Accelerated Genome Sequencing for Precision Medicine

Jack Wadden

Sequal Inc.

Jack Wadden

- UVA PhD 2018
- Research interests:
  - Parallel architecture, application specific accelerators, reconfigurable computing, hardware description languages, bio-informatics
- Post-doc at UM
  - Working with Satish Narayanasamy*, Reetu Das*, and David Blaauw*
  - Researching:
    - Low-latency genome sequencing for intra-operative cancer diagnosis
    - Low-latency, and low-cost metagenomic testing for infection diagnosis
- Senior Architect at ***Sequal Inc.***
  - Cloud-based whole genome sequencing software as a service
  - Startup founded by * as a spin-off from exciting academic research

Sequal Team

Professor, UM, IEEE Fellow
Co-founder, Ambiq (Series D)
Co-founder, CubeWorks (Series A)
Co-founder, Sequal
Advisor to Mythic (Series A)
Expertise: VLSI Design

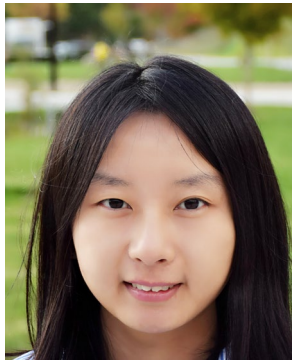
David Blaauw
CSO, Sequal

Asst. Professor, UM
Sloan Fellow
ISCA and MICRO Hall of fame
Expertise: Computer architecture

Reetu das
CTO, Sequal

Assoc. Professor, UM
NSF CAREER
ISCA and ASPLOS HoF
Expertise: Parallel systems

Satish Narayanasamy
CEO, Sequal



Asst. Professor, UM,
Pulmonary and Critical Care Medicine

Expertise: Lung disease, sequencing, microbiome

Xiao Wu

UM MS'18
5+ years experience in hardware design

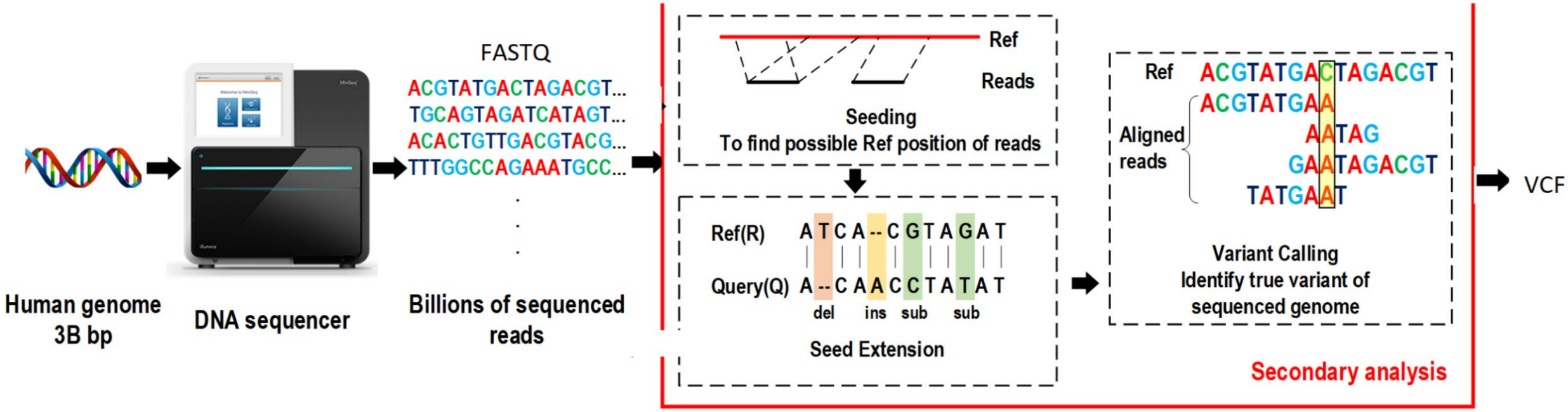
Kush Goliya

U. Virginia, PhD'18
10+ years experience in system design

Jack Wadden

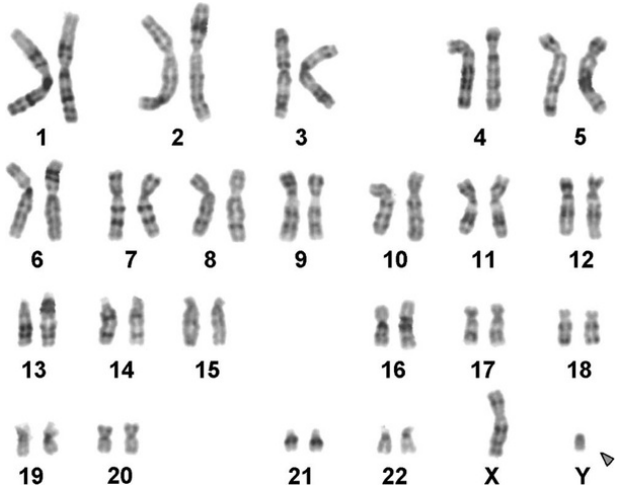
We do Whole Genome Sequencing (WGS)

WGS involves "reading" your DNA code



We do Whole Genome Sequencing (WGS)

WGS involves "reading" your DNA code

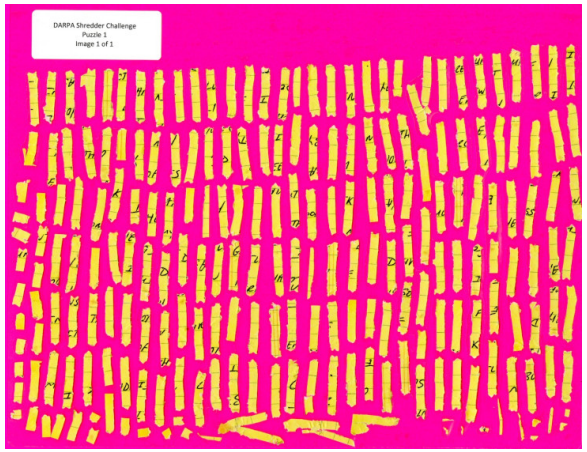


3.2 Billion Base Pairs (ATGC)

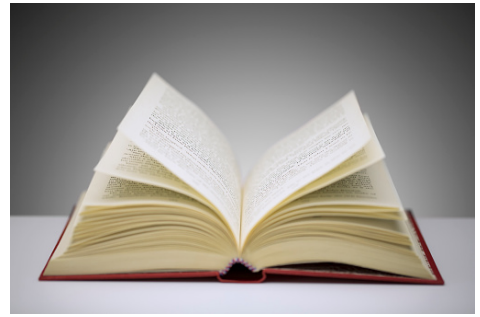
What "typos" do you have?

We do Whole Genome Sequencing (WGS)

WGS involves "reading" your DNA code



3.2 Billion Base Pairs (ATGC)



What was your book?

What "typos" do you have?

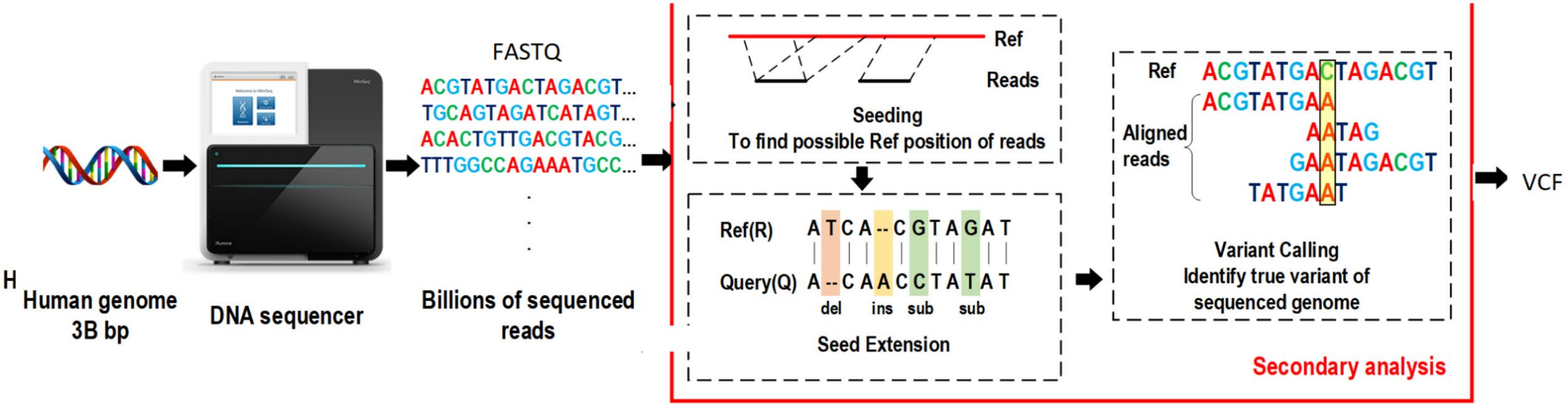
What DNA snippets were in your cells?

Primary Analysis

Secondary Analysis

We do Whole Genome Sequencing (WGS)

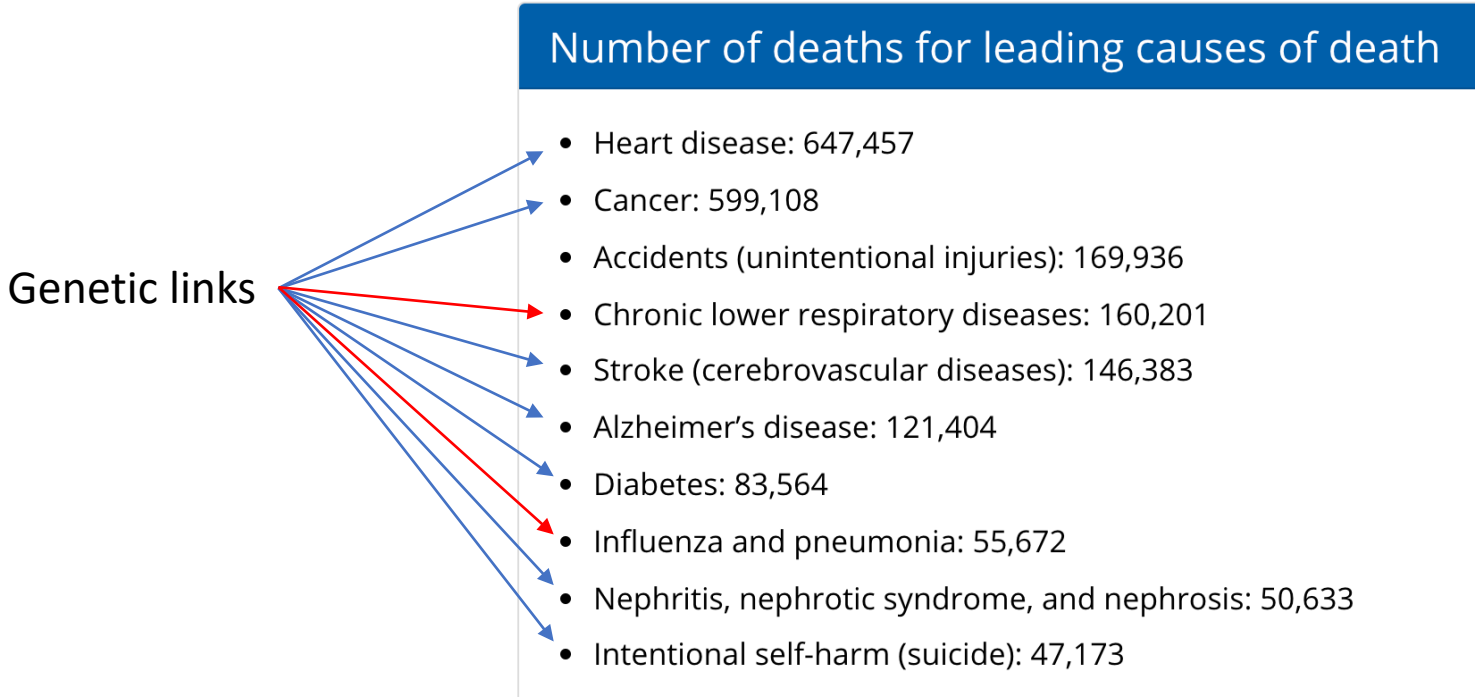
WGS involves "reading" your DNA code



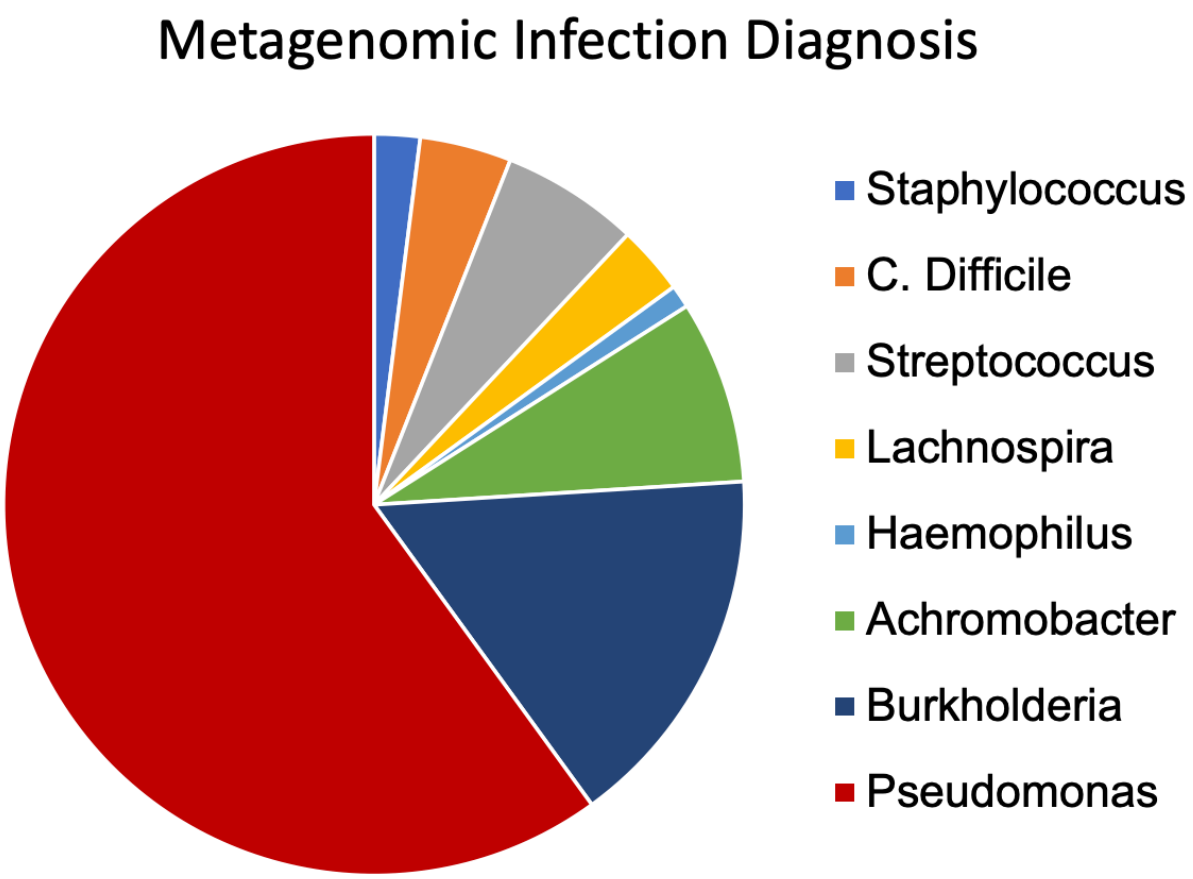
Why is this useful?

