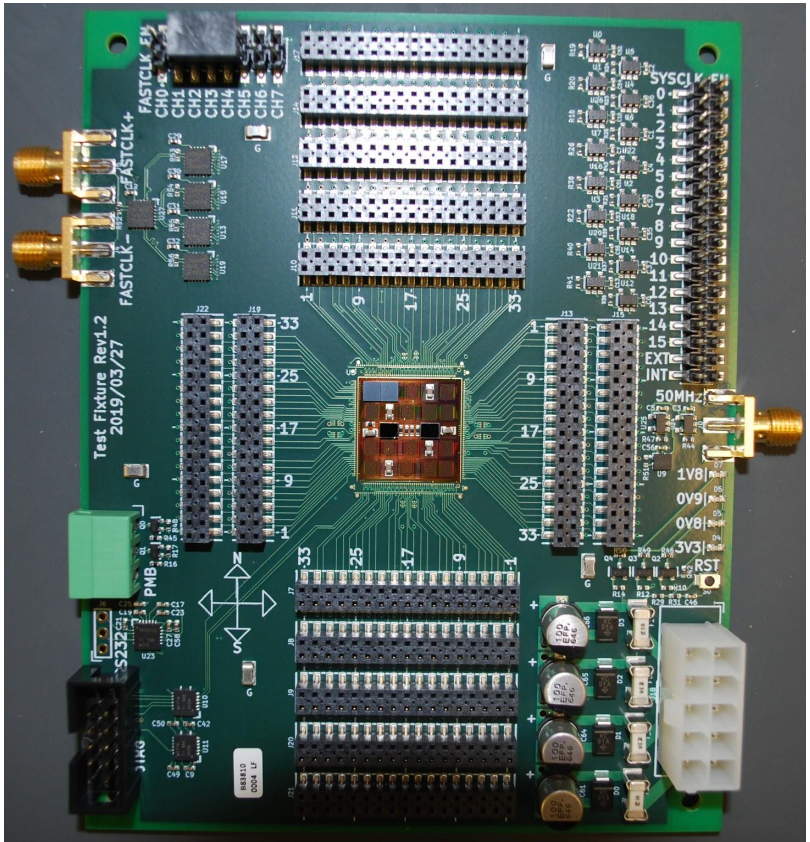
A compiler for systems of chiplets

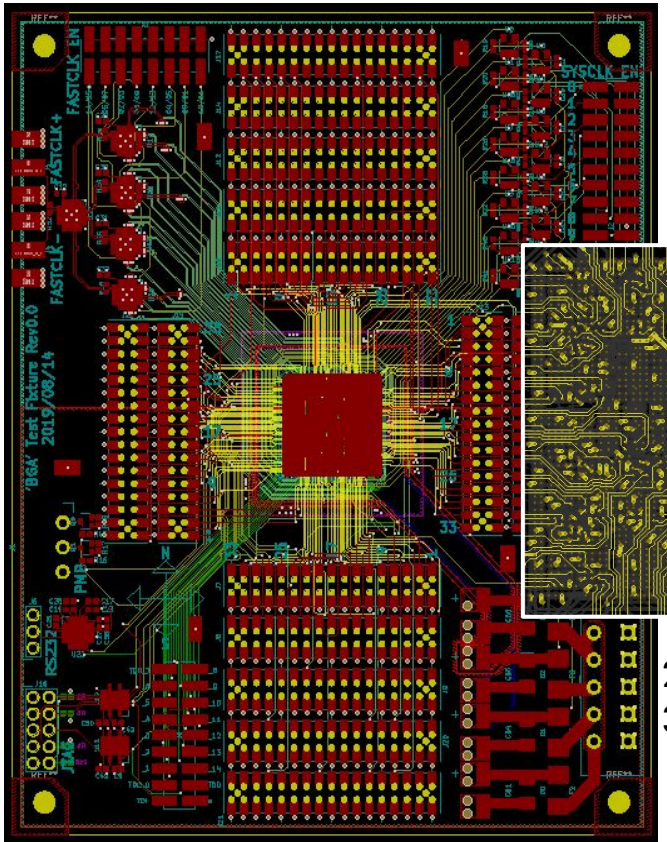


```
pcb_module daughter_card (daisy_chain:Int):
  port rs485:rs485
  port in : pin[daisy_chain]
  port out : pin[daisy_chain]
  inst conn : (ecdb/samtec/QTH/component(30))
  package(conn) at loc(0.0, 0.0) on top
  net (rs485.A, conn.p[1])
  net (rs485.B, conn.p[2])
  for i in 0 to daisy_chain do :
    net (conn.p[3 + 2 * i], in[i])
    net (conn.p[4 + 2 * i], out[i])

pcb_module mother_card :

  inst power_jack : (ecdb/phoenix_contact/mc/component(2))
  net gnd (power_jack.p[2])
  set_power_source(power_jack.p[1], power_jack.p[2], 9.0)

  inst proc : (ecdb/stmicroelectronics/STM32F427/module)

  inst ethernet_jack : (ecdb/pulse_electronics/J00-0009NL/component
  require rgmii:rgmii from proc
  require eth:ethernet_1000 from ethernet_jack
  connect phy (rgmii eth)
```



