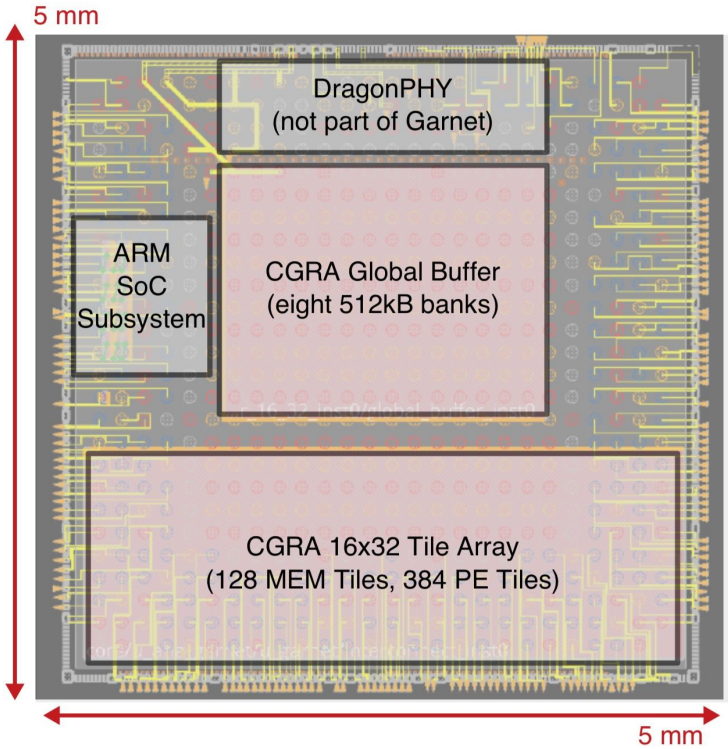
AHA Hardware Team

What Amber looks like + Agile teamwork + Next steps

Christopher Torng

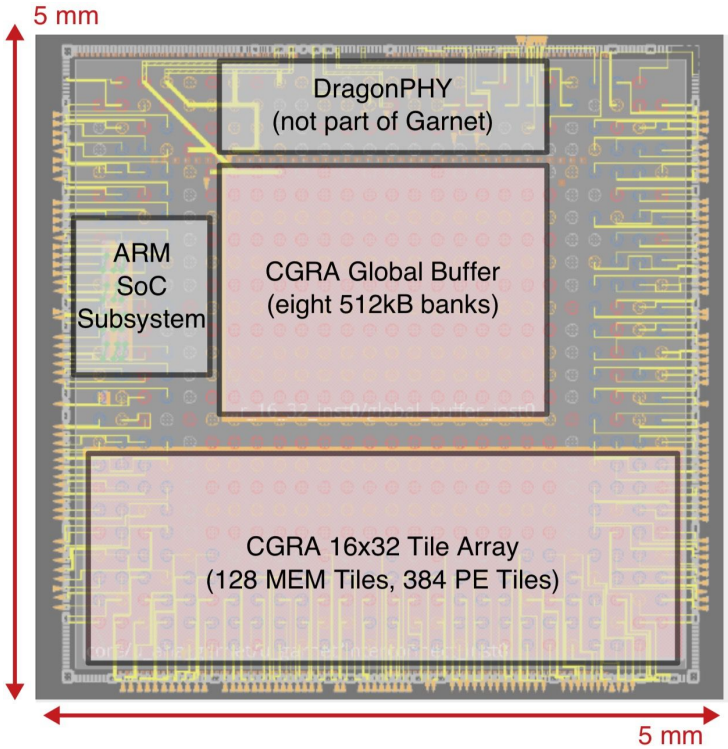
AHA
Agile Hardware Center

What Garnet SoC was



- 5mm x 5mm, TSMC16, flip chip
- Blocks
  - ARM SoC Subsystem
  - Global Buffer (4MB)
  - 32x16 PE/MEM tile array
- CGRA+GB (750MHz) and ARM SoC (500MHz)
- Single source of truth with software toolchain
  - PEak (PE), Lake (MEM), Canal (INTER)