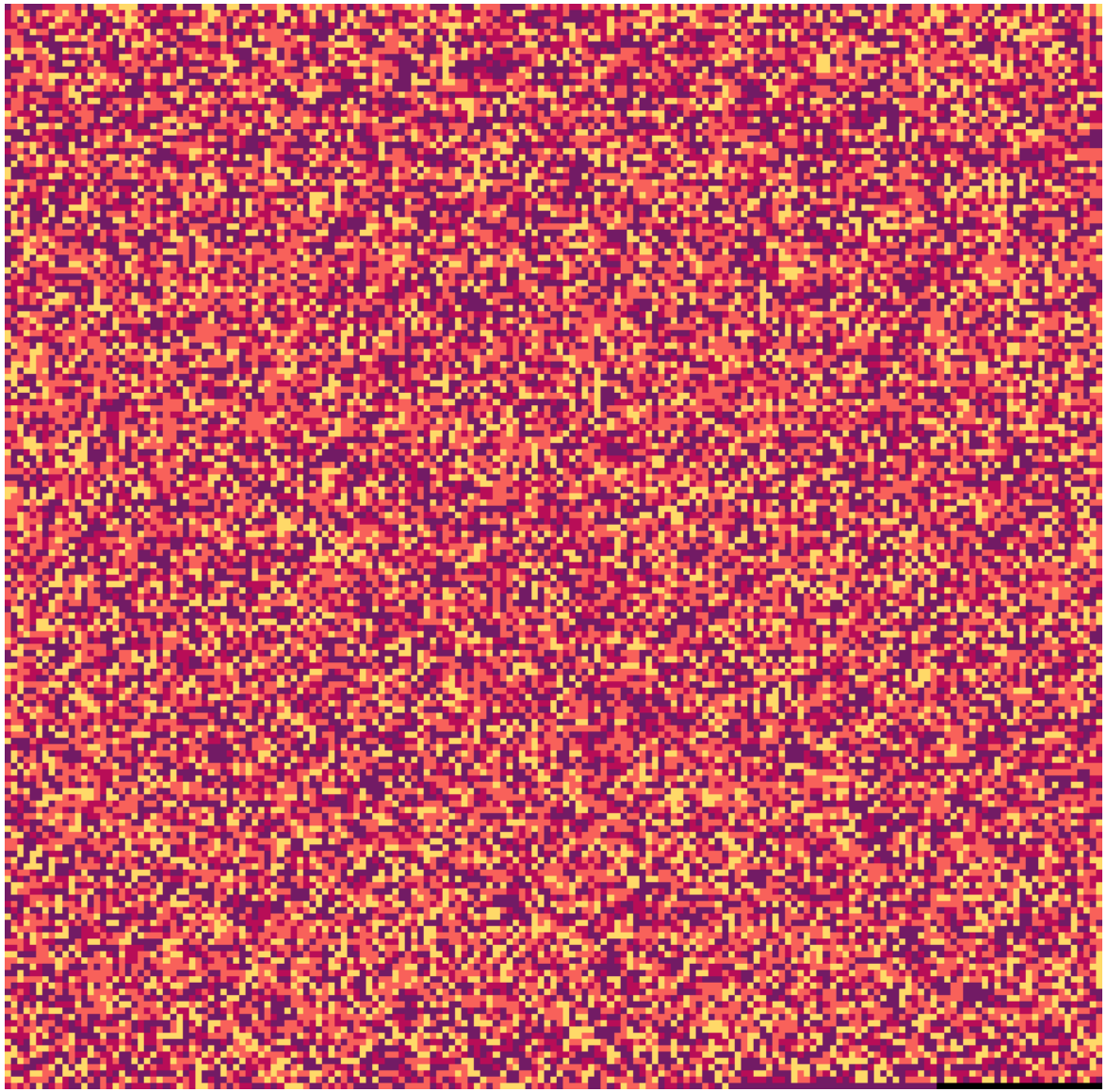
Many diseases are caused by undesirable genetic mutations

- **Cancer**
- Huntingtons
- Cycstic fibrosis
- Alpha-beta-thalassemias
- Sickle cell anemia
- Marfan syndrome
- Fragile X syndrome
- Hemochromatosis
- ….. Literally thousands

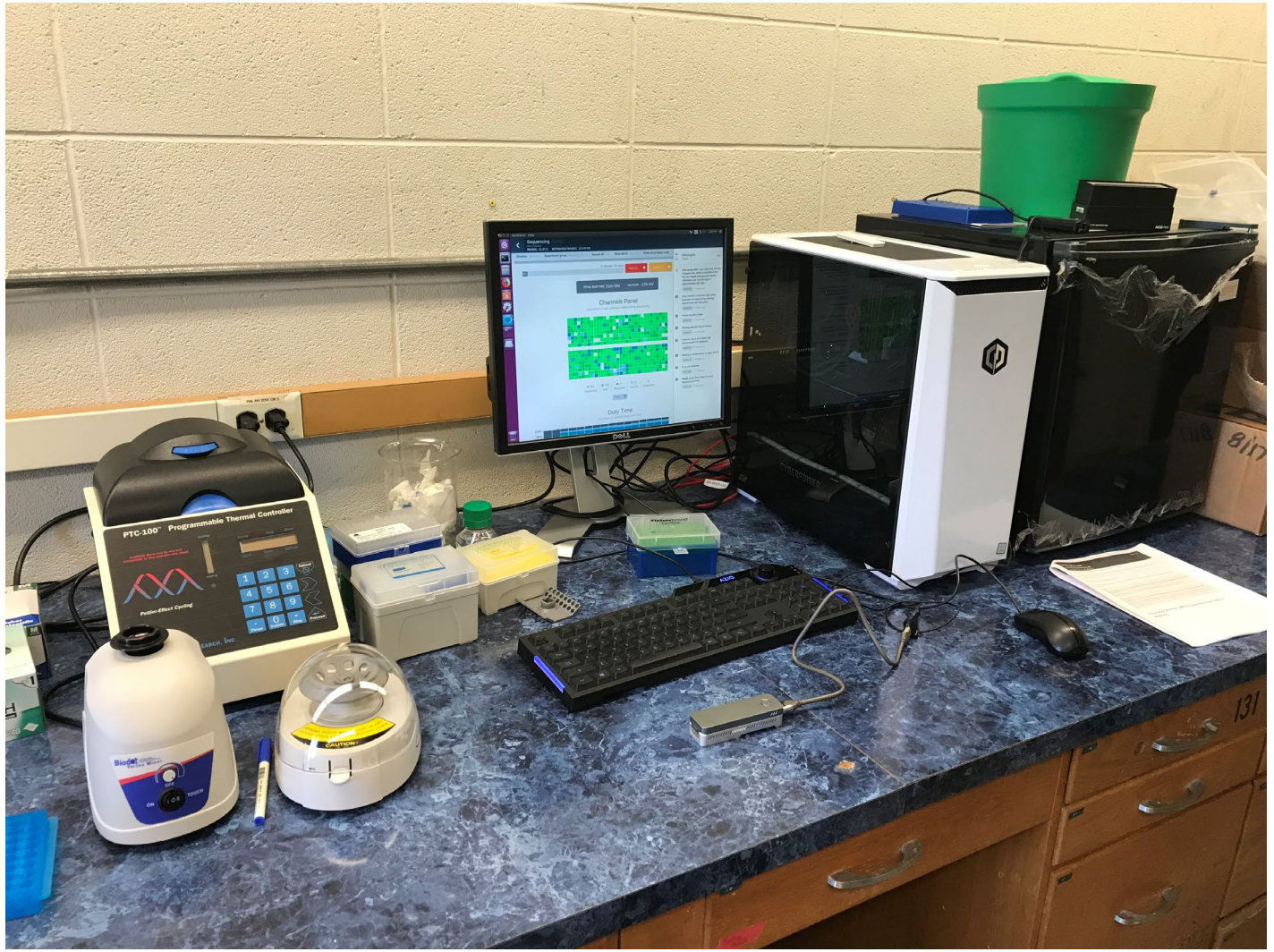


Infections can be diagnosed using DNA/RNA sequencing





DNA sequencers are becoming cheaper, and portable



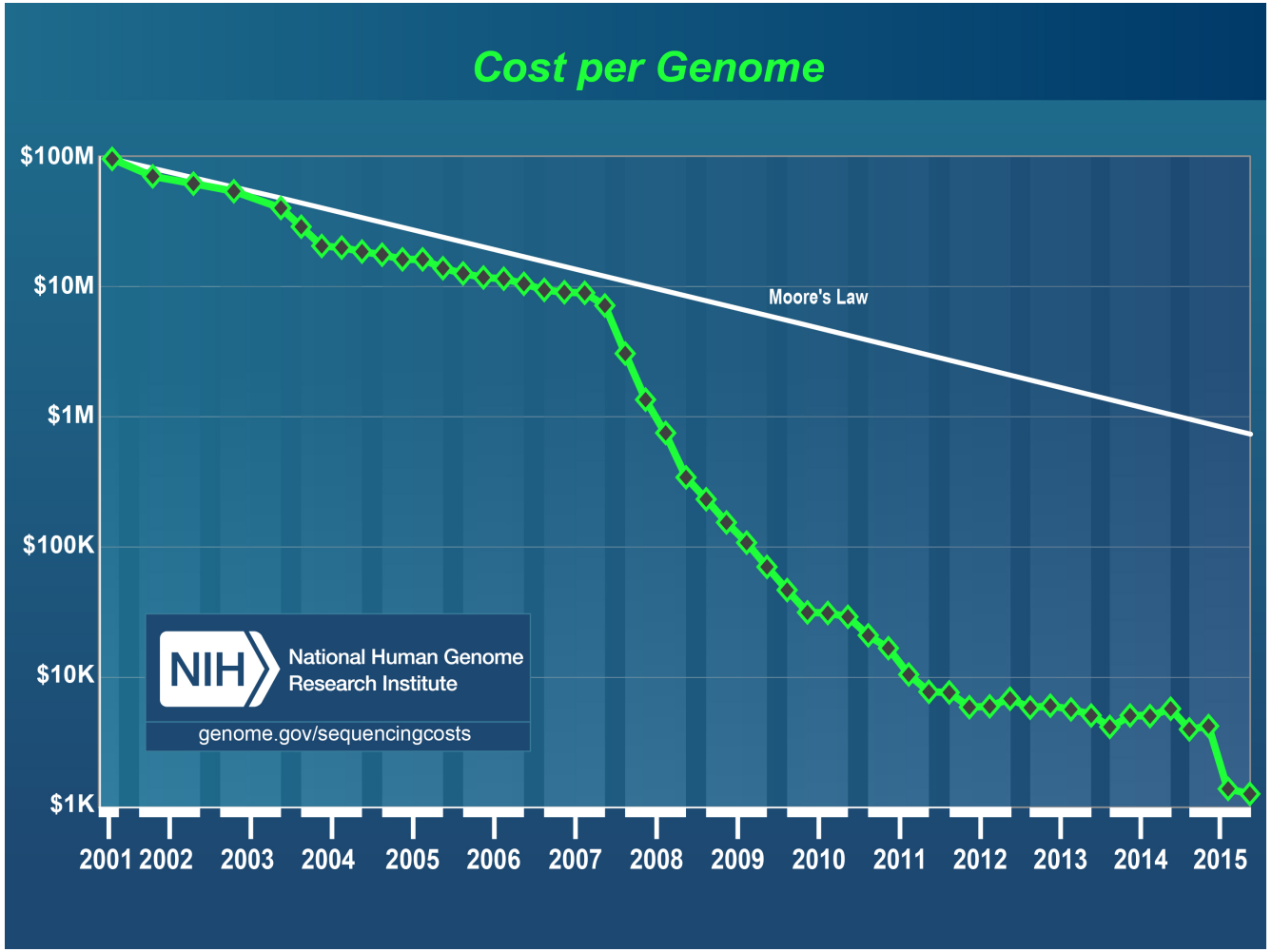
Genome Sequencing Costs are Plummeting

Cost per whole human genome

$100 M → $100*

(2001) (2020)

- Illumina's projection



The Human Genome is Getting More Complete

New Results Posted August 16, 2019.