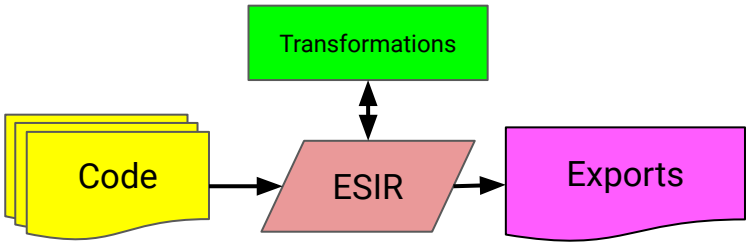
Two chiplet boards generated from software



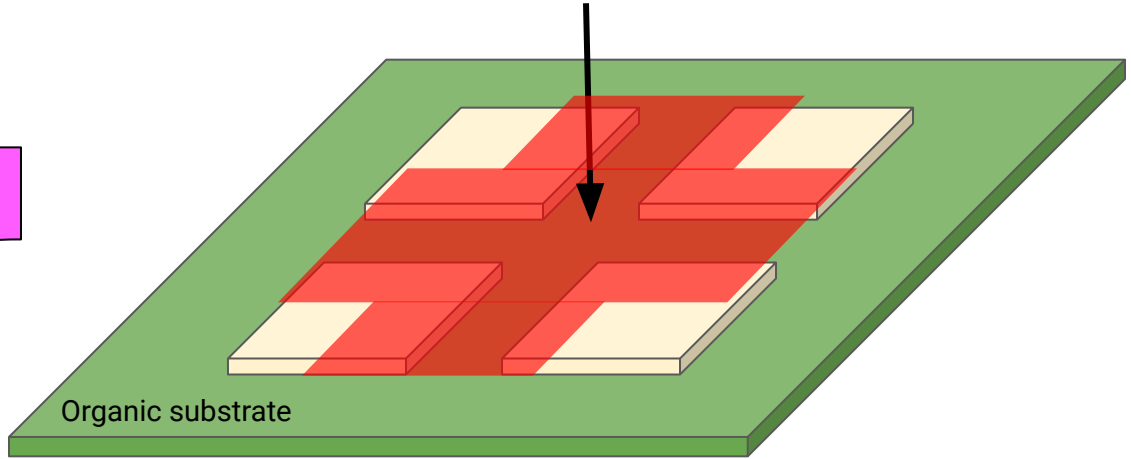


2500 pin
300um pitch

Chiplet compiler prototype



Target: Chiplet network using Bunch of Wires

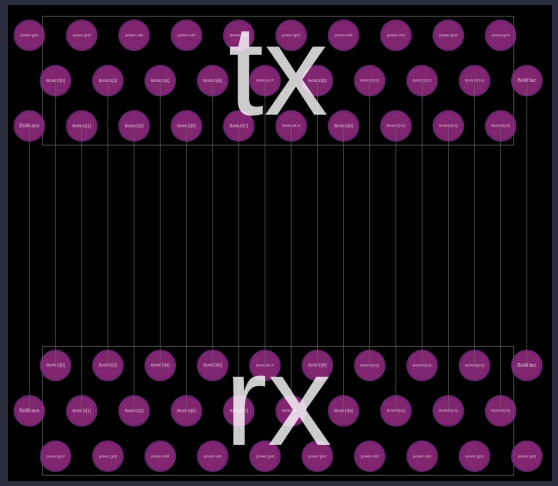


Bunch of wires - Slice

```
defn bunch-of-wires-slice-bumps (pitch:Double, rx?:True|False)

 inst tx-slice    : bunch-of-wires-slice(1000.0e-3, false)
 inst rx-slice    : bunch-of-wires-slice(1000.0e-3, true)
 inst tiny-slice  : bunch-of-wires-slice(40.0e-3, false)
```