Amber SoC is similar (1/3)

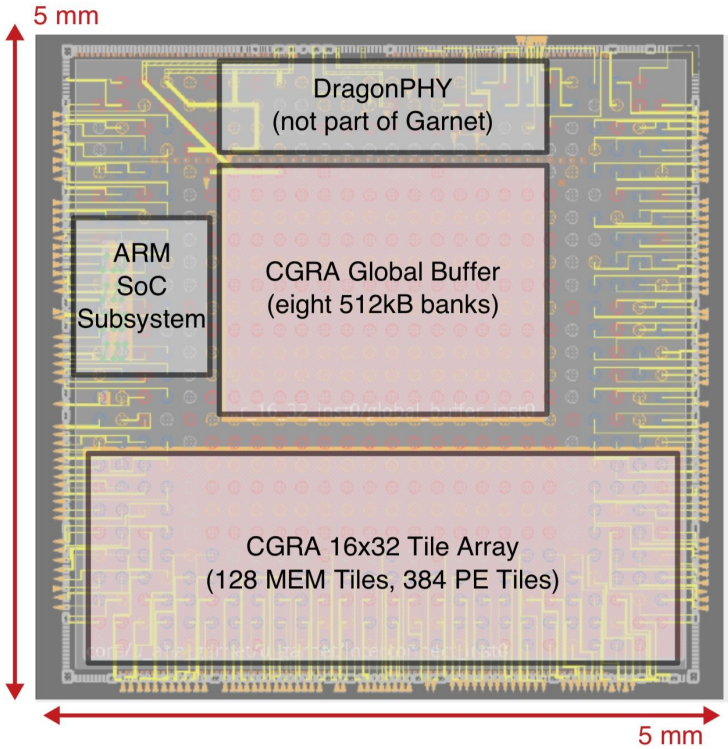


The **architecture** is mostly the same.

Among the larger design differences are:

- PEs with ponds (~30% area overhead)
- Lake after its diet (~30% area reduction)
- Interconnect / config tweaks (~5-10% area red.)
- Other optimizations for area, power, and timing

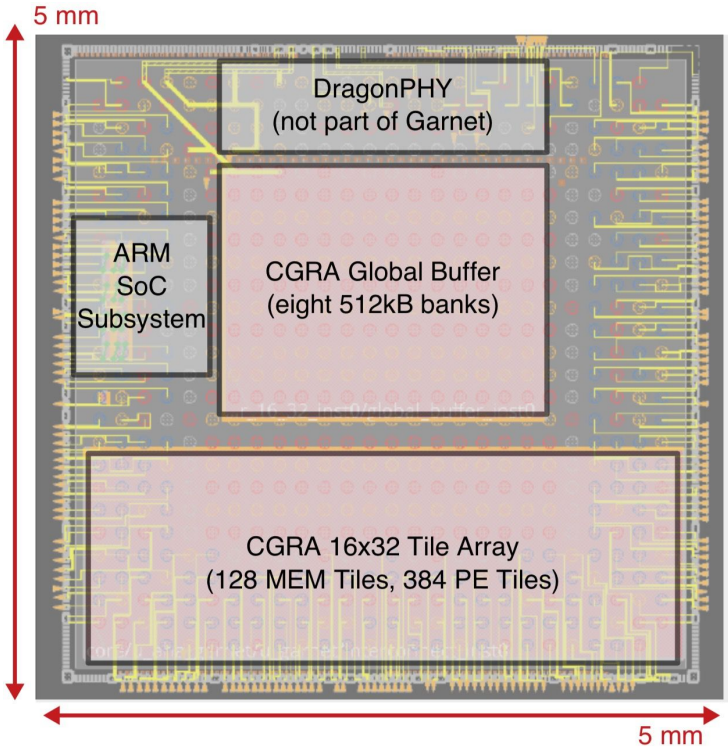
Amber SoC is similar (2/3)



The key **new experiments** we wanted to enable:

1. Heterogeneous PE arrays -- for a future demo
2. Virtualization-related changes
   - Tweaks for GLB to stream configuration bitstreams to individual columns
   - Tweaks for first row of CGRA to redirect I/O

Amber SoC is similar (3/3)



New support in infrastructure for test/DSE:

- E.g., Generated random CoreIR graphs for random directed testing of the PEs

- E.g., Templated generated random CoreIR graphs for random directed testing of the MEMs

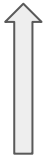
- E.g., Ongoing work at SoC level with apps

This infrastructure supports both Amber SoC as well as the various papers going to ISCA/DAC.

