High-Performance Scientific Computing
Lecture 11: GPU Performance, Applications

MATH-GA 2011 / CSCI-GA 2945 · November 21, 2012

Today

Tool of the day: Advanced Version Control

GPU performance

Outline

Tool of the day: Advanced Version Control

GPU performance

Version control demo time

Outline

Tool of the day: Advanced Version Control

GPU performance

- Less control, more data
- GPUs and Latency
- Understanding GPUs

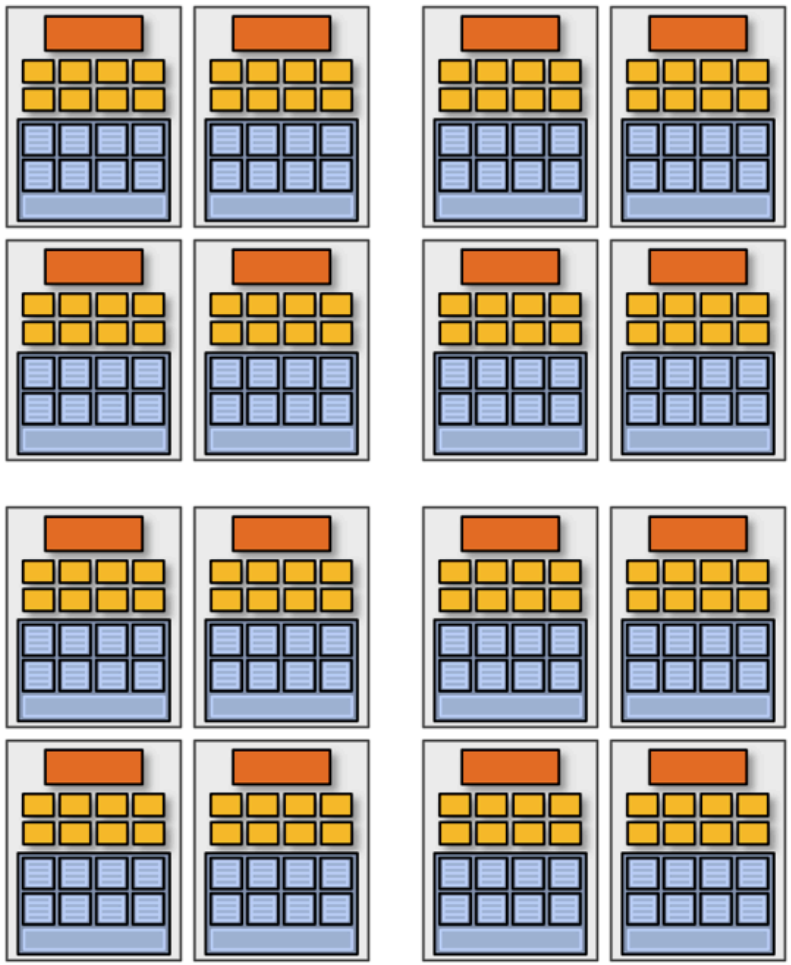
Outline

Tool of the day: Advanced Version Control

GPU performance

- Less control, more data
- GPUs and Latency
- Understanding GPUs

Gratuitous Amounts of Parallelism!



Credit: Kayvon Fatahalian (Stanford)

Gratuitous Amounts of Parallelism!

Example:

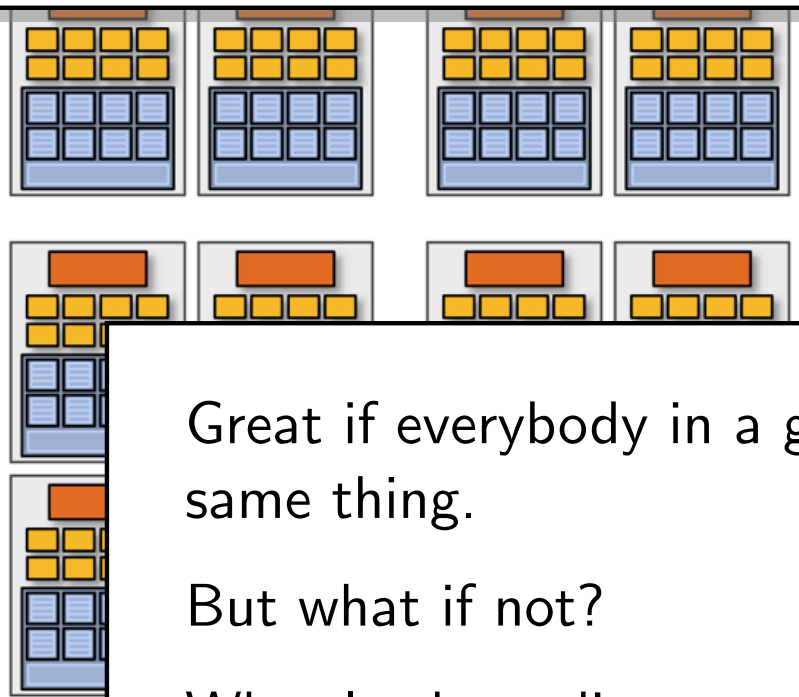
128 instruction streams in parallel
16 independent groups of 8 synchronized streams

Credit: Kayvon Fatahalian (Stanford)

Gratuitous Amounts of Parallelism!

Example:

128 instruction streams in parallel
16 independent groups of 8 synchronized streams



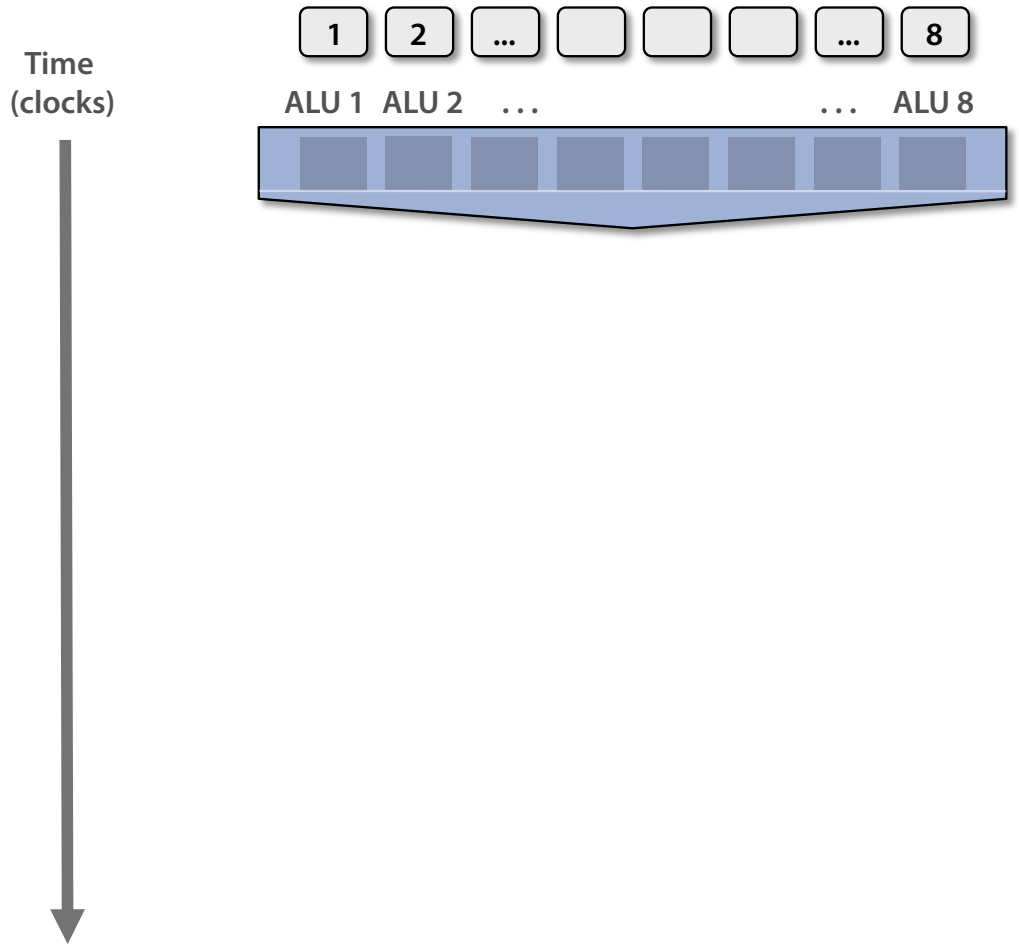
Great if everybody in a group does the same thing.