Comment on this paper

Telomere-to-telomere assembly of a complete human X chromosome

Karen H. Miga, Sergey Koren, Arang Rhie, Mitchell R. Vollger, Ariel Gershman, Andrey Bzikadze, Shelise Brooks, Edmund Howe, David Porubsky, Glennis A. Logsdon, Valerie A. Schneider, Tamara Potapova, Jonathan Wood, William Chow, Joel Armstrong, Jeanne Fredrickson, Evgenia Pak, Kristof Tigyi, Milinn Kremitzki, Christopher Markovic, Valerie Maduro, Amalia Dutra, Gerard G. Bouffard, Alexander M. Chang, Nancy F. Hansen, Françoisen Thibaud-Nissen, Anthony D. Schmitt, Jon-Matthew Belton, Siddarth Selvaraj, Megan Y. Dennis, Daniela C. Soto, Ruta Sahasrabudhe, Gulhan Kaya, Josh Quick, Nicholas J. Loman, Nadine Holmes, Matthew Loose, Urvashi Surti, Rosa ana Risques, Tina A. Graves Lindsay, Robert Fulton, Ira Hall, Benedict Paten, Kerstin Howe, Winston Timp, Alice Young, James C. Mullikin, Pavel A. Pevzner, Jennifer L. Gerton, Beth A. Sullivan, Evan E. Eichler, Adam M. Phillippy

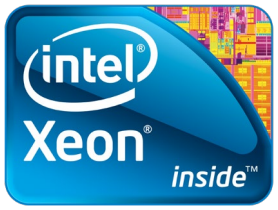
doi: https://doi.org/10.1101/735928

This article is a preprint and has not been certified by peer review [what does this mean?].

After nearly two decades of improvements, the current human reference genome (GRCh38) is the most accurate and complete vertebrate genome ever produced. However, no one chromosome has been finished end to end, and hundreds of unresolved gaps persist [1,2]. The

Secondary analysis (computer stuff) is starting to dominate cost

How much does secondary analysis cost?



Cost	Latency	Platform
~$41	~12 hrs.	Amazon AWS CPU cloud

~160x

$0.26	~20 min	Amazon AWS FPGA cloud (F1)

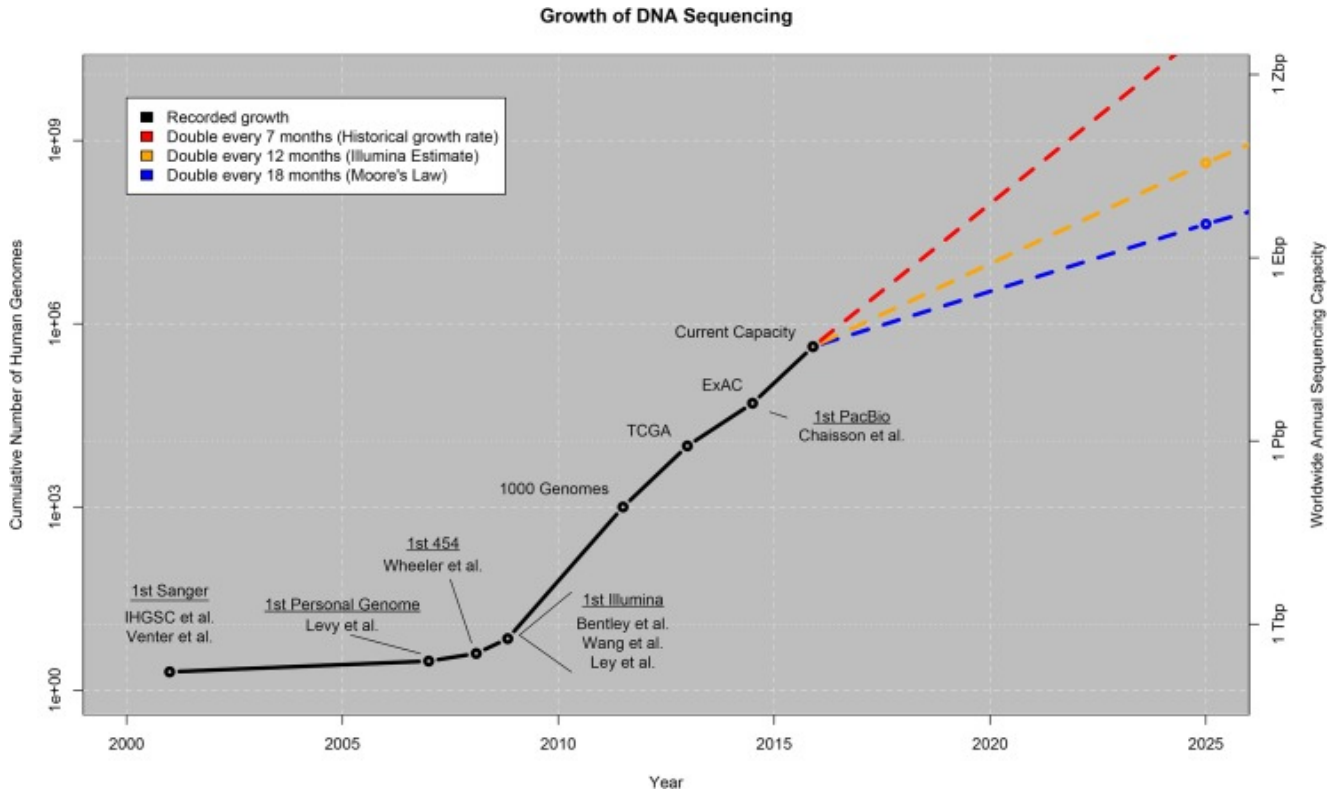
Sequal Accelerated WGS Pipeline

Guarantees Broad Institute's gold standard output
(bit-equivalent to software BWA-MEM + GATK)

What is the market for WGS?

Human WGS Market: < 1% Penetration today

~10x Growth Potential in 5-7 years



Today:
1 million human genomes sequenced

In 5 years:
12+ million human genomes*

~160 million per year market

(Only WGS. Does not include other uses of sequencing)

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4494865/

*Ack: Canaccord Genuity

What is our business model?

Customers

Hospitals

Research Centers

Governmental agencies

Grail, Dante

23andMe

Genetic and sequencing services

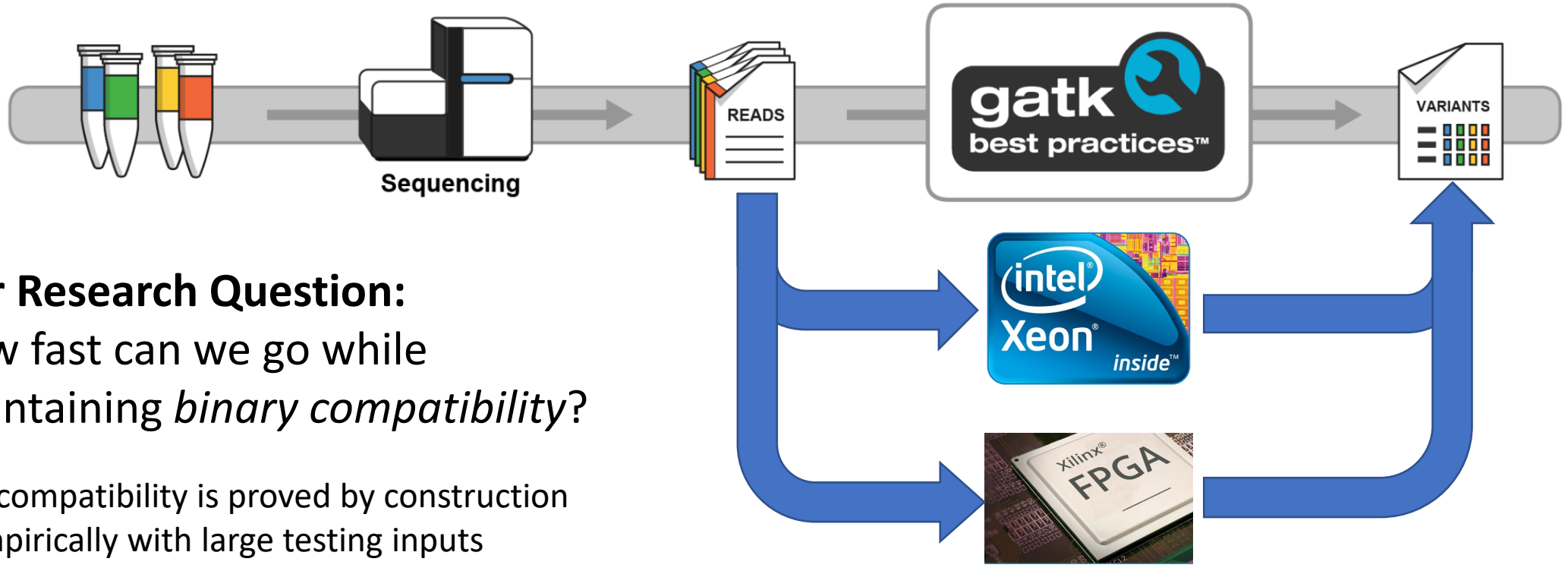
Sequal on Amazon or Sequal Compute Cloud

WGS pipeline
(BWA-MEM + GATK)

amazon web services

Cost isn’t everything….

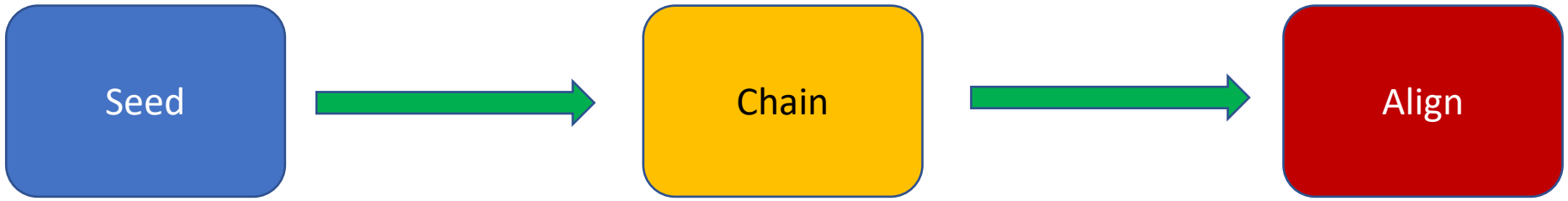
Clinical practice and research require ***stable***, ***validated*** pipelines



Our Research Question:
How fast can we go while maintaining *binary compatibility*?

Binary compatibility is proved by construction and empirically with large testing inputs

BWA-MEM Read Alignment Overview



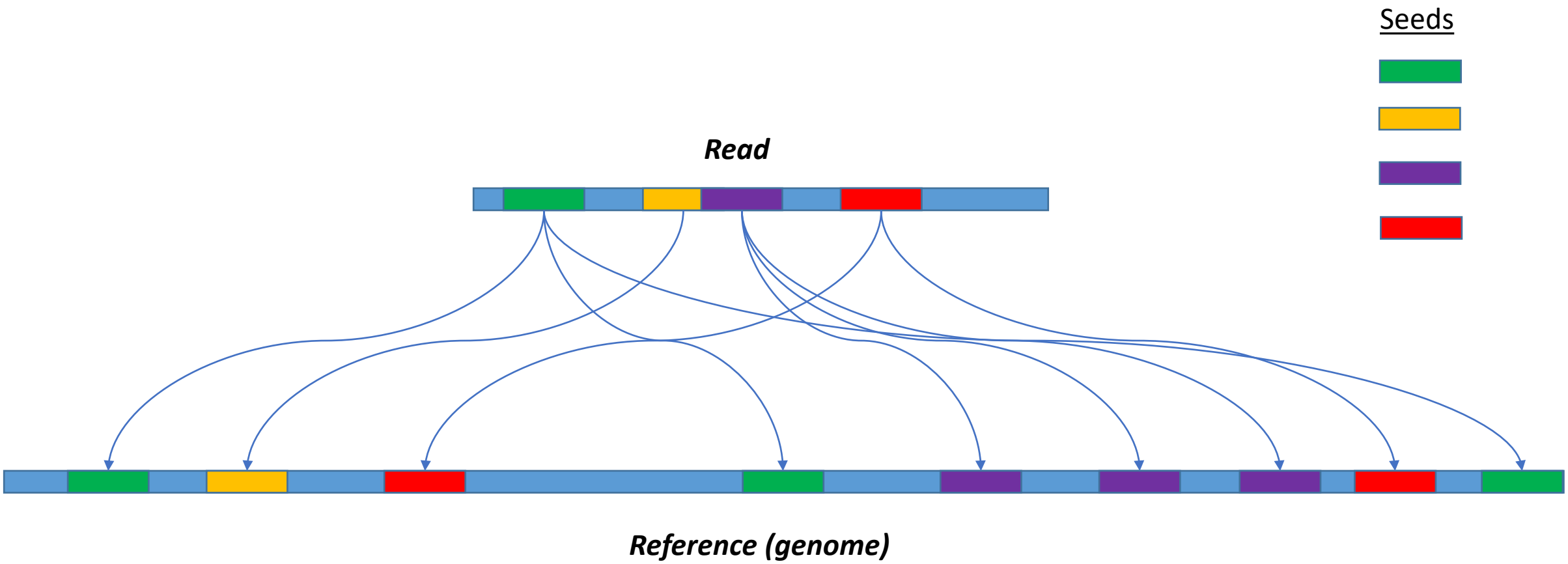
Take small snippets of the read and look-up where they might belong in the reference

Find the sequence of seed locations in the reference that correspond to the read's seeds

Use a string scoring algorithm to find the exact alignment of the read to the genome