Bunch of wires - Bundle and component

```
bundle bunch-of-wires :
 port D : pin[16]
 pin fec
 pin aux
 port clk : diff-pair

bunch-of-wires-slice (pitch:Double, rx?:True|False) :
 port BoW : bunch-of-wires


 inst tx : bunch-of-wires-slice(130.0e-3, false)
 inst rx : bunch-of-wires-slice(130.0e-3, true)
 net (tx.BoW rx.BoW)
```



Bunch of wires - Stack

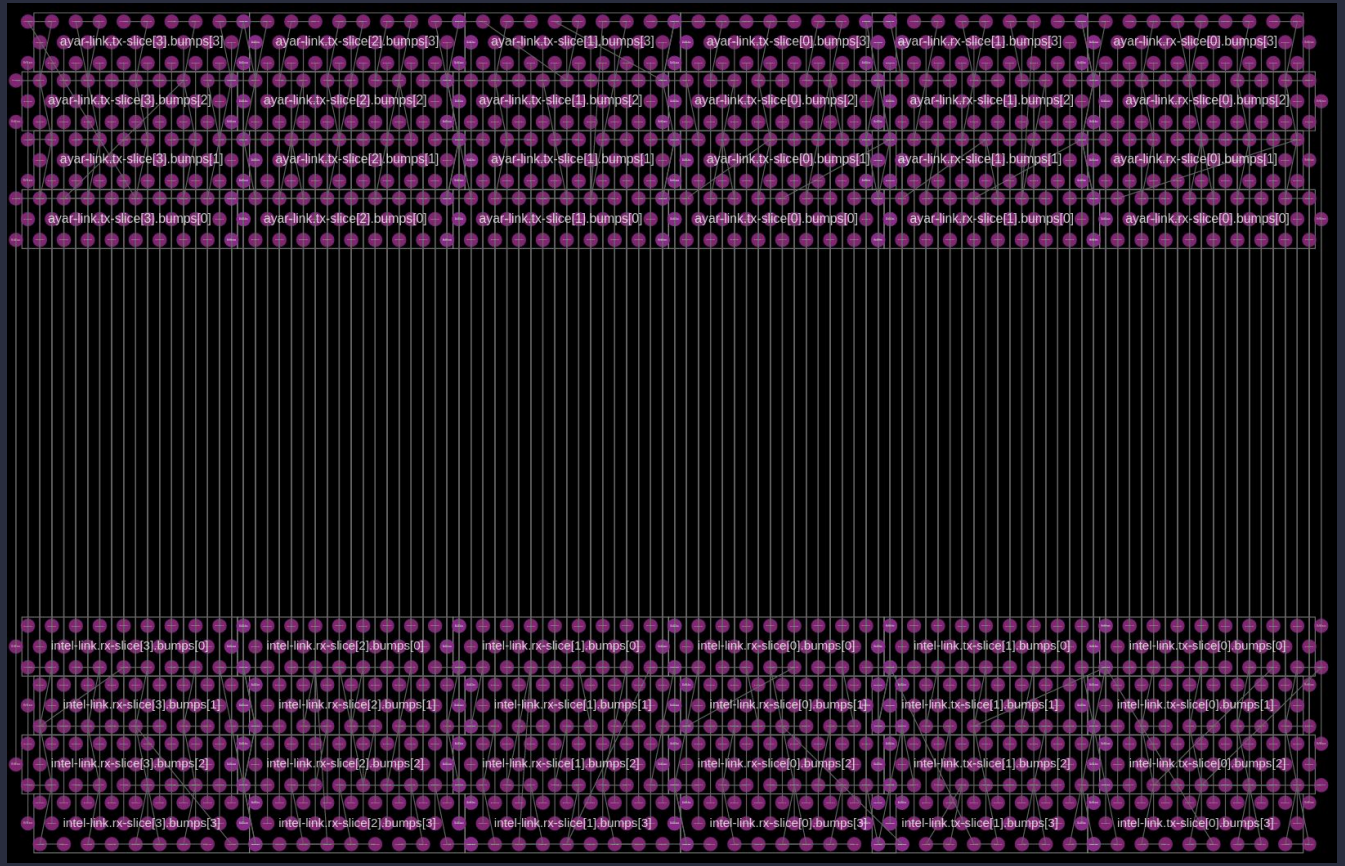
```
bunch-of-wires-stack (depth:Int, pitch:Double, rx?:True|False):
```

```
 inst rx-stack : bunch-of-wires-stack(4, 130.0e-3, true)
```