But what if not?

What leads to divergent instruction streams?

Credit: Kayvon Fatahalian (

Branches



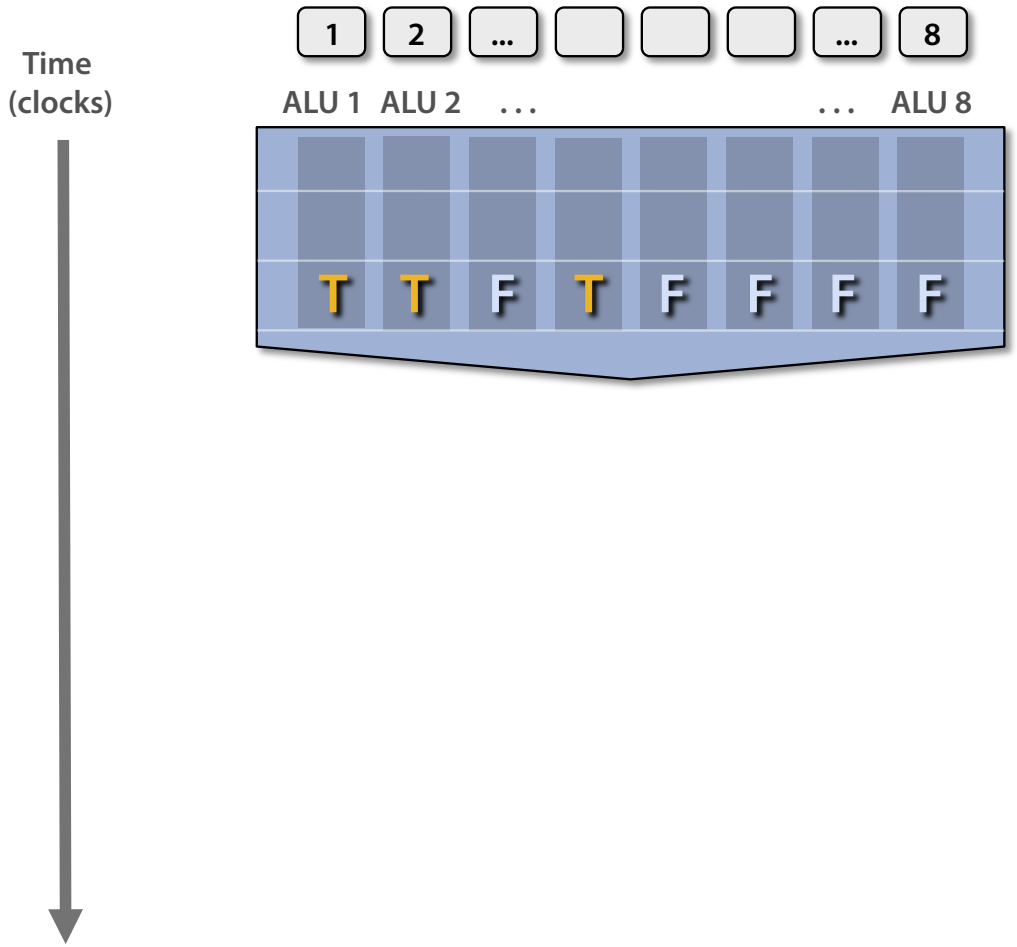

```
<unconditional
shader code>

if (x > 0) {
    y = pow(x, exp);
    y *= Ks;
    refl = y + Ka;
} else {
    x = 0;
    refl = Ka;
}

<resume unconditional
shader code>
```

Credit: Kayvon Fatahalian (Stanford)

Branches



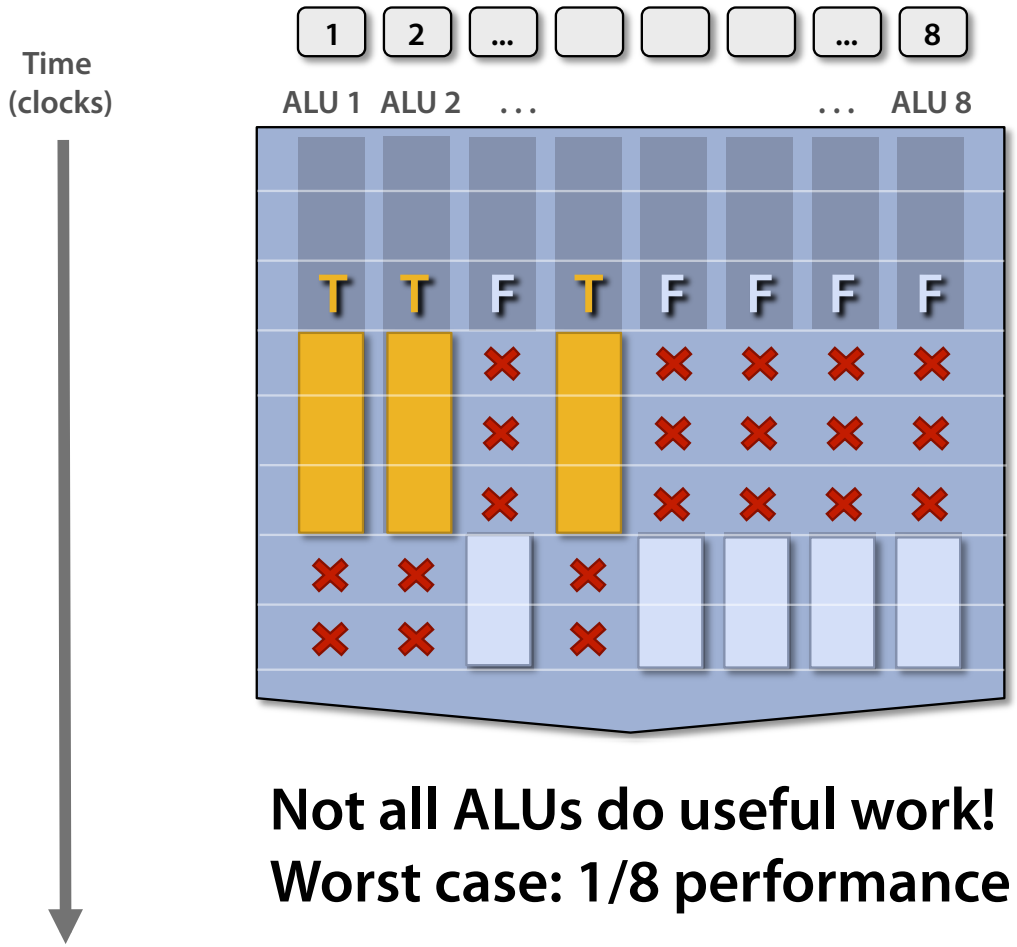

```
<unconditional
shader code>

if (x > 0) {
    y = pow(x, exp);
    y *= Ks;
    refl = y + Ka;
} else {
    x = 0;
    refl = Ka;
}

<resume unconditional
shader code>
```