Seeding high-level overview

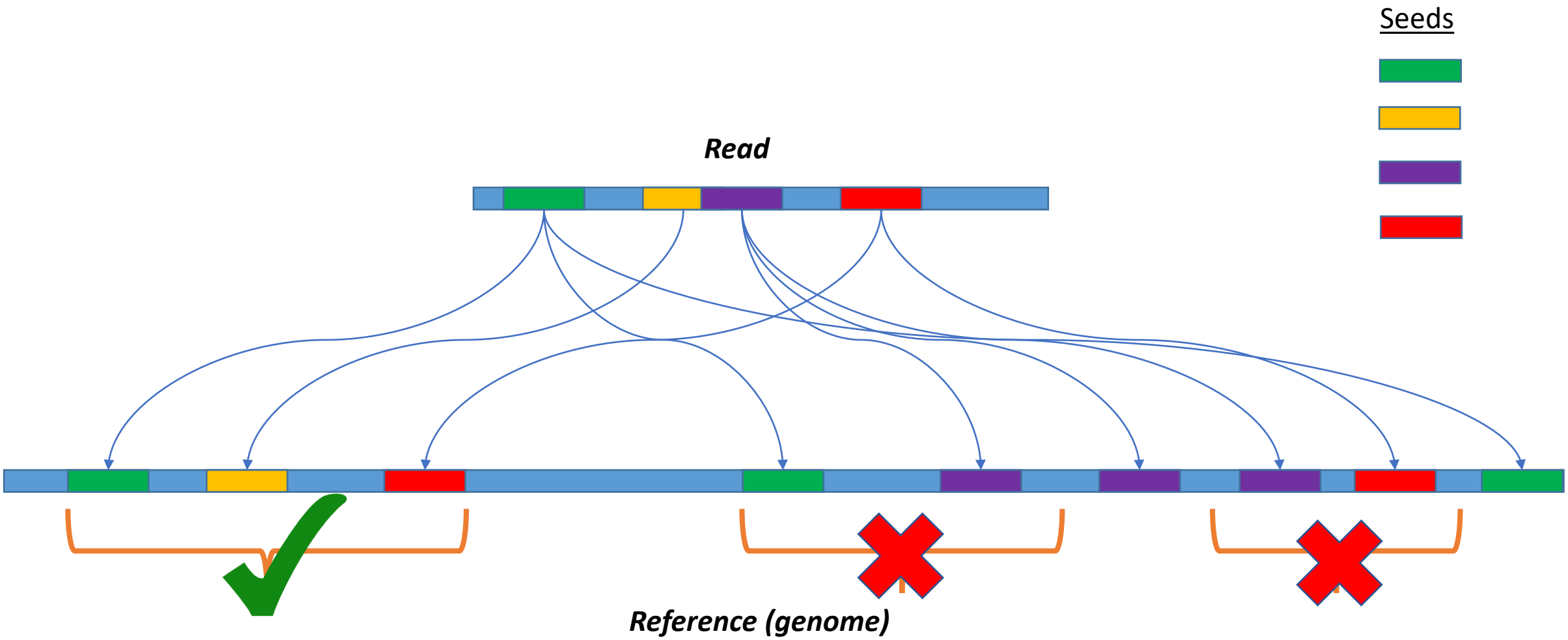
Seed



Where does the read belong?

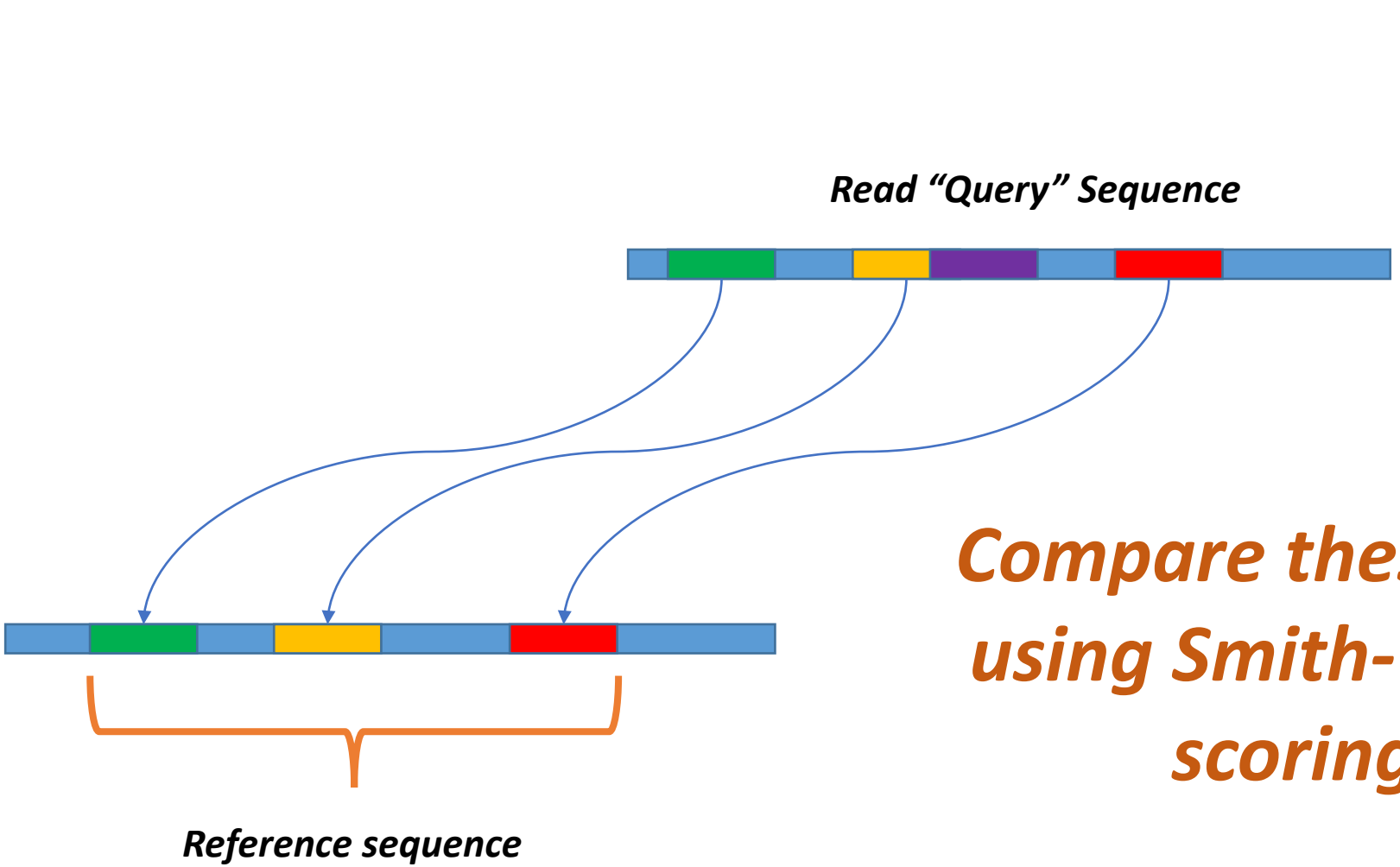
Chaining high-level overview

Chain



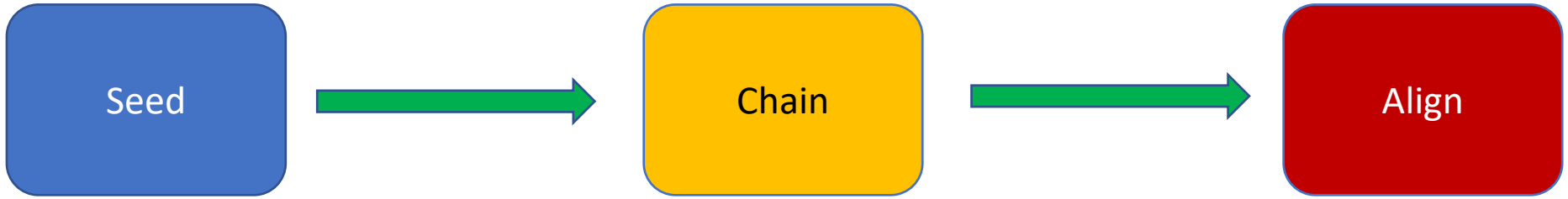
Alignment High-level Overview

Align



Compare these two sequences using Smith-Waterman string scoring algorithm

BWA-MEM Alignment Overview

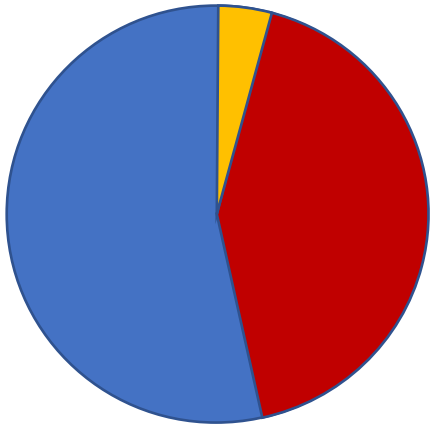


Take small snippets of the read and look them up in the reference

Find the sequence of seeds in the reference that's most like the read's sequence

Use string scoring algorithm to find where the read is not exactly like the reference