The TSMC shuttle date has a screenshot

From our TSMC representative:

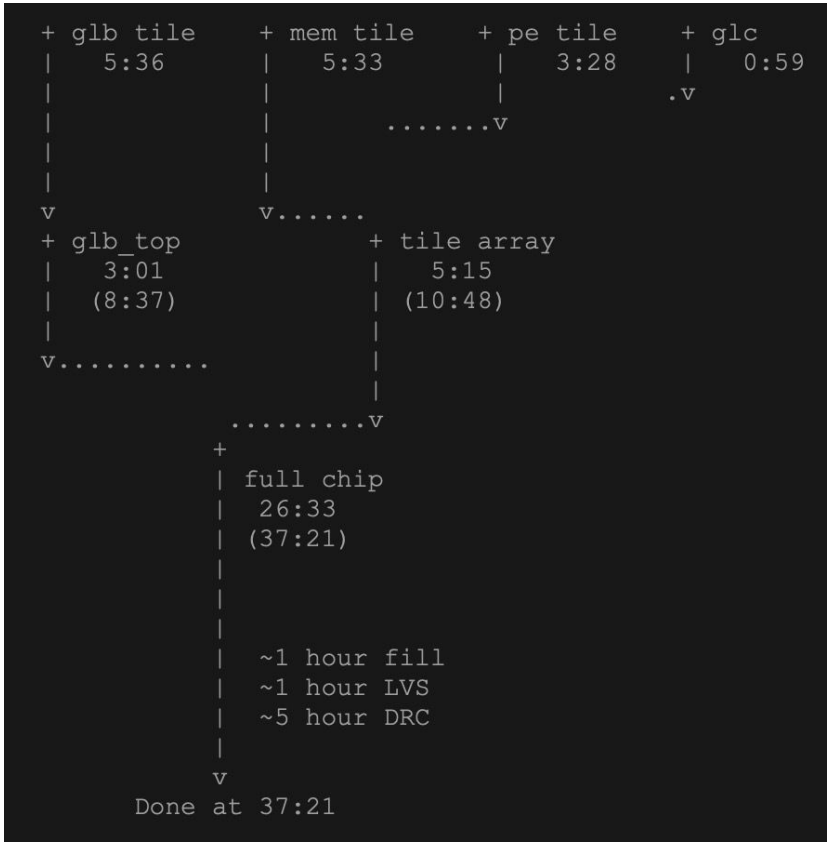
Customer Product Name	Status	Step	Tape-in Date	Shuttle Type	Resv Date
Aha-Garnet (Stanford Horowitz)	Approved	2	12/01/2020	Pre-general offer 16 nm CMOS LOGIC FinFET Compact (Shrink) LL ELK Cu 1P13M 0.8/1.8V	07/16/2020



How the hardware team has moved towards agile processes

We have weekly full chip releases

A complete full chip spin takes about 36 hours to finish (on the arm machine)