Credit: Kayvon Fatahalian (Stanford)

Branches



Not all ALUs do useful work!
Worst case: 1/8 performance

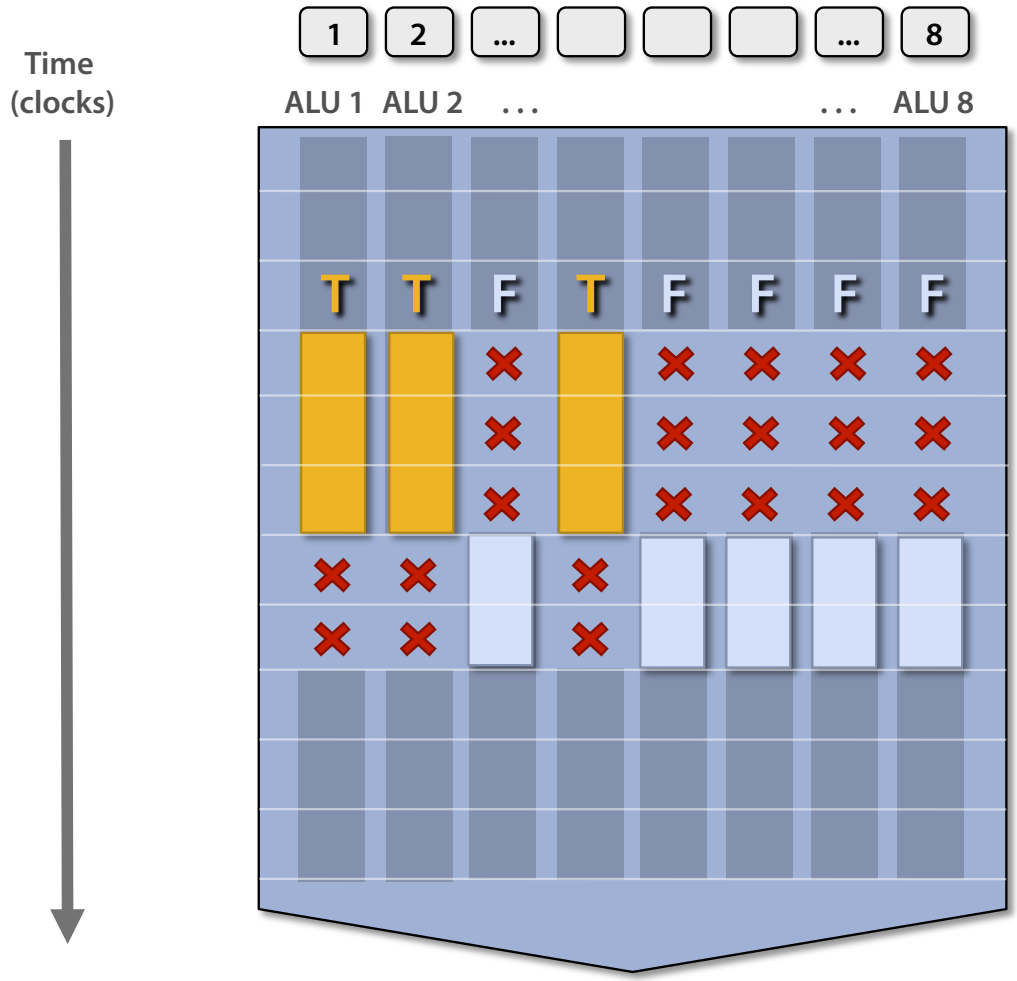
```
<unconditional
shader code>

if (x > 0) {
    y = pow(x, exp);
    y *= Ks;
    refl = y + Ka;
} else {
    x = 0;
    refl = Ka;
}

<resume unconditional
shader code>
```

Credit: Kayvon Fatahalian (Stanford)

Branches




```
<unconditional
shader code>

if (x > 0) {
    y = pow(x, exp);
    y *= Ks;
    refl = y + Ka;
} else {
    x = 0;
    refl = Ka;
}

<resume unconditional
shader code>
```

Credit: Kayvon Fatahalian (Stanford)

Branch demo time

Outline

Tool of the day: Advanced Version Control

GPU performance
- Less control, more data
- GPUs and Latency
- Understanding GPUs

GPUs vs Latency

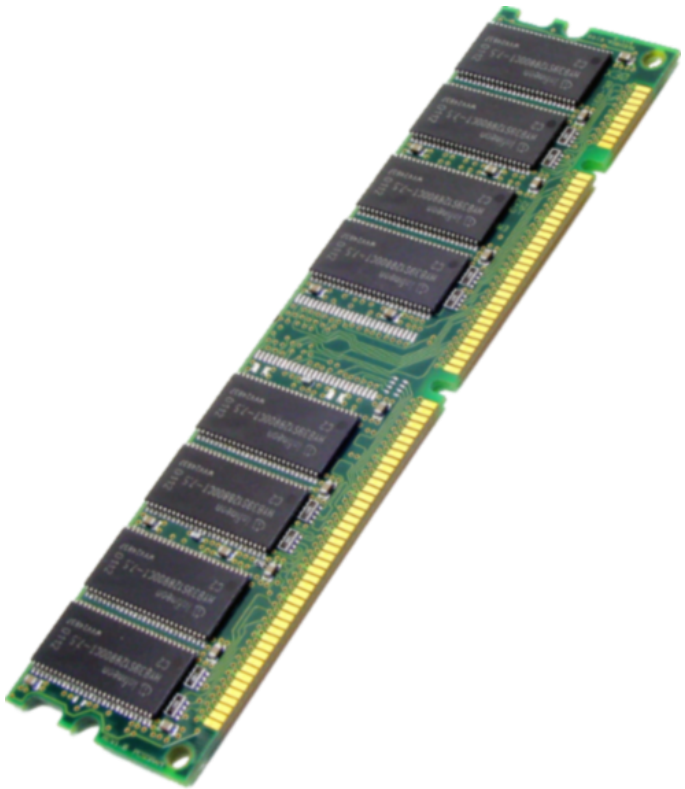
Problem

Memory still has very high latency. . .
. . . as do many other things. . .
. . . but we've removed most of the hardware that helps us deal with that.