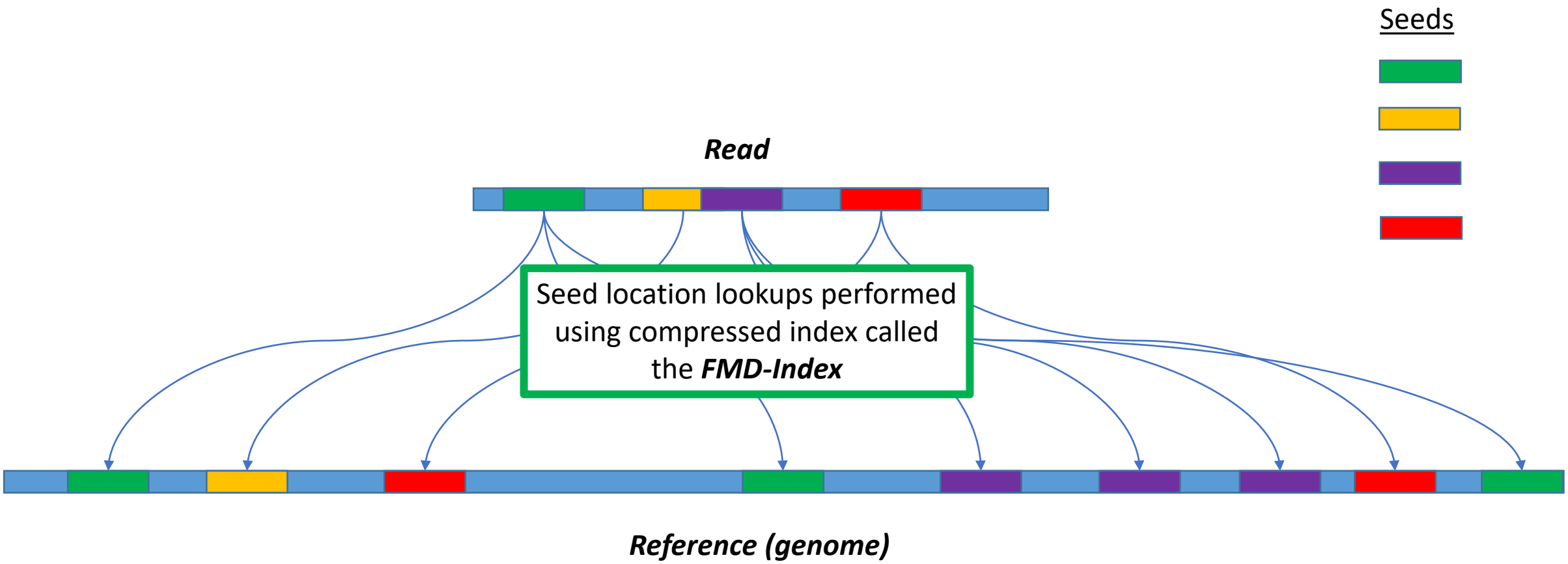
This is not



This is extremely well studied

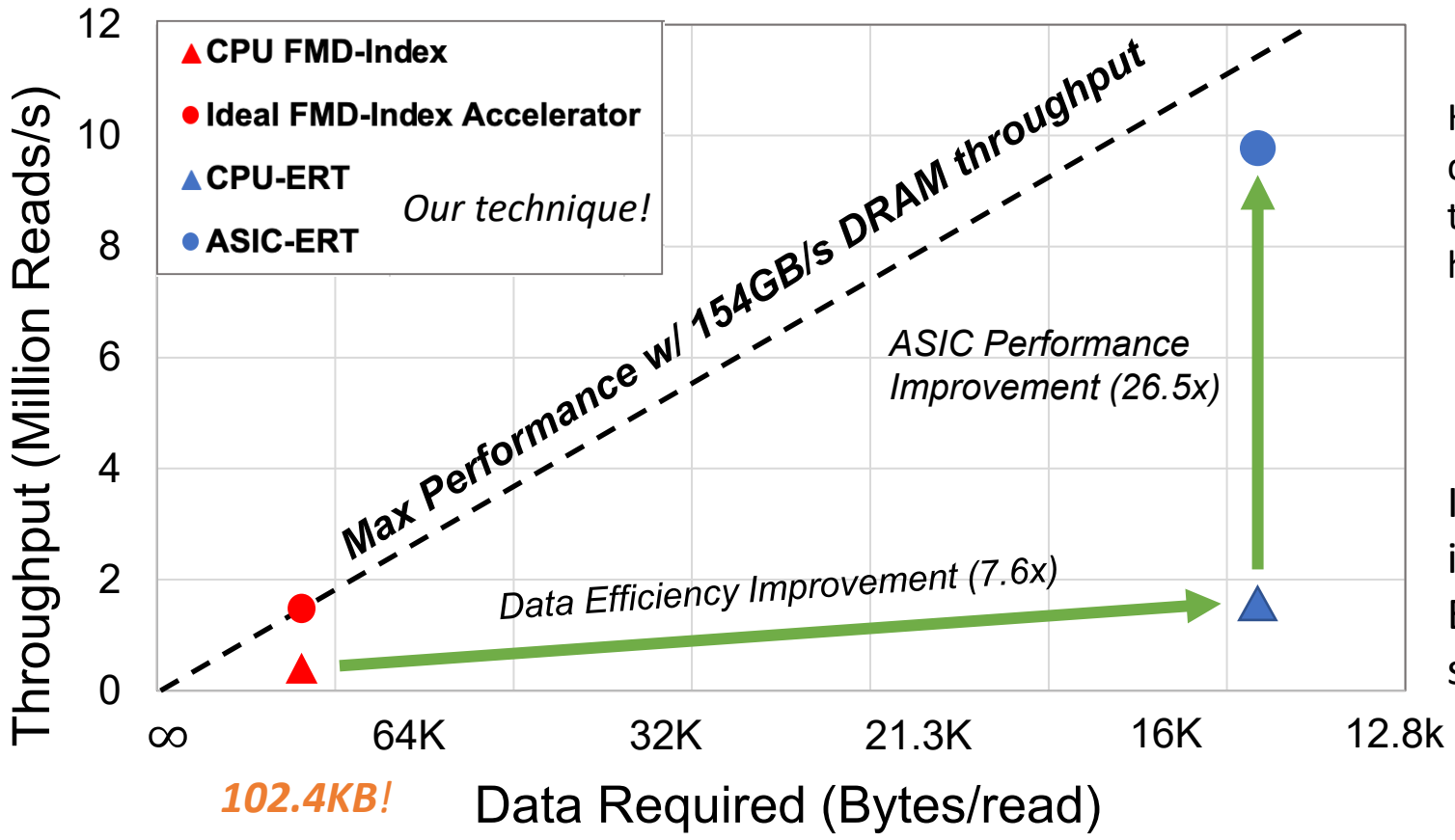
Seeding high-level overview

Seed



Roofline model shows we need to ditch the FMD-Index….

"Performance"

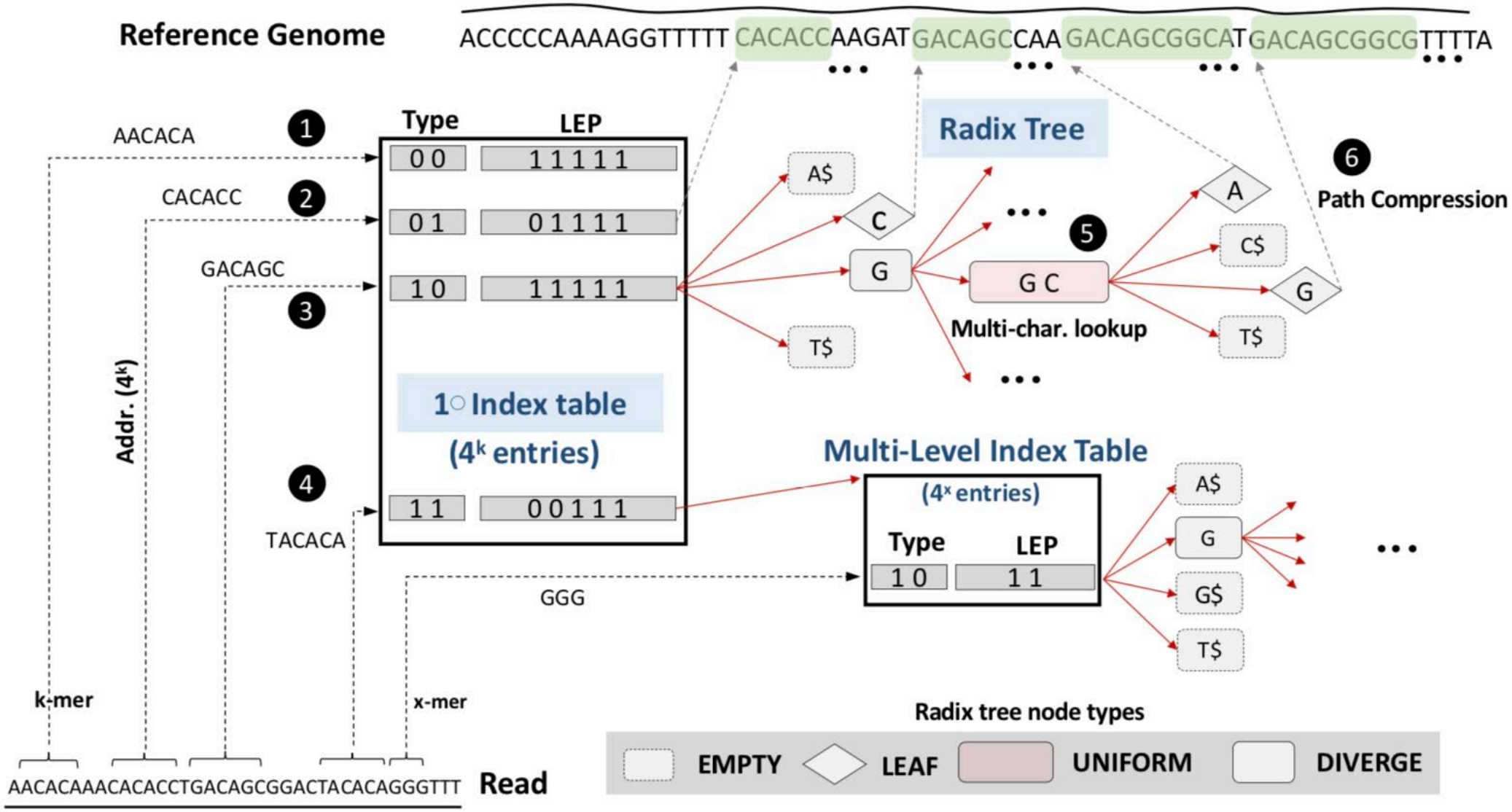


"Data Efficiency"

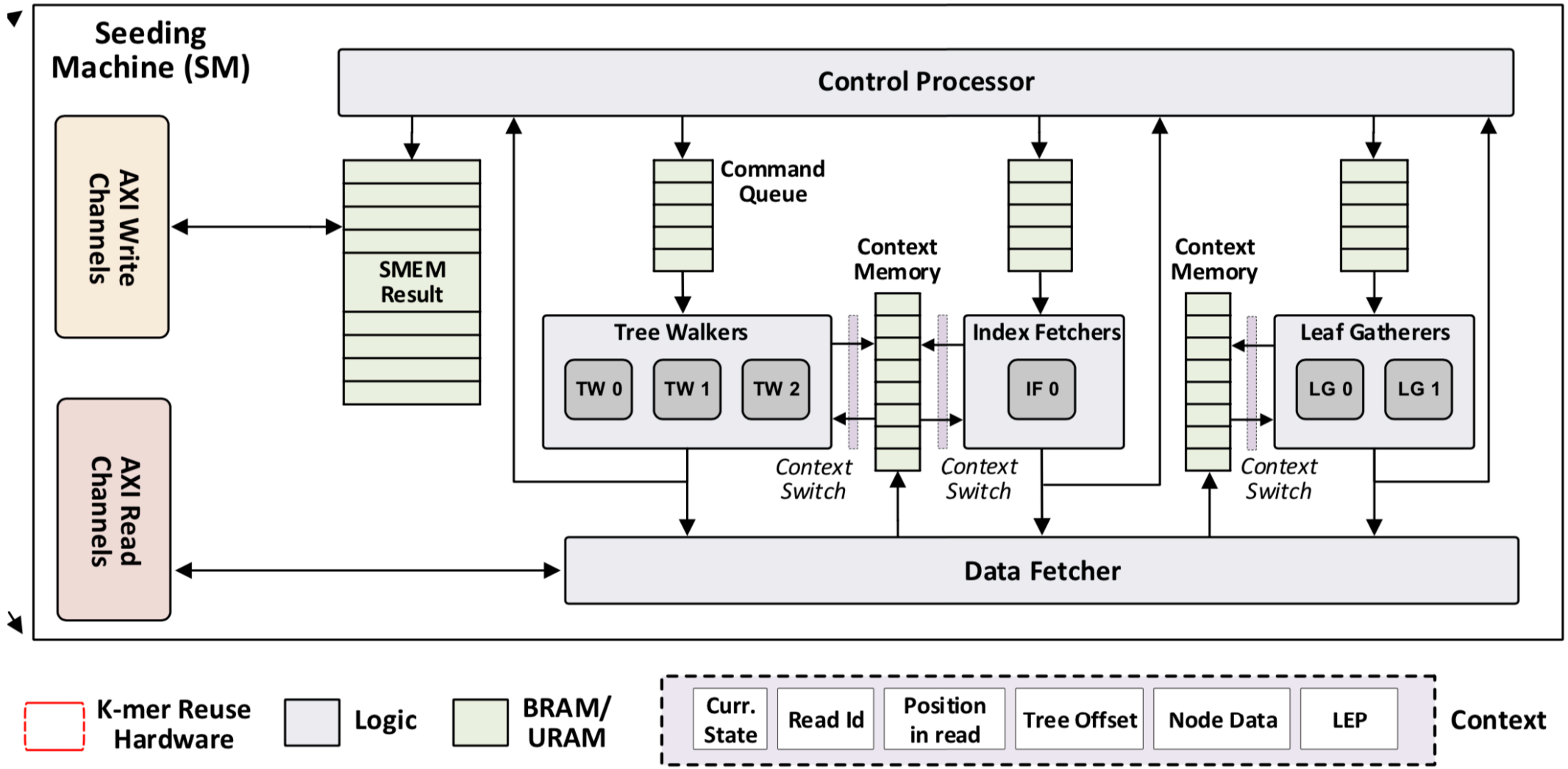
Hardware accelerator for ERT data structure lookups helps us take advantage of this added headroom!

Instead of FMD-Index, we invent a new data structure--ERT--for bandwidth efficient seeding

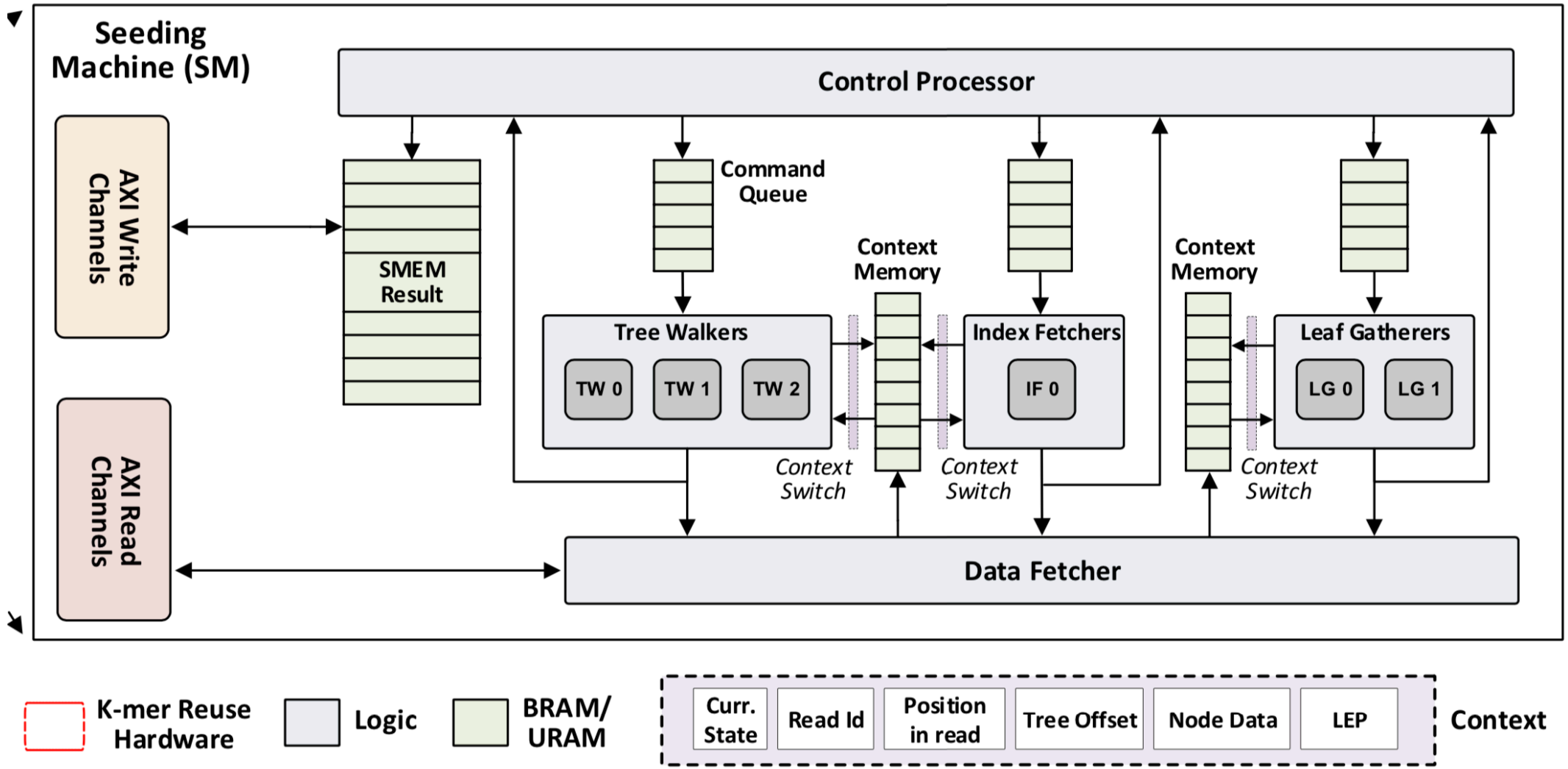
Data structure is an index into a set of trees



Accelerator is a set of ALUs for pointer chasing



Accelerator is a set of processors connected to DRAM via a high-bandwidth crossbar



How does “software” engineering work at Sequal?