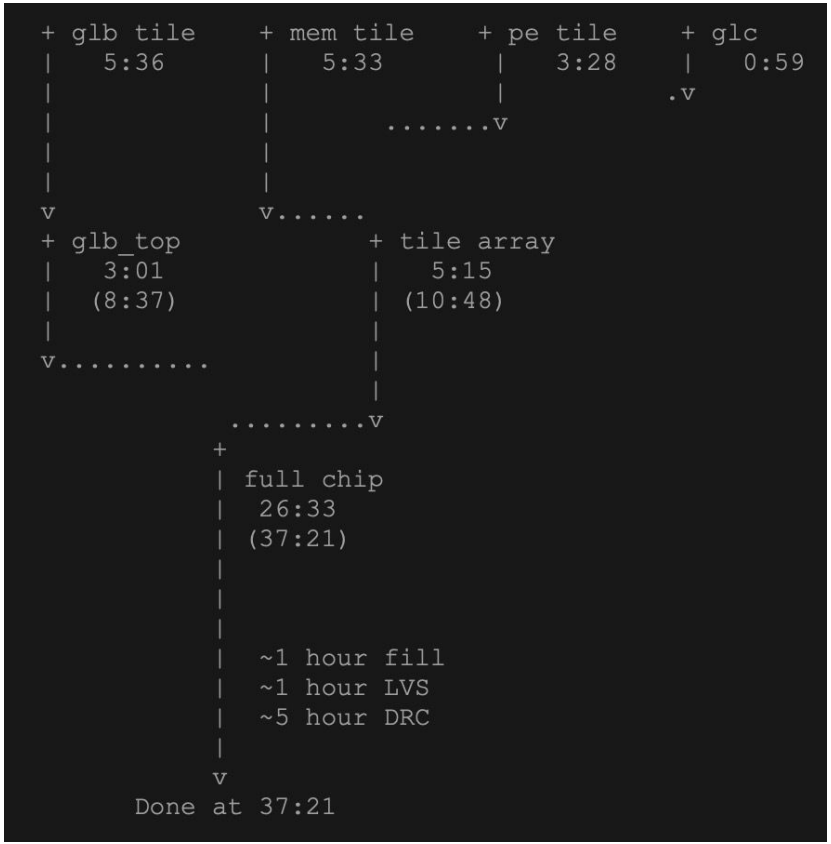
We have weekly full chip releases

A complete full chip spin takes about 36 hours to finish (on the arm machine)

Problems:

- Could not kick this off on each commit
- Somewhat fragile

The natural outcome is to let it stagnate



We have weekly full chip releases

A complete full chip spin takes about 36 hours to finish (on the arm machine)

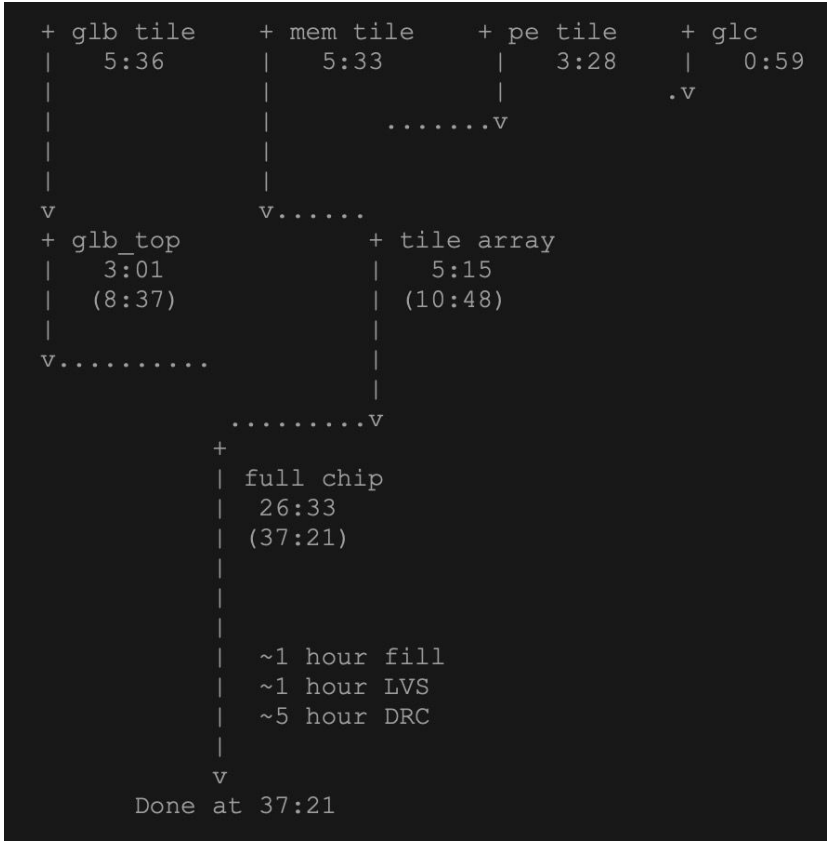
Problems:

- Could not kick this off on each commit
- Somewhat fragile

The natural outcome is to let it stagnate

Solution:

- A weekly release build
- Concept of a release slot every week



We have weekly full chip releases

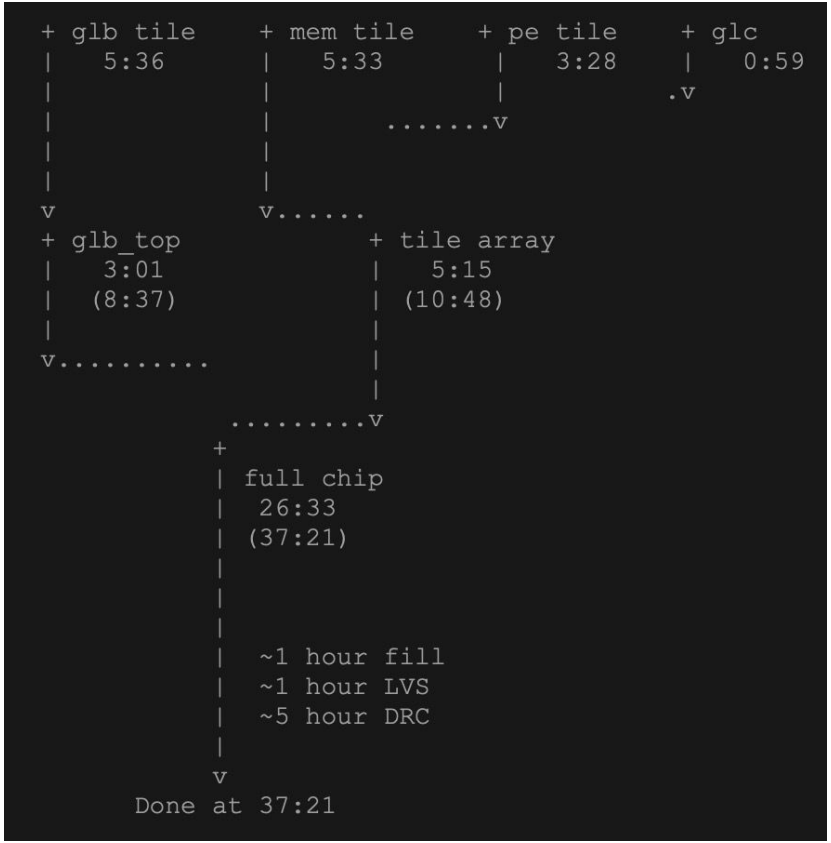
Some success **reflection notes** from working with this for ~1.5 months:

- The first few releases were very fragile (lots of babying)

- Most recent releases are more stable

- The team now has a *language* about whether a feature will "make it into the Friday release" (and which Friday)