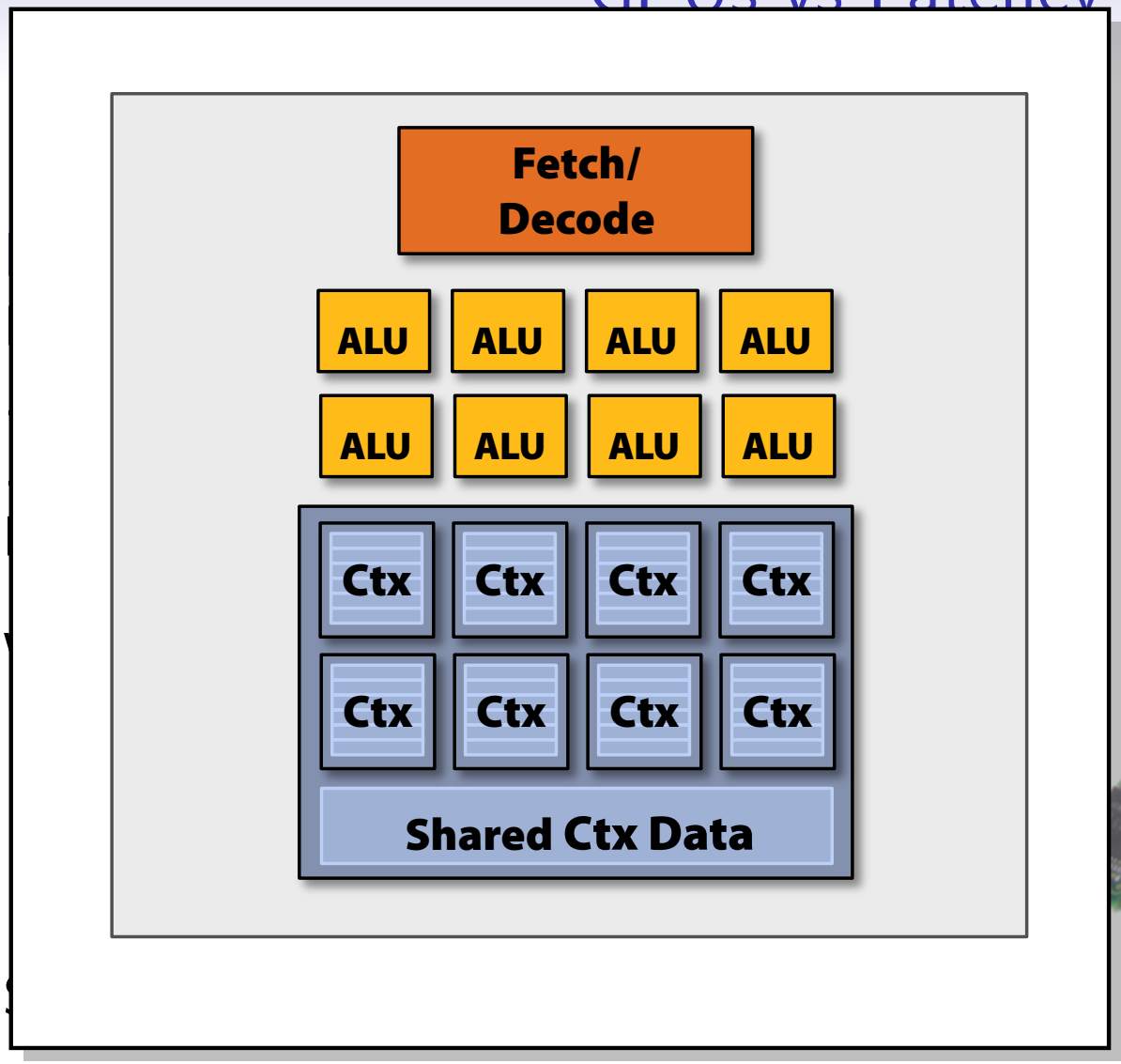
We've removed

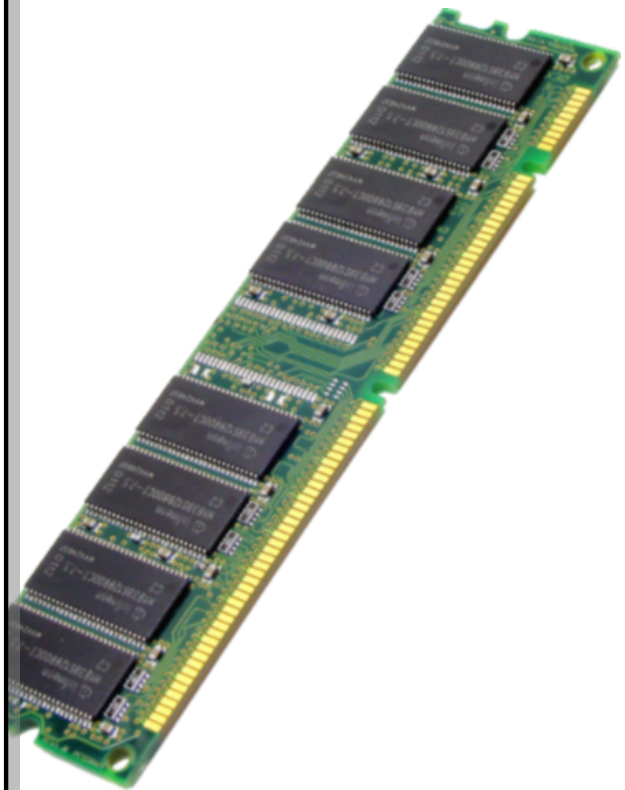
- caches
- branch prediction
- out-of-order execution

So what now?