UM MS’18
5+ years experience
in hardware design

Kush Goliya

U. Virginia, PhD’18
10+ years experience
in system design

Jack Wadden

- Two person team
- We sit 10 ft from each other
- We have little ”formal” software engineering background
- We write 99% Verilog/1% C

How does “software” engineering work at Sequal?

Kush Goliya

UM MS’18
5+ years experience
in hardware design

Jack Wadden

U. Virginia, PhD’18
10+ years experience
in system design

- Our codebase is so small, that we don’t have too many version control issues
- We have an integration testbench, and a production testbench that we test code against before it is pulled into the main branch
- Final verification is “do we match BWA-MEM software output on benchmark input”?

How does “software” engineering work at Sequal?

Kush Goliya

UM MS’18
5+ years experience
in hardware design

Jack Wadden

U. Virginia, PhD’18
10+ years experience
in system design

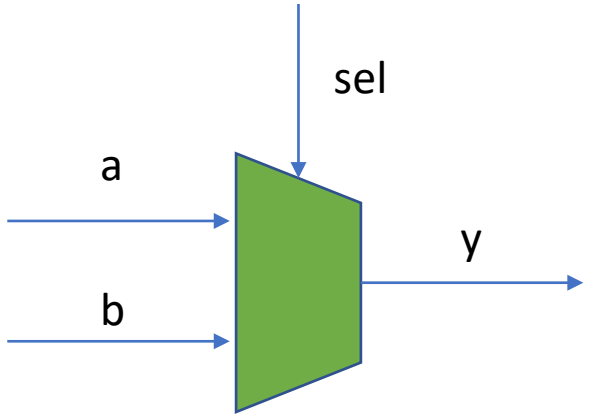
We have been working for 1.5-2yrs to build, fine-tune, re-build, re-fine-tune performance in order to meet an acceptable performance for an investor pitch demo

Background on Verilog and FPGA development

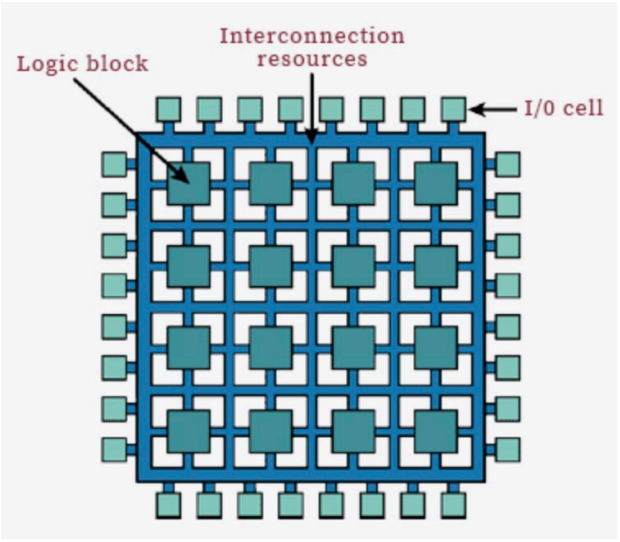
- Language designed for describing logic (not hardware)
- Consists of many parallel functions (always blocks) that all operate in parallel
- Variables in functions (always blocks) should be declared before usage
- Each parallel function is synthesized automatically into a Boolean logic network

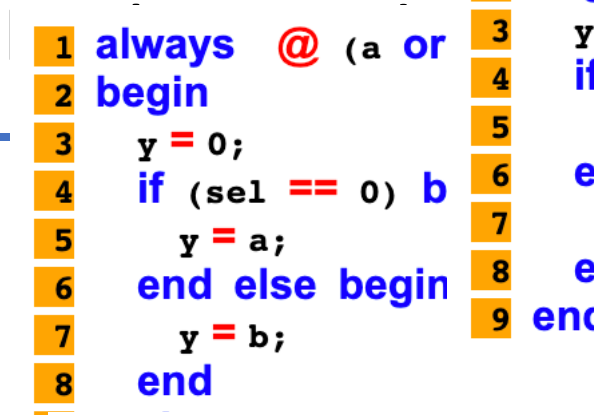
```
wire a;
wire b;
wire y;
wire sel;

always @ (a or b or sel)
begin
  y = 0;
  if (sel == 0) begin
    y = a;
  end else begin
    y = b;
  end
end
```

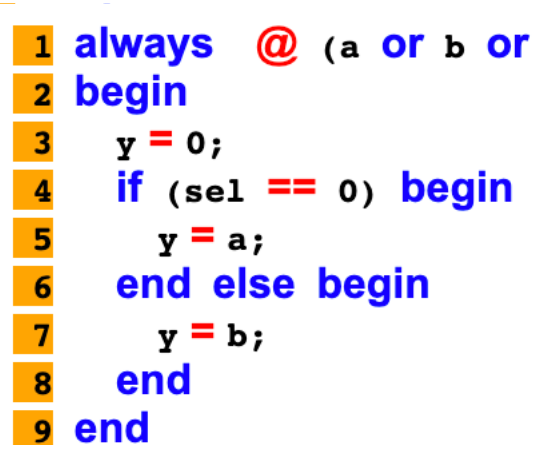


Field Programmable Gate Array (FPGA)

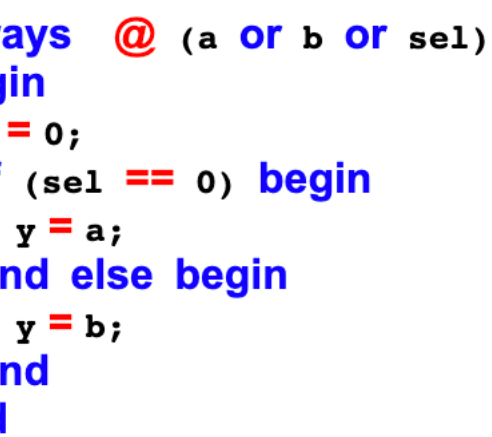




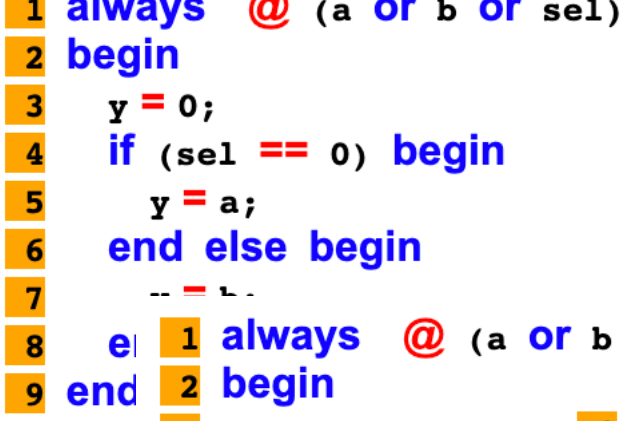
```
1 always @ (a or
2 begin
3   y = 0;
4   if (sel == 0) b
5     y = a;
6   end else begin
7     y = b;
8   end
```



```
1 always @ (a or b or
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```



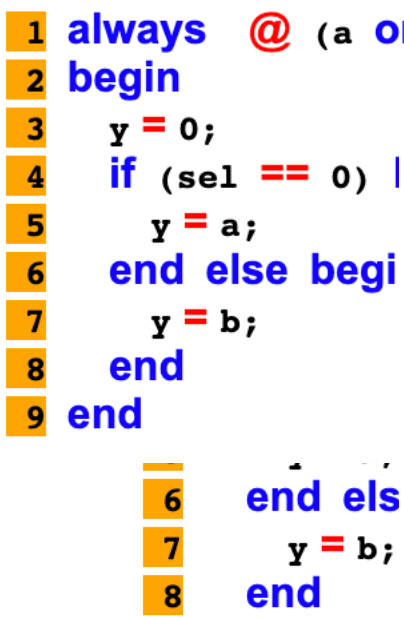
```
ays @ (a or b or sel)
gin
= 0;
 (sel == 0) begin
 y = a;
nd else begin
 y = b;
nd
```



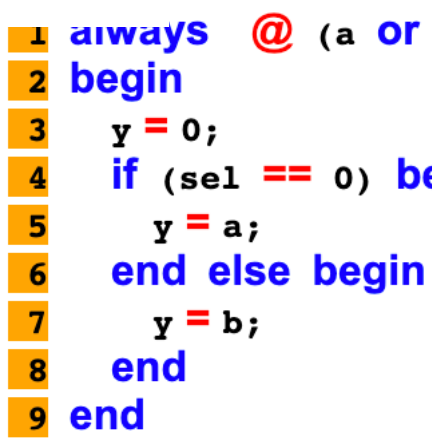
```
1 always @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```



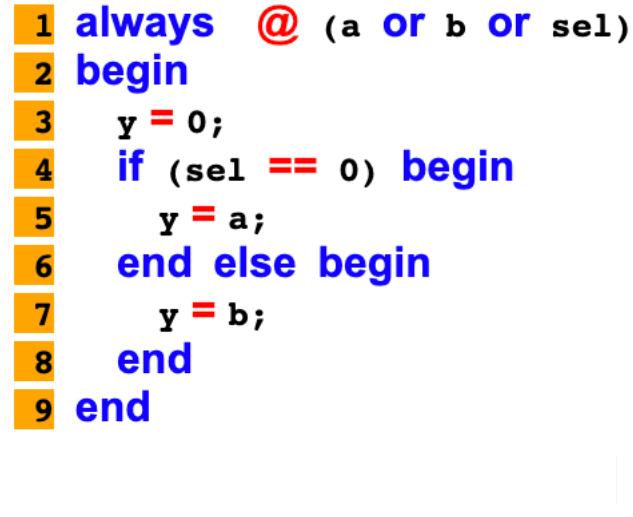
```
1 always @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```



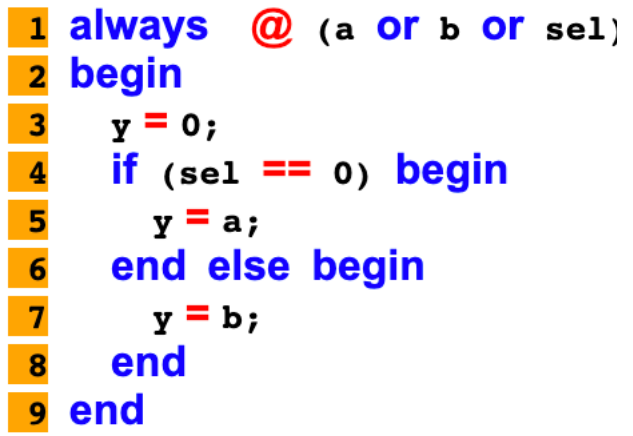
```
1 always @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) b
5     y = a;
6   end else begi
7     y = b;
8   end
9 end
```



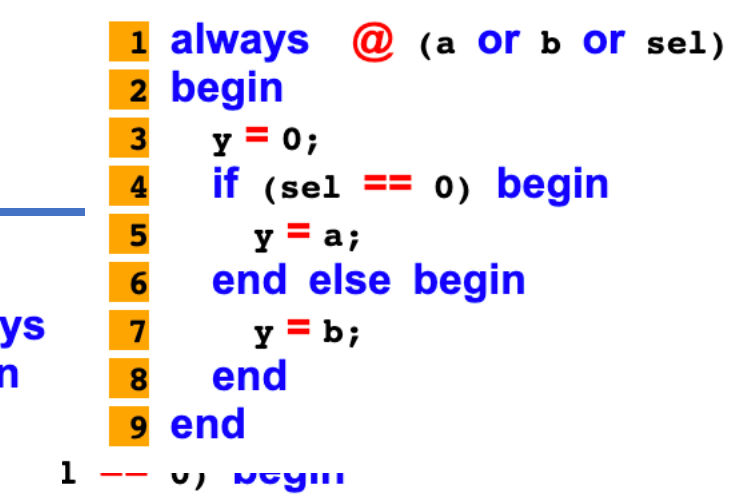
```
1 always @ (a or
2 begin
3   y = 0;
4   if (sel == 0) be
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```