- We have quite a few golden GDS's



We have weekly full chip releases

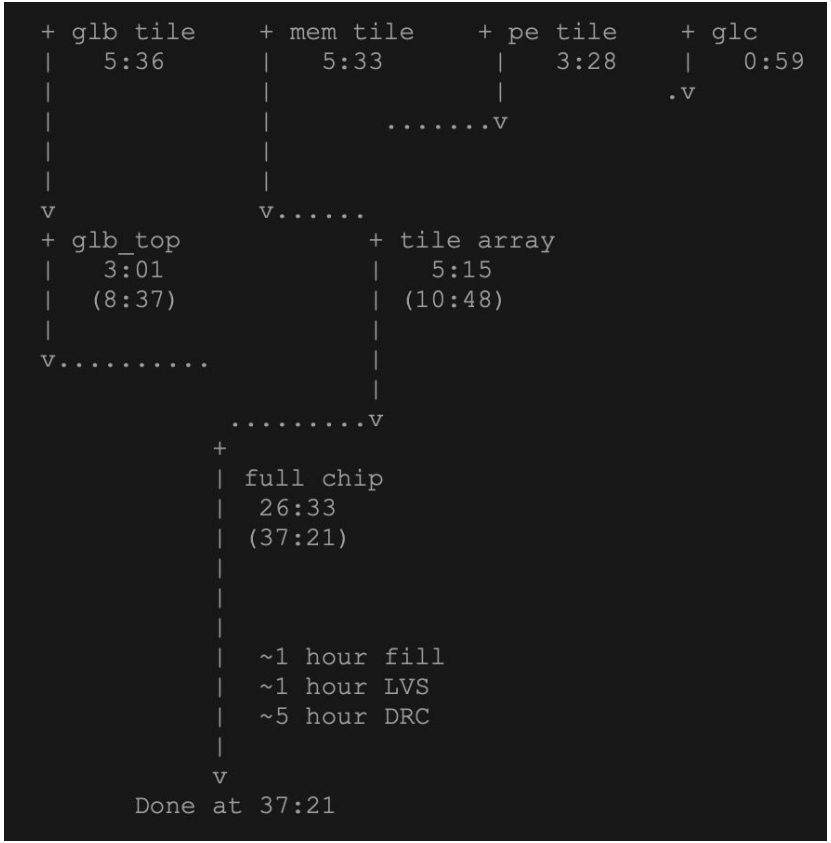
Some **caveats**:

- We have this set up locally (not on the Cadence cloud)

- Still requires some babying each week. Is the babying worth the benefits?

Future potential in our bucket list:

- CI to provide intermediate entry points through mflowgen stash pre-built steps



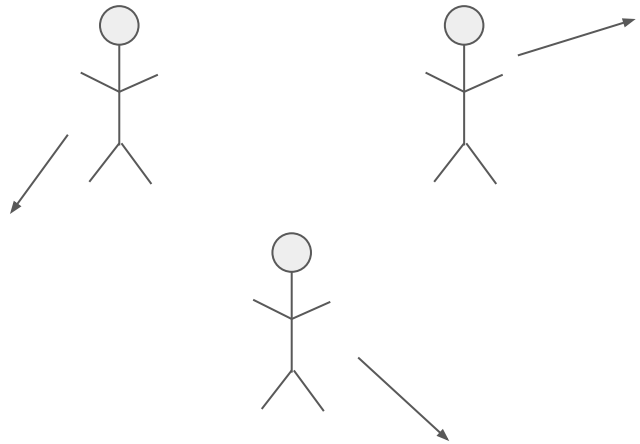
We are seriously trying out biweekly agile sprints

In the past, we did not truly act as an agile team.

We are seriously trying out biweekly agile sprints

In the past, we did not truly act as an agile team.

- N students push in N different directions
- A repeated question of "what is our goal?"



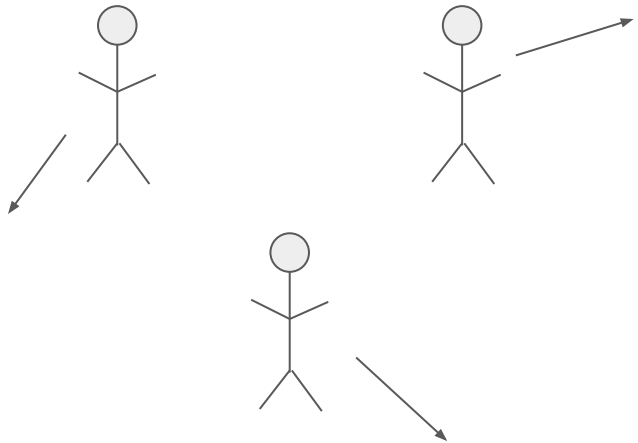
We are seriously trying out biweekly agile sprints

In the past, we did not truly act as an agile team.