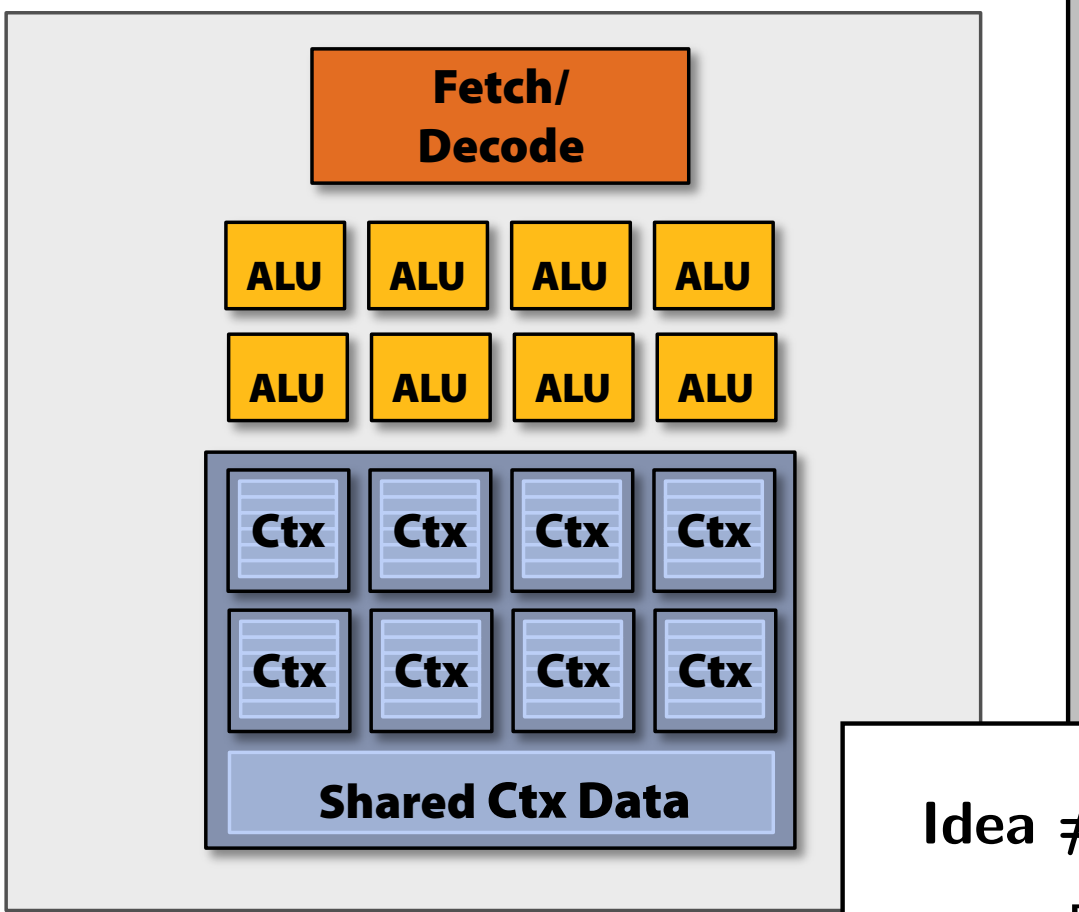


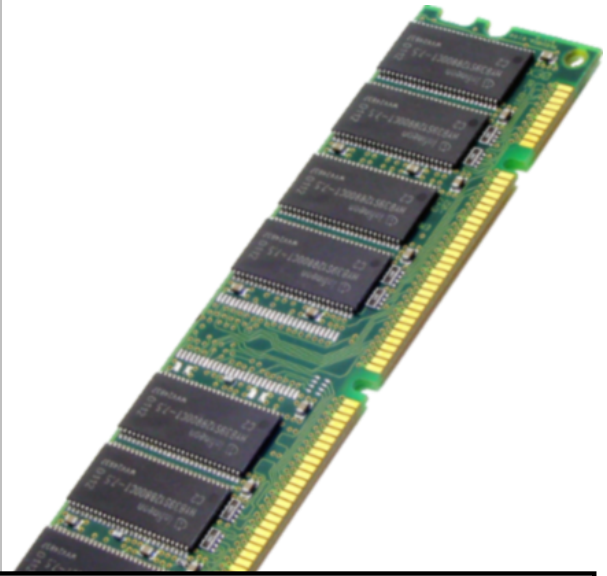
GPUs vs Latency





GPUs vs Latency

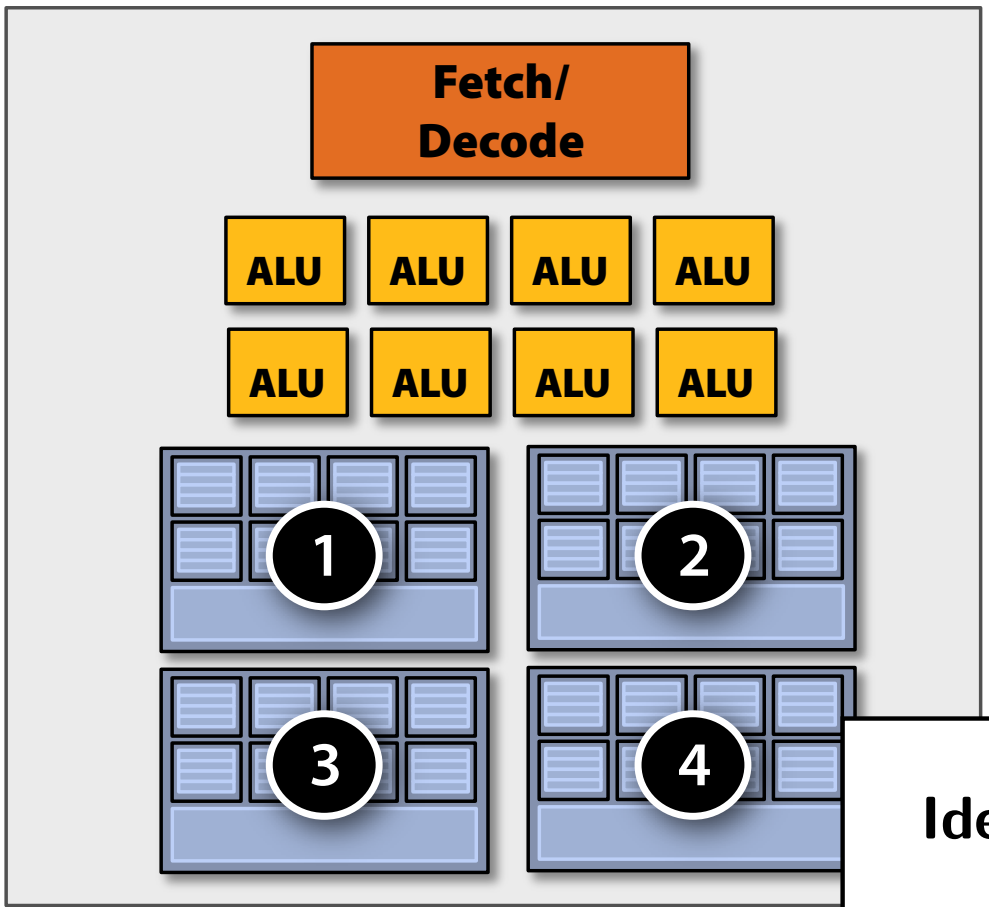


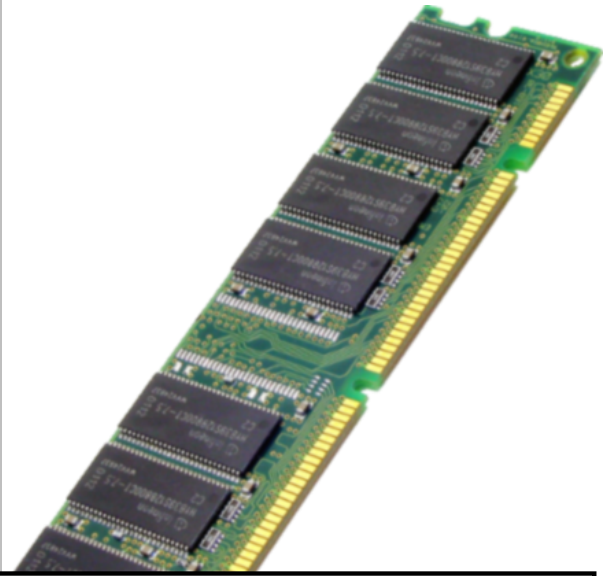


Idea #3

	Even more parallelism
+	Some extra memory
=	A solution!