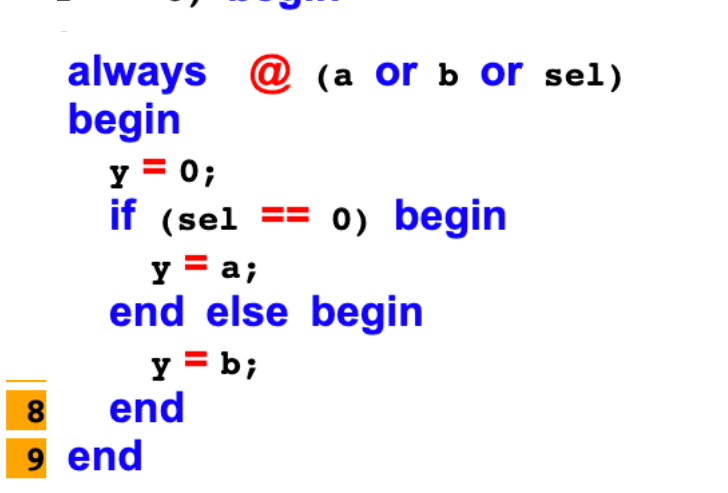
```
1 always @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```



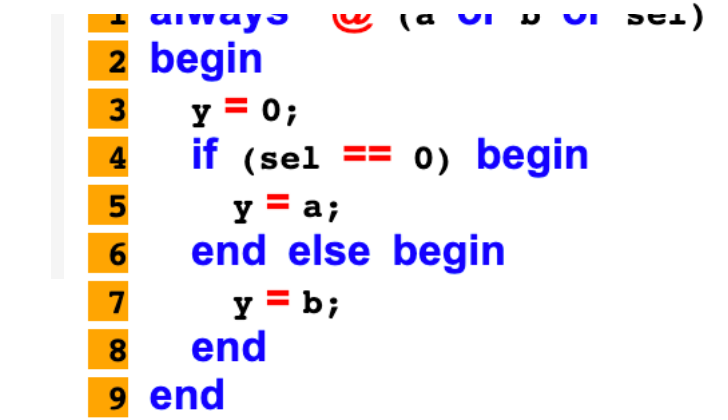
```
1 always @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```



```
1 always @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```

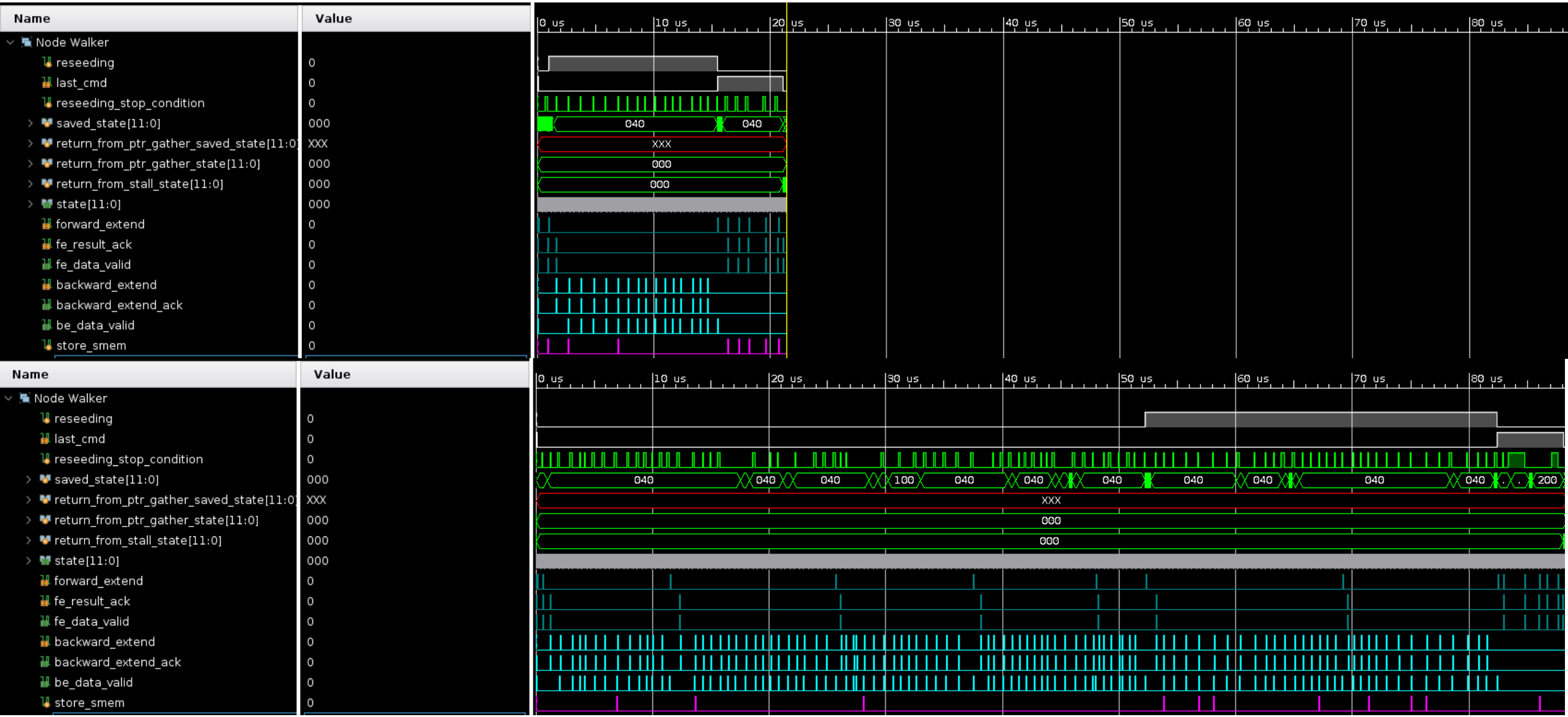


```
always @ (a or b or sel)
begin
  y = 0;
  if (sel == 0) begin
    y = a;
  end else begin
    y = b;
  end
end
```

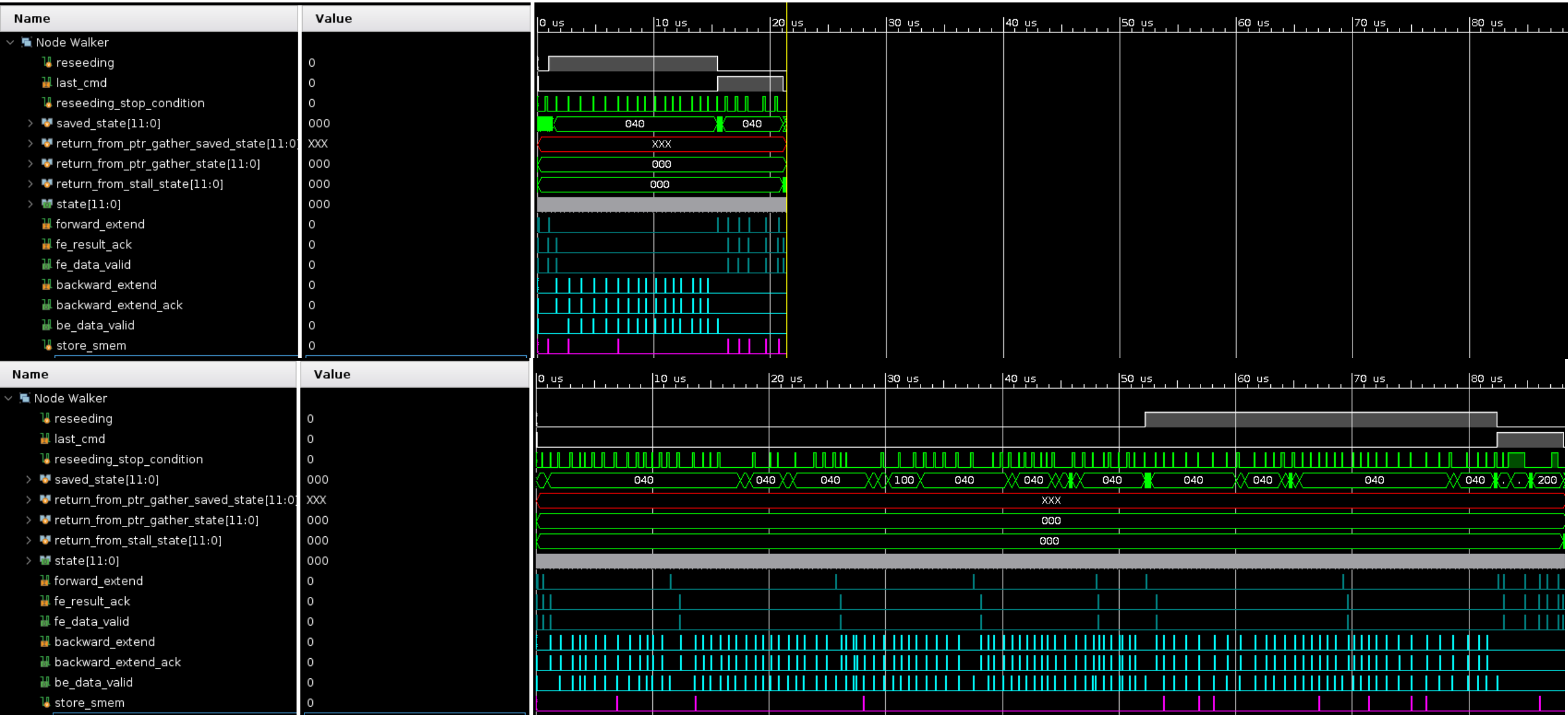


```
1 always @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```

We use wave forms to help debug massively parallel programs



We heavily leverage “testbench” debugging, but it’s not a cure all



Classic issue with hardware debugging: lack of introspection

Software Debugging

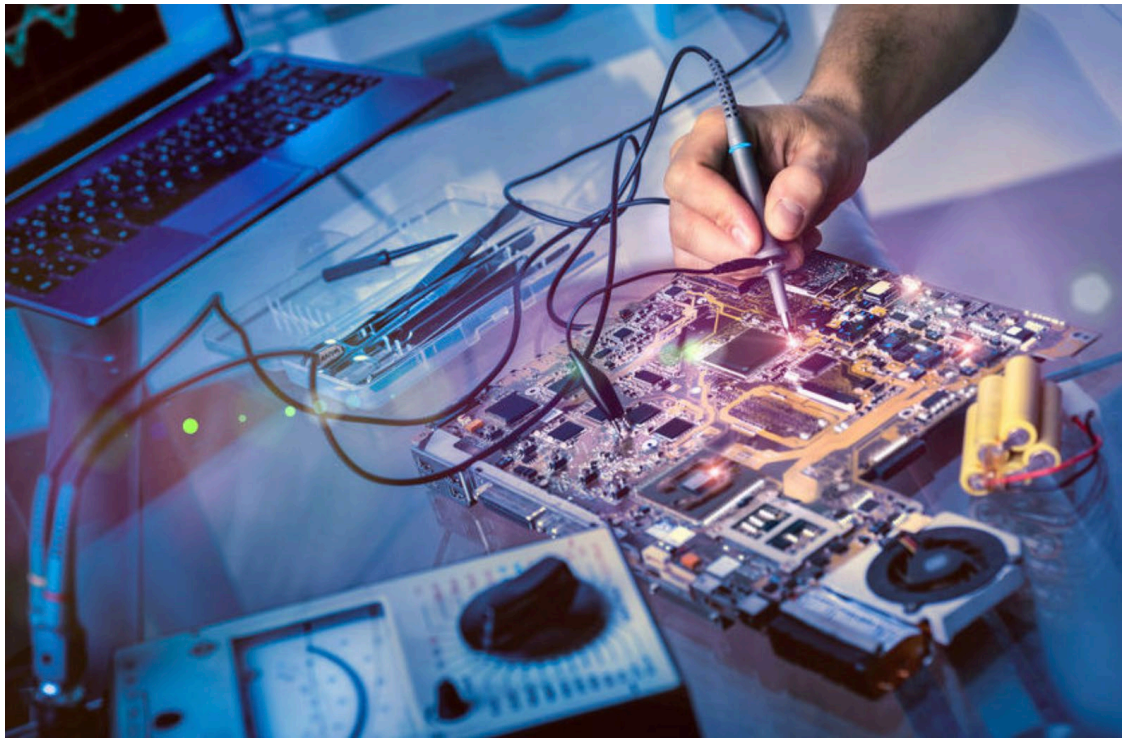
```
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x000000000040053b in factorial at example.c:4
(gdb) run
Starting program: /home/akulkarni/projects/gdb-basics/factorial

Breakpoint 1, factorial (n=5, a=1) at example.c:4
4               printf("Value of n is %d\n",n);
(gdb) condition 1 n==2
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x000000000040053b in factorial at example.c:4
        stop only if n==2
        breakpoint already hit 1 time
(gdb) continue
Continuing.
Value of n is 5
Value of n is 4
Value of n is 3

Breakpoint 1, factorial (n=2, a=60) at example.c:4
4               printf("Value of n is %d\n",n);
```

Fairly easy to know states of your program as it runs on real hardware

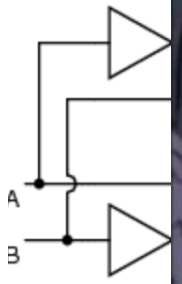
Hardware Debugging



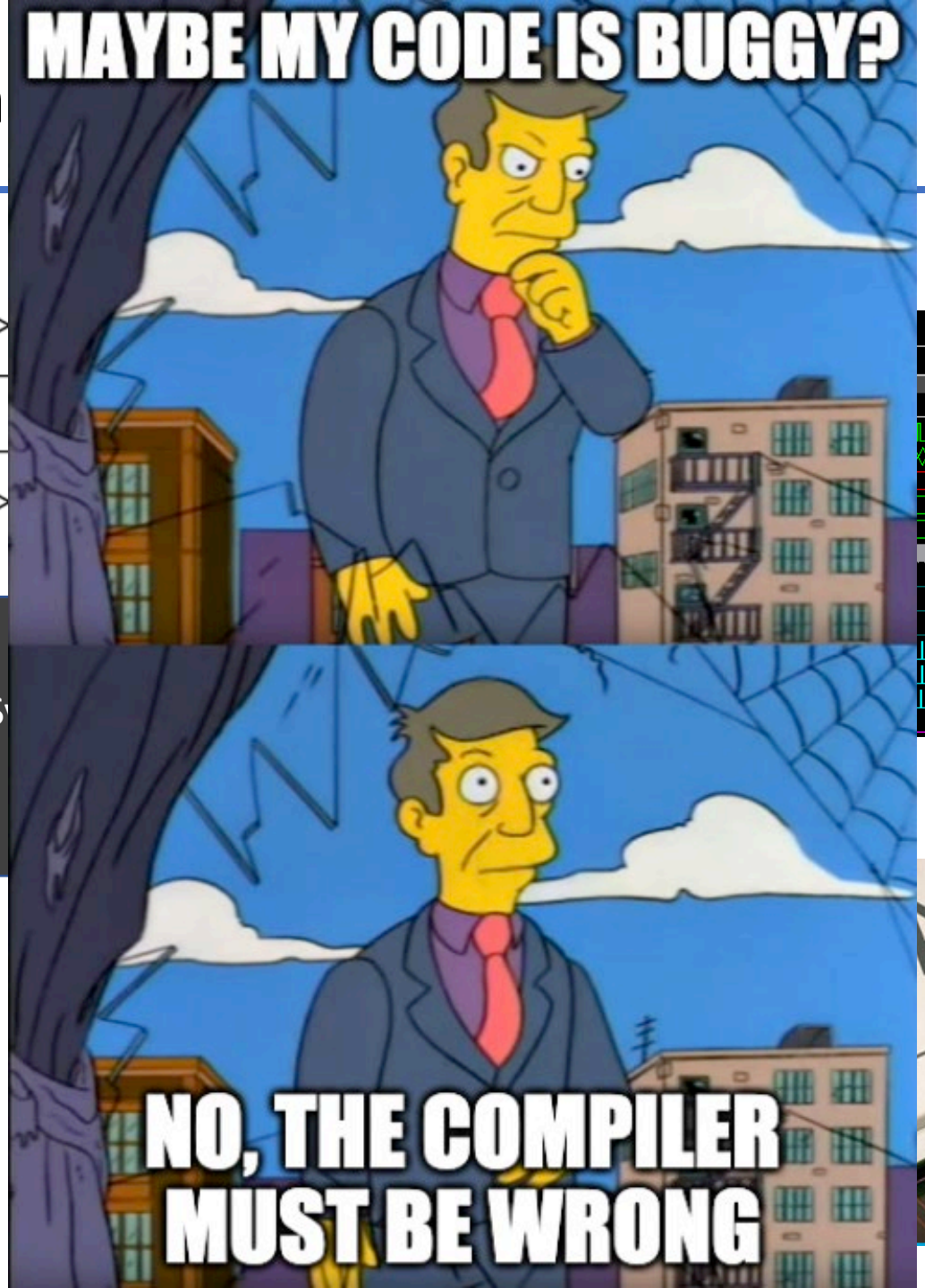
You can’t practically do this on a 10 billion transistor chip…