Bunch of wires - Link

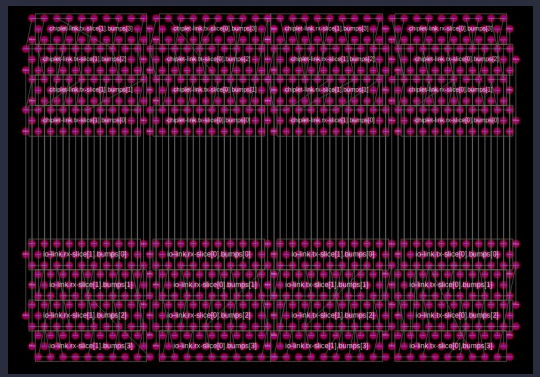
```
defn connect-bunch-of-wires (x:JITXObject, y:JITXObject):
 inside pcb-module :
   net (x.rx, y.tx)
   net (y.rx, x.tx)

bunch-of-wires-link-bumps (tx-size:Int, rx-size:Int, depth:Int, pitch:Double, flip?:True|False)

 inst ayar-link  : bunch-of-wires-link-bumps(16, 8,  4, 130.0e-3, true)
 inst intel-link : bunch-of-wires-link-bumps(8,  16, 4, 130.0e-3, false)

 place(ayar-link)  at loc(0.0, 0.0) on Top
 place(intel-link) at loc(0.0, -2.0, 180.0) on Top

 connect-bunch-of-wires(ayar-link.io intel-link.io)
```

Bunch of wires - Link



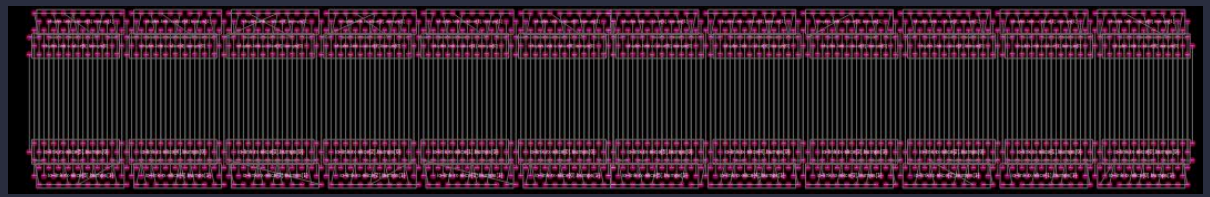
```
substrate-pitch = 60.0e-3
link = BoW-link( :
   bit-width = 100
   tx-clk = 1.0e6
   max-edge-width = 3.0
   source-terminated? = true)

optimize-link-parameters(link)
```

Minimize routing depth and unused pins. Report power.