GPUs vs Latency

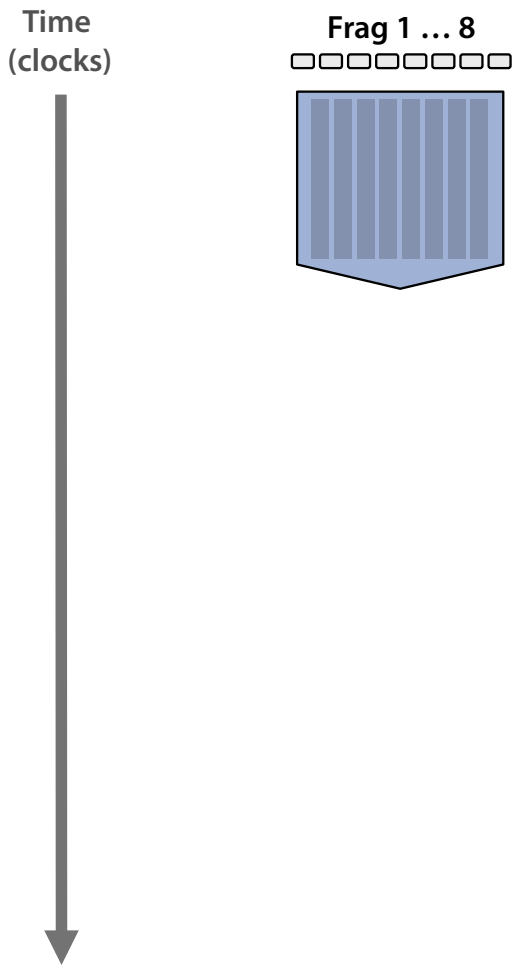


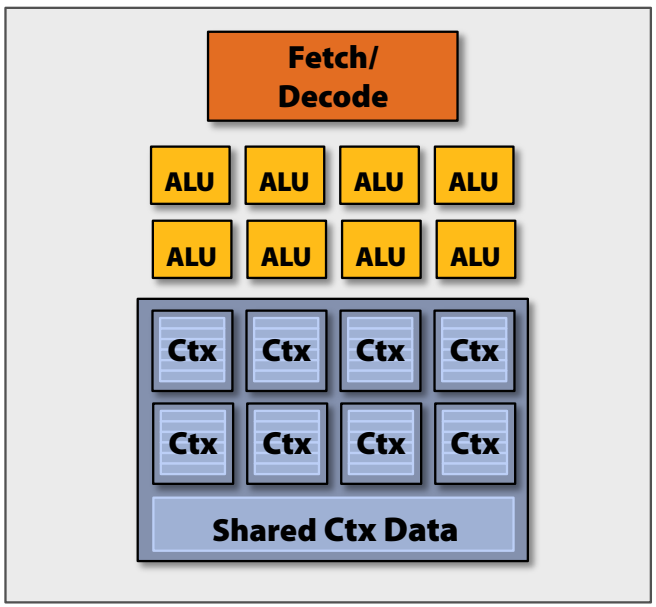


Idea #3

	Even more parallelism
+	Some extra memory
=	A solution!