- N students push in N different directions
- A repeated question of "what is our goal?"

This was **good for students**:

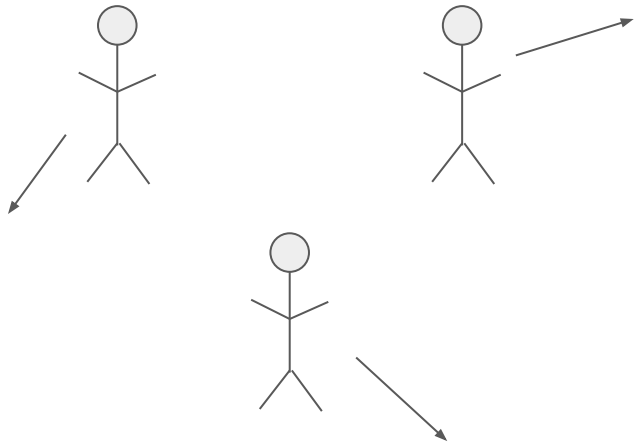
- Students own a piece as experts, get papers



We are seriously trying out biweekly agile sprints

In the past, we did not truly act as an agile team.

- N students push in N different directions
- A repeated question of "what is our goal?"



This was **good for students**:

- Students own a piece as experts, get papers

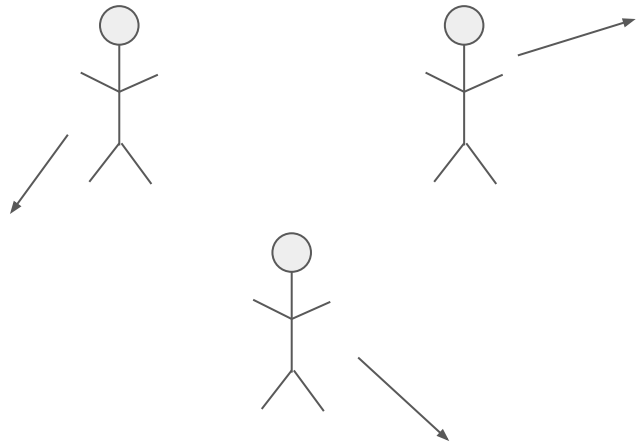
This was **also bad for students**:

- Students in silos
- Direction unclear
- Bad boundaries/interfaces between tools

We are seriously trying out biweekly agile sprints

In the past, we did not truly act as an agile team.

- N students push in N different directions
- A repeated question of "what is our goal?"



This was **good for students**:

- Students own a piece as experts, get papers

This was **also bad for students**:

- Students in silos
- Direction unclear
- Bad boundaries/interfaces between tools

Fewer integration-level efforts that demonstrate value in the whole project

We are seriously trying out biweekly agile sprints

Demonstrate incremental yet meaningful value to the whole project.

The key procedure as an agile team:

1. Lay out a **meaningful sprint goal** with most of the two-week effort framed out

2. Pass this to a sprint team of students who buy in to what they find meaningful

3. At the end of the time box, we can demonstrate that new product

Agile sprints should almost always succeed, otherwise the framing of the sprint was off target.

We are seriously trying out biweekly agile sprints

We are using tools commonly used in agile sprint teams

Moving Forward

Sprint goals that span across compiler/hardware

An agile view of the upcoming quarter

There are 10 weeks in a quarter, which makes room for ~5 focused team efforts to deliver something meaningful but incremental in **a set amount of time**.

Four biweekly sprints before Amber SoC:

- September 21st to October 4th

- October 5th to October 18th

- October 19th to November 1st

- November 2nd to November 15th

An agile view of the upcoming quarter

There are 10 weeks in a quarter, which makes room for ~5 focused team efforts to deliver something meaningful but incremental in **a set amount of time**.

Four biweekly sprints before Amber SoC:

- September 21st to October 4th
- October 5th to October 18th
- October 19th to November 1st
- November 2nd to November 15th

Move entirely to the cloud

Complete integration of ponds

A development focus on test/DSE infra for Amber/ISCA

Randomly generated CoreIR testing