> Power per BoW link : 0.0001152W

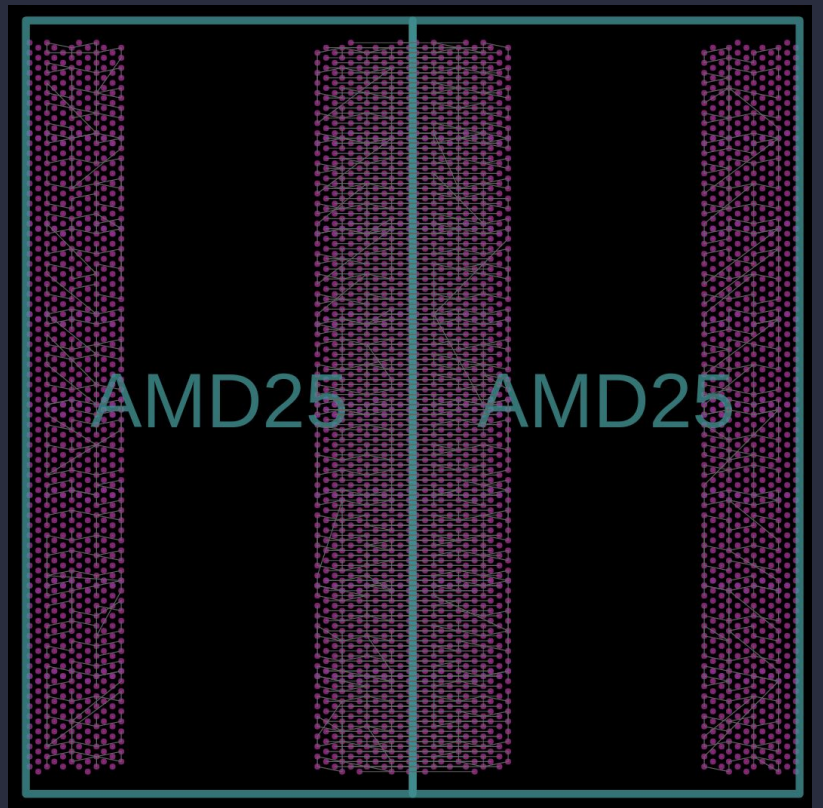
```
substrate-pitch = 60.0e-3
link = BoW-link( :
   bit-width = 150
   tx-clk = 2.0e6
   max-edge-width = 7.5
   source-terminated? = true)

optimize-link-parameters(link)
```



> Power per BoW link : 0.0003456W

Bunch of wires - Chiplet

```
inst amd-25 :  chiplet("AMD25", false, false, [16 16], [16 16], 4, 130.0e-3, Rectangle(5.0, 10.0))[2]

place(amd-25[0]) at loc(2.5, 10.0) on Top
place(amd-25[1]) at loc(-2.5, 10.0) on Top

connect-bunch-of-wires(amd-25[0].west-io amd-25[1].east-io)
```

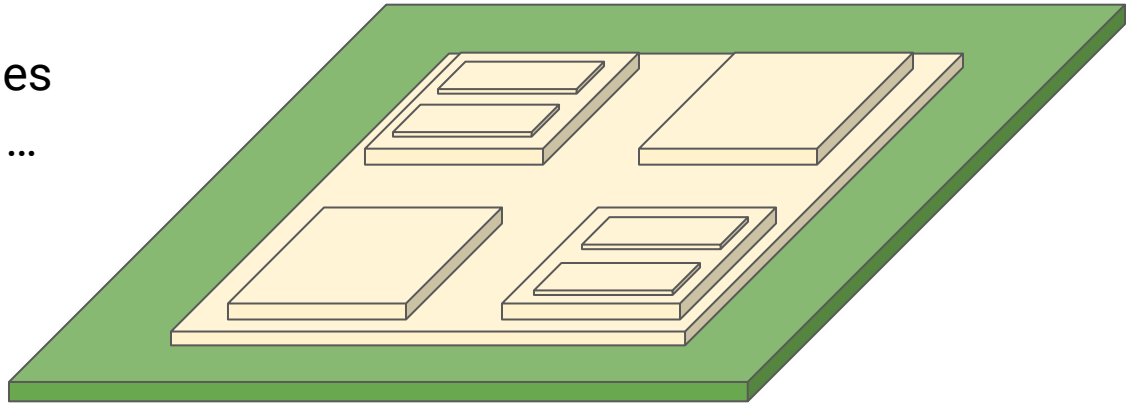


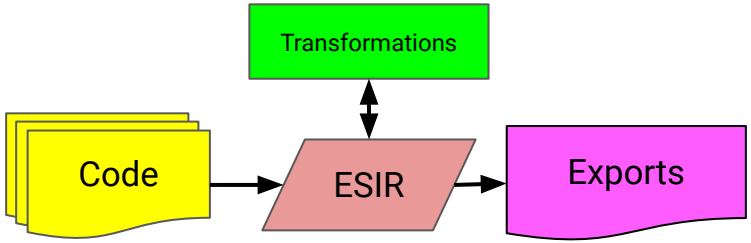
Next steps

Support more varieties of packages
- Stacking, interposers, EMIB, …

Add more export targets
- Bump map integration
- Glue logic for verification

Hook into upstream architecture optimization





Summary

We need design tools to support optimization for disaggregation

A software-defined approach integrates chiplet, package, and board design

We can now interactively generate and evaluate networks of chiplets

Contact information: Duncan Haldane - d.haldane@jitx.com

Questions?