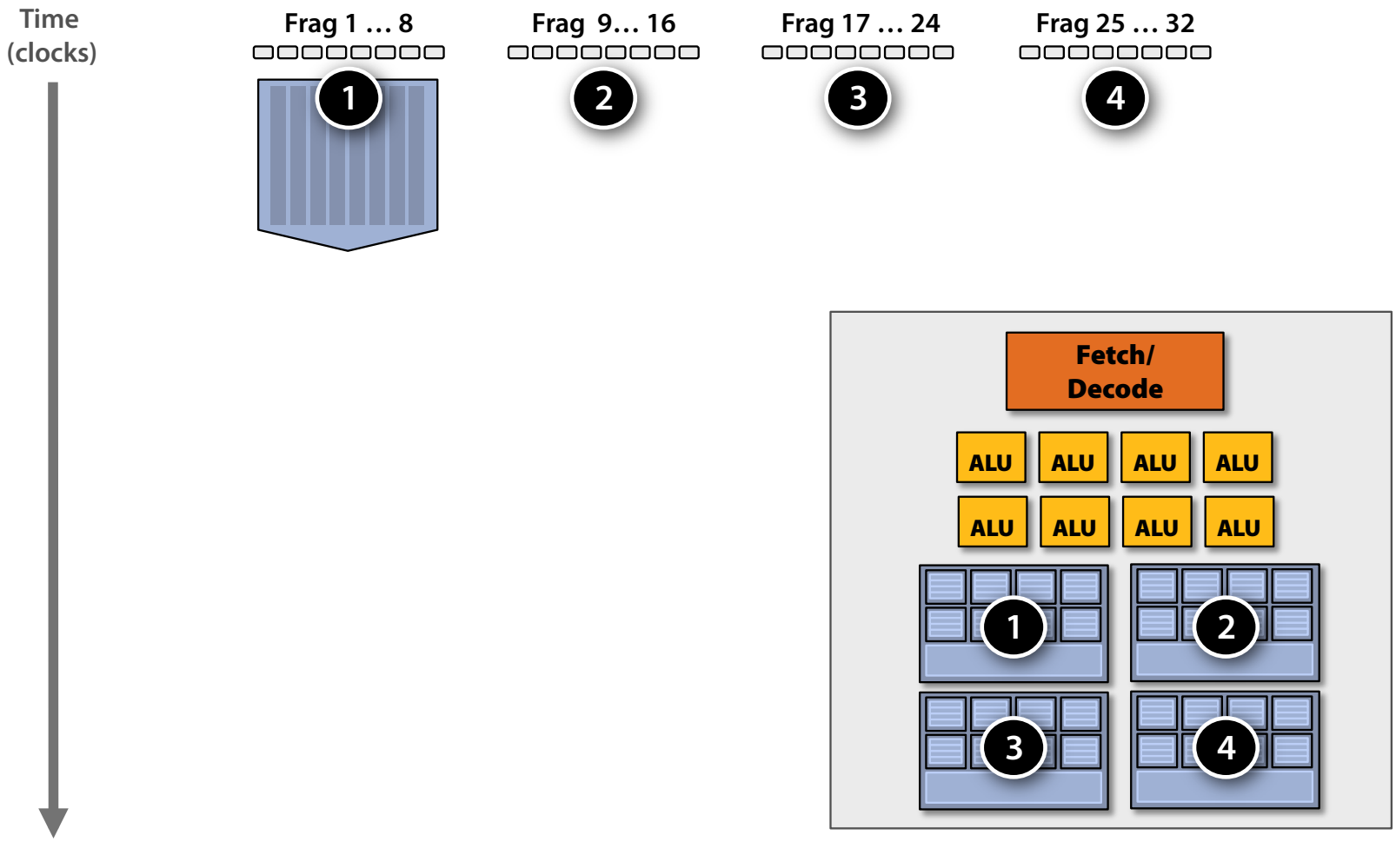
Hiding Memory Latency





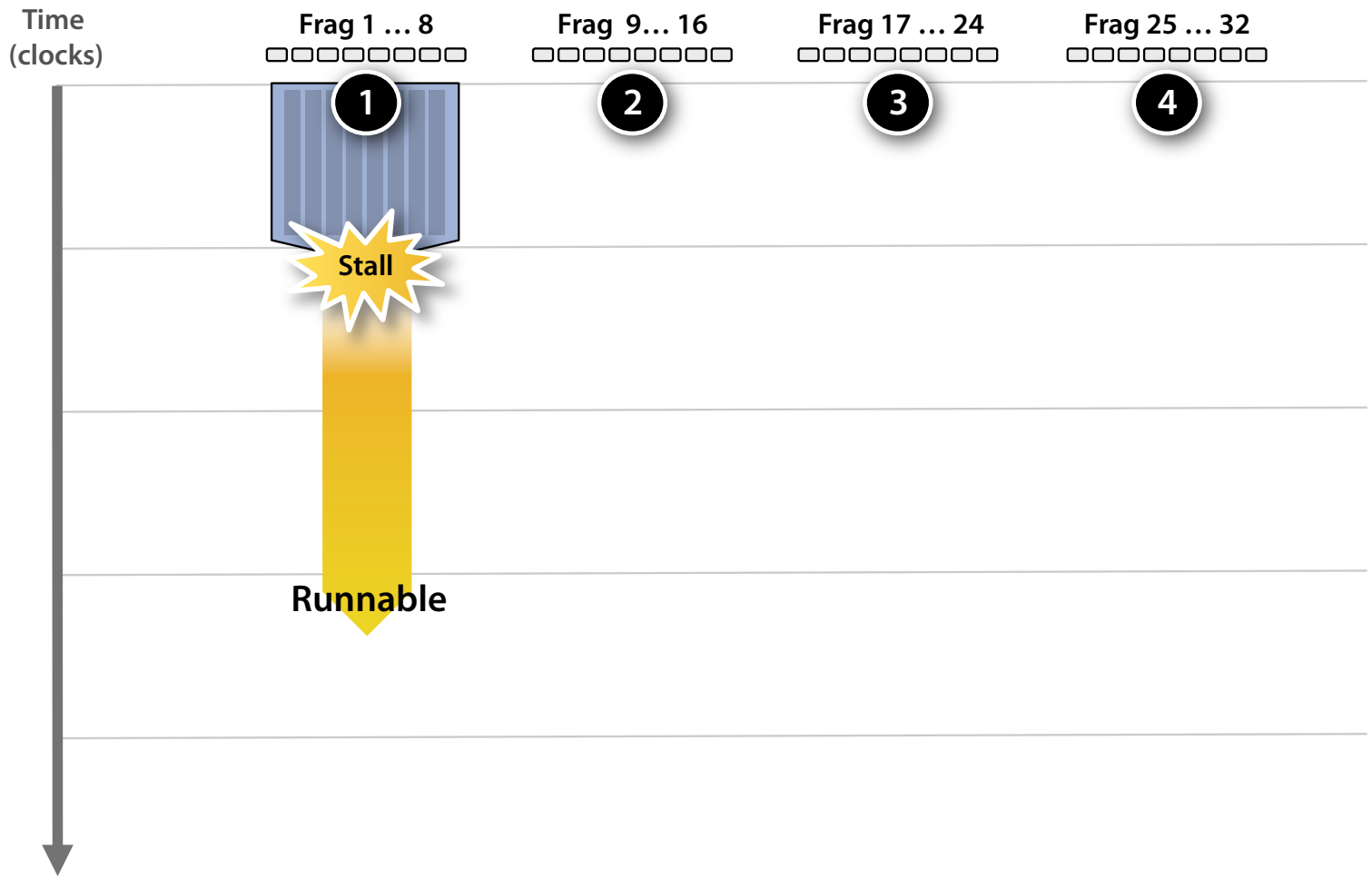
Credit: Kayvon Fatahalian (Stanford)

Hiding Memory Latency



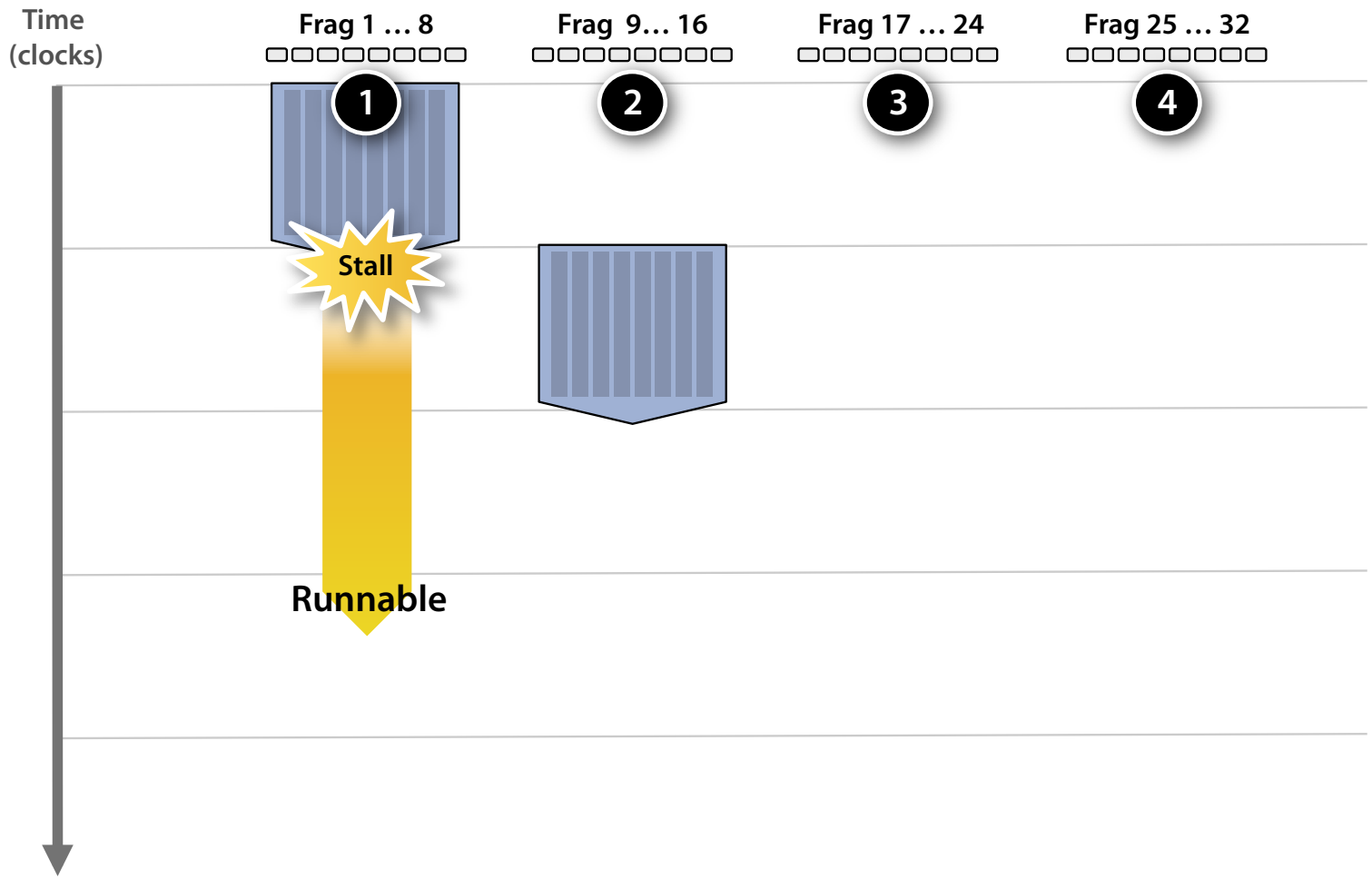
Credit: Kayvon Fatahalian (Stanford)