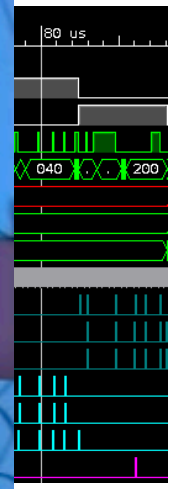
Software engineerin

```
1 always  @ (a or b or sel)
2 begin
3   y = 0;
4   if (sel == 0) begin
5     y = a;
6   end else begin
7     y = b;
8   end
9 end
```

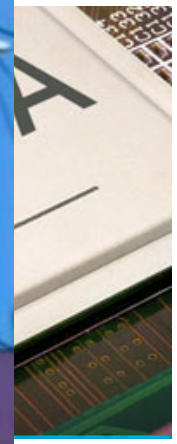


Code





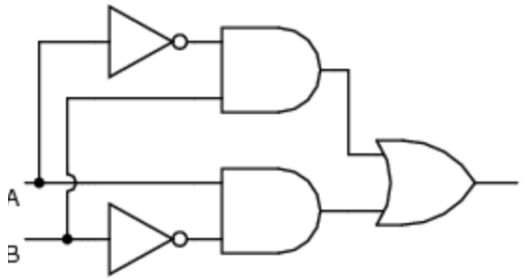
Works!



Fails!

Our toolchain assumptions were incorrect!

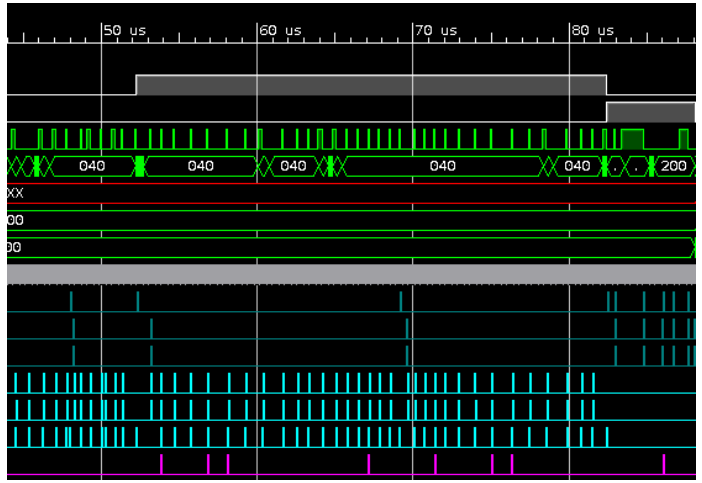
```
1 always  @ (a or b or sel)
2 begin
3    y = 0;
4    if (sel == 0) begin
5       y = a;
6    end else begin
7       y = b;
8    end
9 end
```



Code

Synthesis Tool

Simulation



Works!

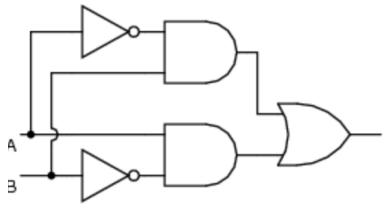
Real Hardware

Fails!

Our toolchain assumptions were incorrect!

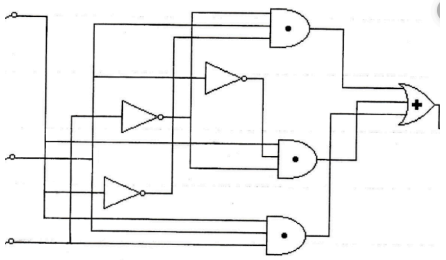
```
1 always  @ (a or b or sel)
2 begin
3    y = 0;
4    if (sel == 0) begin
5       y = a;
6    end else begin
7       y = b;
8    end
9 end
```

Code

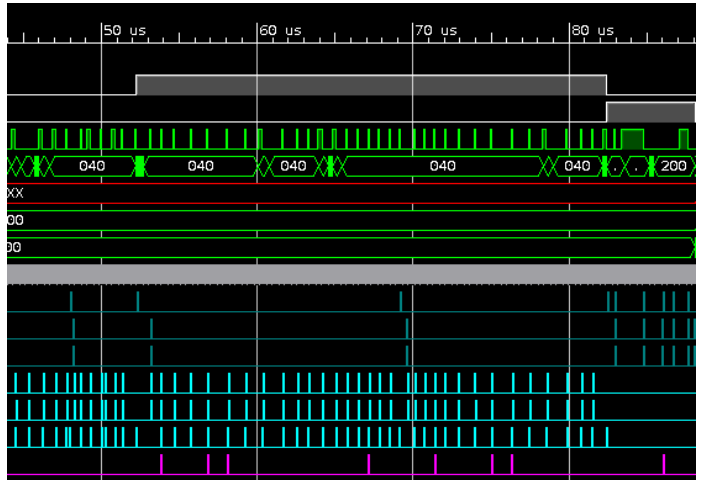


Synthesis Tool

Synthesis Tool



Simulation



Works!

Real Hardware

Fails!

Bug came from a difference in how Verilog is compiled

```
wire [7:0] eight_bit_bus;

always @(*) begin

    eight_bit_bus = 0;

end
```

```
always @(*) begin

    eight_bit_bus = 0;

end

wire [7:0] eight_bit_bus;
```

Warning: variable “eight_bit_bus” assigned before declaration

Well, seems fine…. simulation works….

This took us **TWO WEEKS** to figure out…. What is the lesson here?

- **Verilog?**

> If an identifier appears on the left-hand side of a continuous assignment statement, and that identifier has not been declared previously in the scope where the continuous assignment statement appears or in any scope whose declarations can be directly referenced from the scope where the continuous assignment statement appears (see 23.9), then an implicit scalar net of default net type shall be assumed. See 10.3 for a discussion of continuous assignment statements.

  - tl;dr "If you haven't declared it yet, we'll just assume the type to be Boolean and move on"
  - WHYWHYWHYWHYWHYWHY

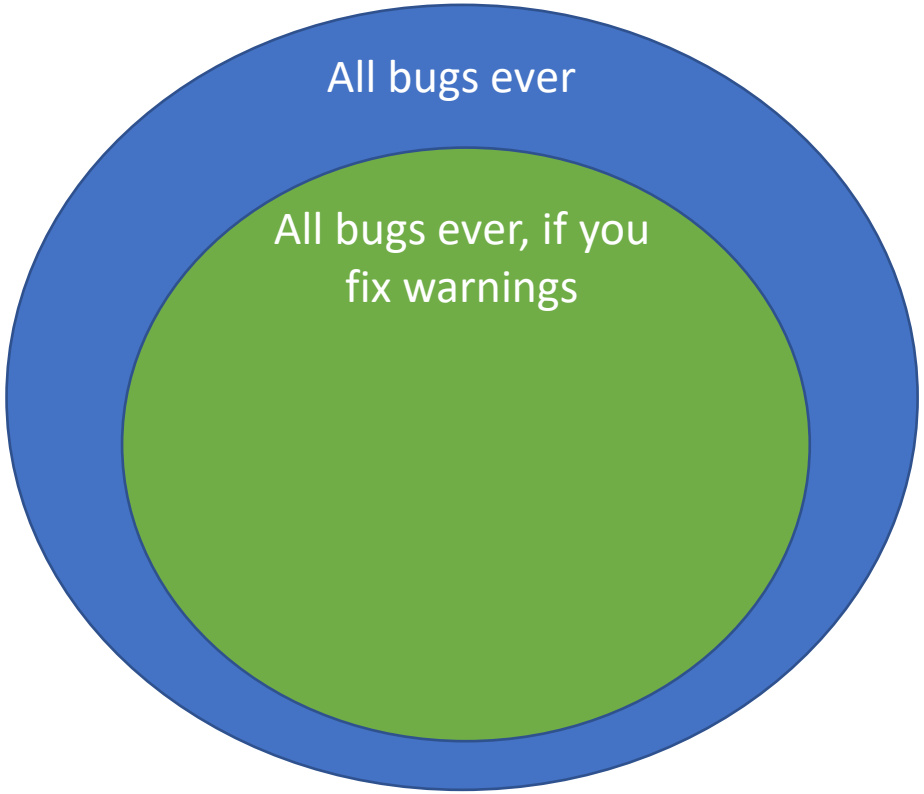
- **Programmers?**
  - Should have looked at warnings and fixed them all…. but simulation was working!
- **Vivado Simulation?**
  - Does ***not*** behave according to the official spec but has more "reasonable" behavior
- **Vivado Hardware?**
  - WHY IS THIS NOT AN ERROR
  - But…to be fair…. it was behaving correctly according to the (terrible) spec

New Practice:

- Warnings are errors. Period. Fix them all.
- Fix all simulation and hardware synthesis warnings before simulation and real-hardware testing.

Lesson: try to reduce the number of ways you can shoot yourself in the foot

Simple “best practices” are designed to reduce the scope of bugs you will ever run into



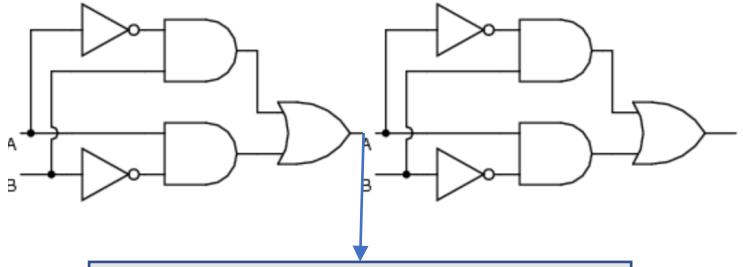
Software engineering struggle/story #2

It works!

“Great!
Make it faster…”

What are the bottlenecks?

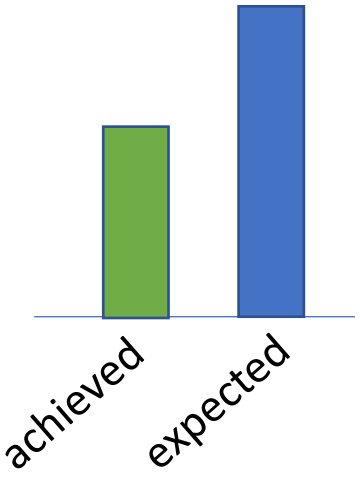
Add Performance Counters!

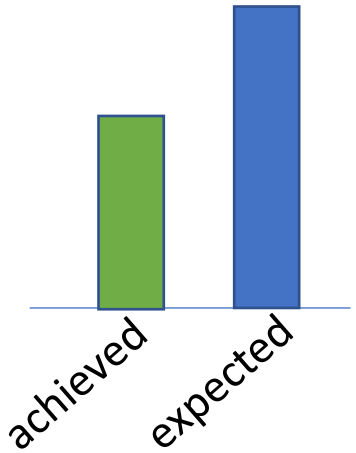


cache misses

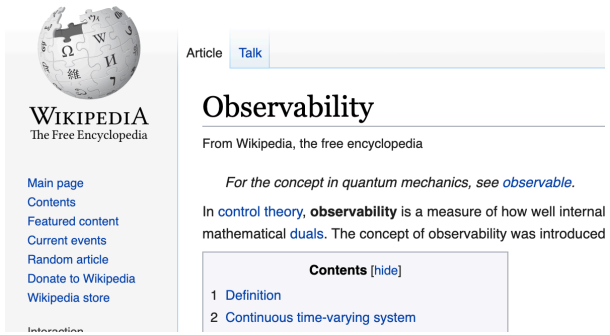
Won’t compile at the same frequency…

Maybe simulation will tell us?





How do we know what’s wrong when we can’t take accurate measurements?



Take-away Lessons

Hardware is notoriously difficult to debug because…

- Circuits are inherently massively parallel “programs”
- Difficult to simulate
- Little to no introspection once you move to real hardware
- Has terrible, and neglected programming languages and development tools

Future directions for software engineering within the company

- Use formal verification tools (just like software community!)
- Set up continuous verification tools (just like software community!)
- Port codebase to higher-level hardware description languages like Chisel, or HLS

Questions?

Ask me about

- DNA sequencing ethics
- Being a part of such a small company
- Working with doctors as customers
- The future of sequencing technology and society
- Taxes