Hiding Memory Latency



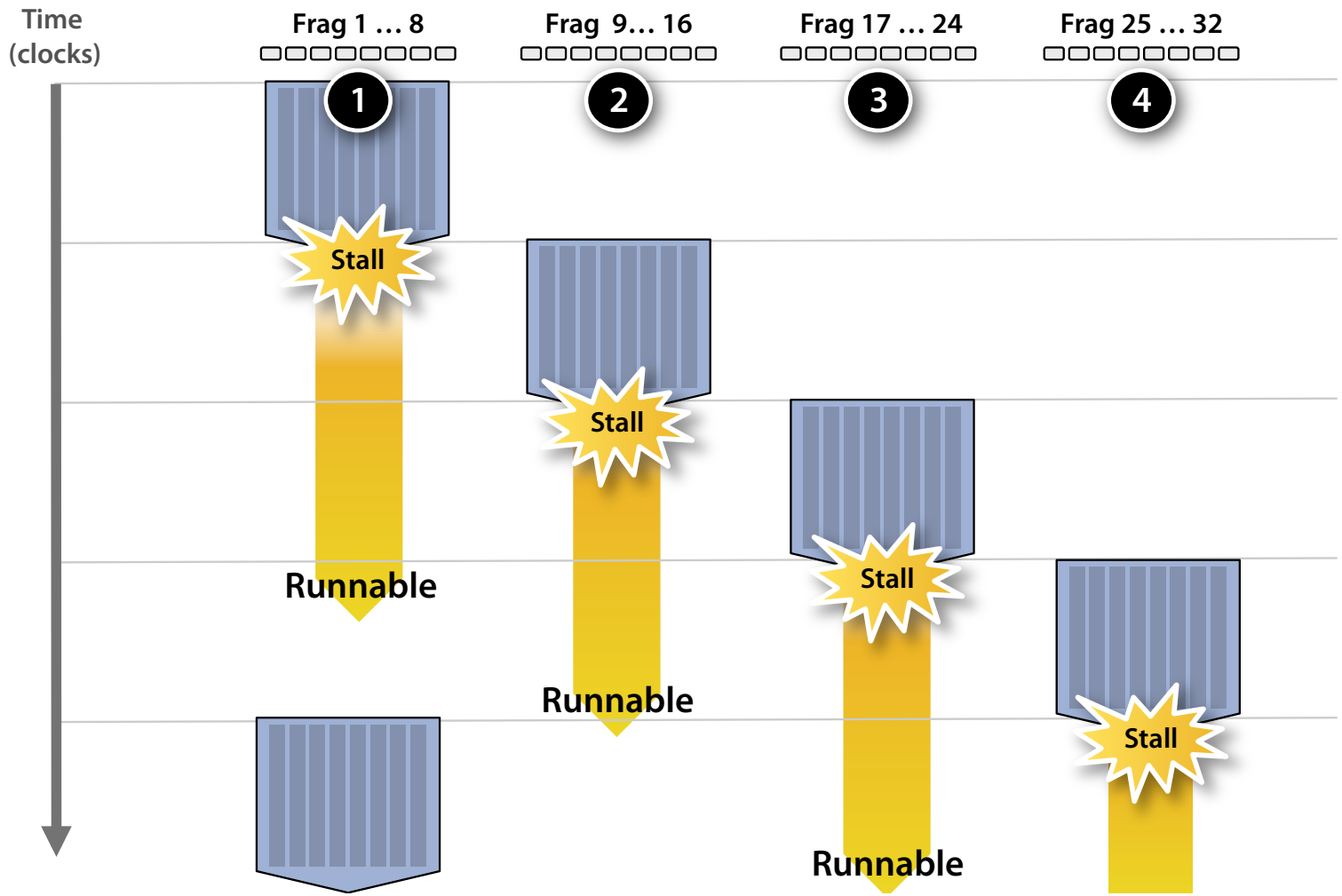
Credit: Kayvon Fatahalian (Stanford)

Hiding Memory Latency



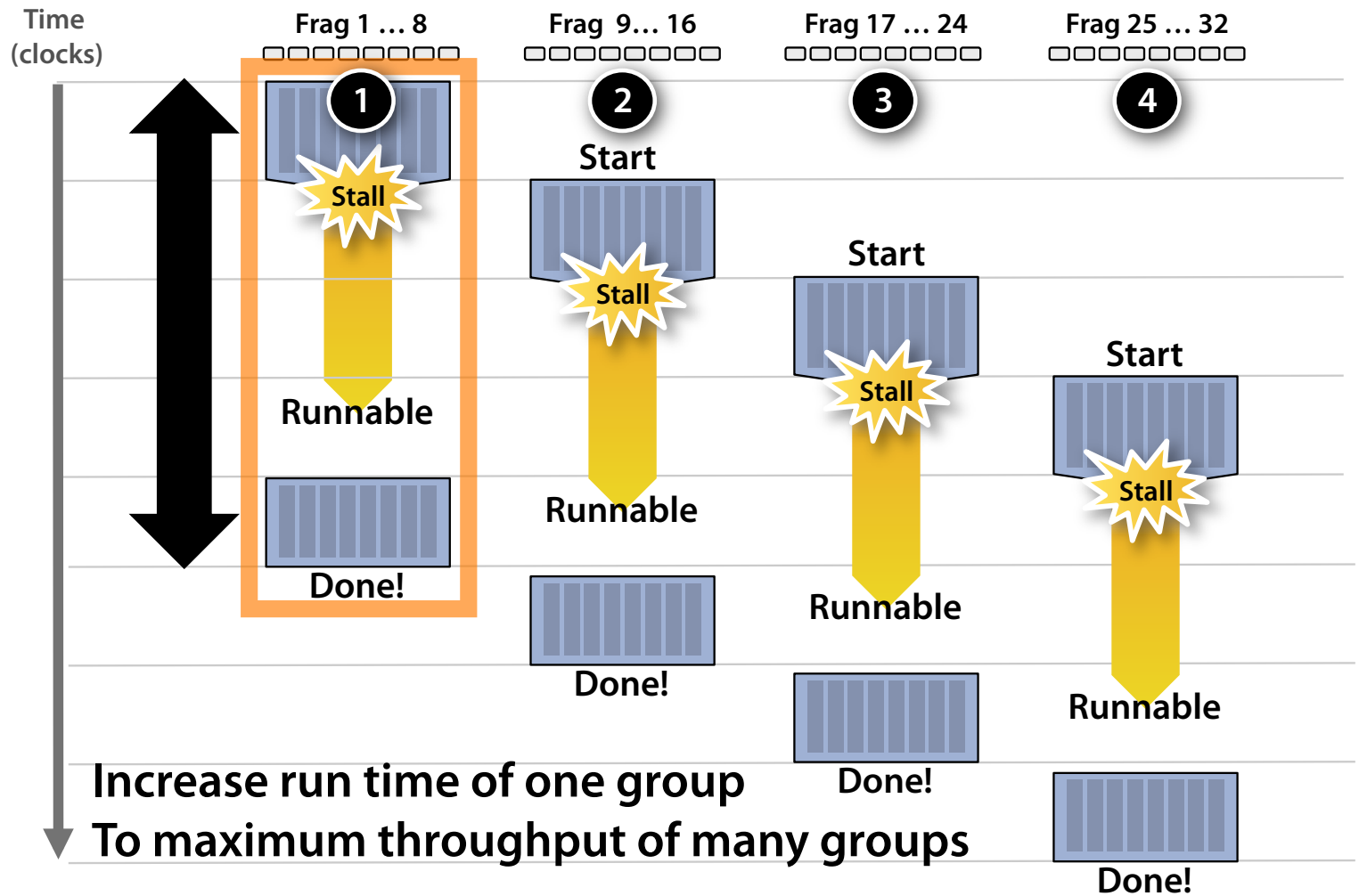
Credit: Kayvon Fatahalian (Stanford)

Hiding Memory Latency



Credit: Kayvon Fatahalian (Stanford)

Hiding Memory Latency



Credit: Kayvon Fatahalian (Stanford)

GPUs and latency demo

Outline

Tool of the day: Advanced Version Control

GPU performance
- Less control, more data
- GPUs and Latency
- Understanding GPUs

Comparing architectures

	GF100	GF104	GK104	GCN	Units
# Warps/Wavefronts	48	48	64	40	
Warp Size	32	32	32	64	W.Item
SP FLOP/clock	64	96	384	128	
Clock	700	650	823	925	MHz
Reg File	128	128	256	256	kiB
Lmem	64	64	64	64	kiB
Lmem BW	64	64	128	128	B/clock

David Kanter / Realworldtech.com

Architecture by the numbers demo

Questions?

?