Protocol Independence

Nick McKeown
Stanford University

Where did SDN ideas start?

Stanford gets too much credit

Roots are in 4D and RCP

- Rexford, Greenberg, Zhang, Maltz, …

SANE/Ethane from Stanford + Berkeley

Many concepts from Nicira

- Network OS: NOX
- OpenFlow
- Distributed OS: ONIX

New concepts all the time

- Berkeley, Berlin, Cornell, Gatech
- Princeton, Stanford, Urbana, …

Is SDN any network device with behavior defined in software?

Is SDN anything with an OpenFlow interface?

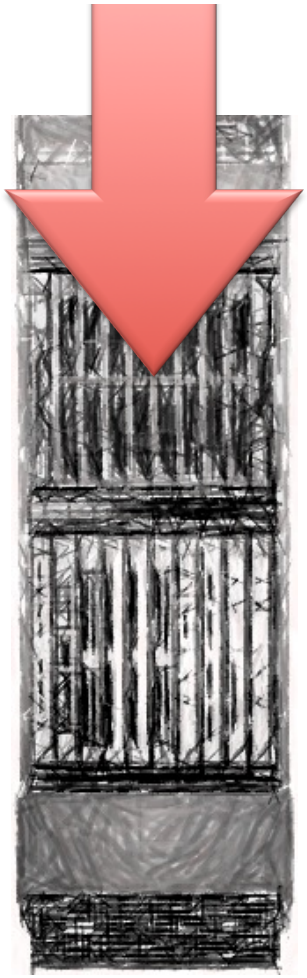
Hundreds of protocols
6,500 RFCs

Tens of millions of lines of code.
Closed, proprietary, outdated.

Billions of gates.
Power hungry and bloated.

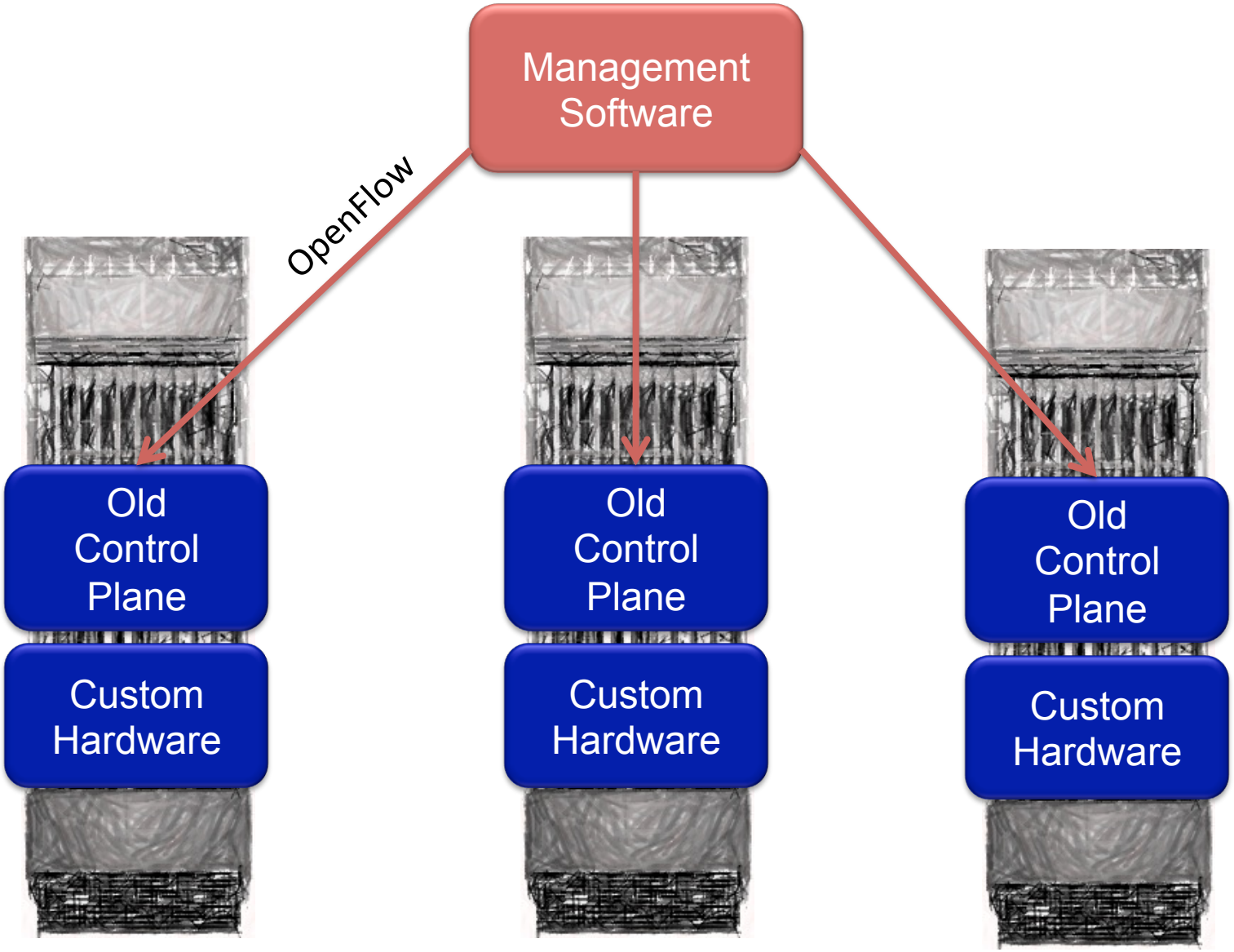
What SDN isn't

Ram in even more lines of code…



"My box now has an OpenFlow interface too!"

What SDN isn't



A network in which the control plane is physically separate from the forwarding plane.

and

A single control plane controls several forwarding devices.

(That's it)

It’s just an idea and a starting point.

Technical Consequences

- Makes crystal clear the distributed systems problem.
- Tells us to solve the distributed systems problem once, rather than multiple times.
- Makes it easier to control diverse switches.

- Particularly if they are protocol independent.

Where is OpenFlow headed?

OpenFlow Goals

1. Protocol-independent forwarding.
2. That can be controlled and repurposed in the field by a remote control plane.

3. [And can be implemented by really fast, low power, switching chips]

Currently OpenFlow v1.x

It is protocol *de*pendent because…

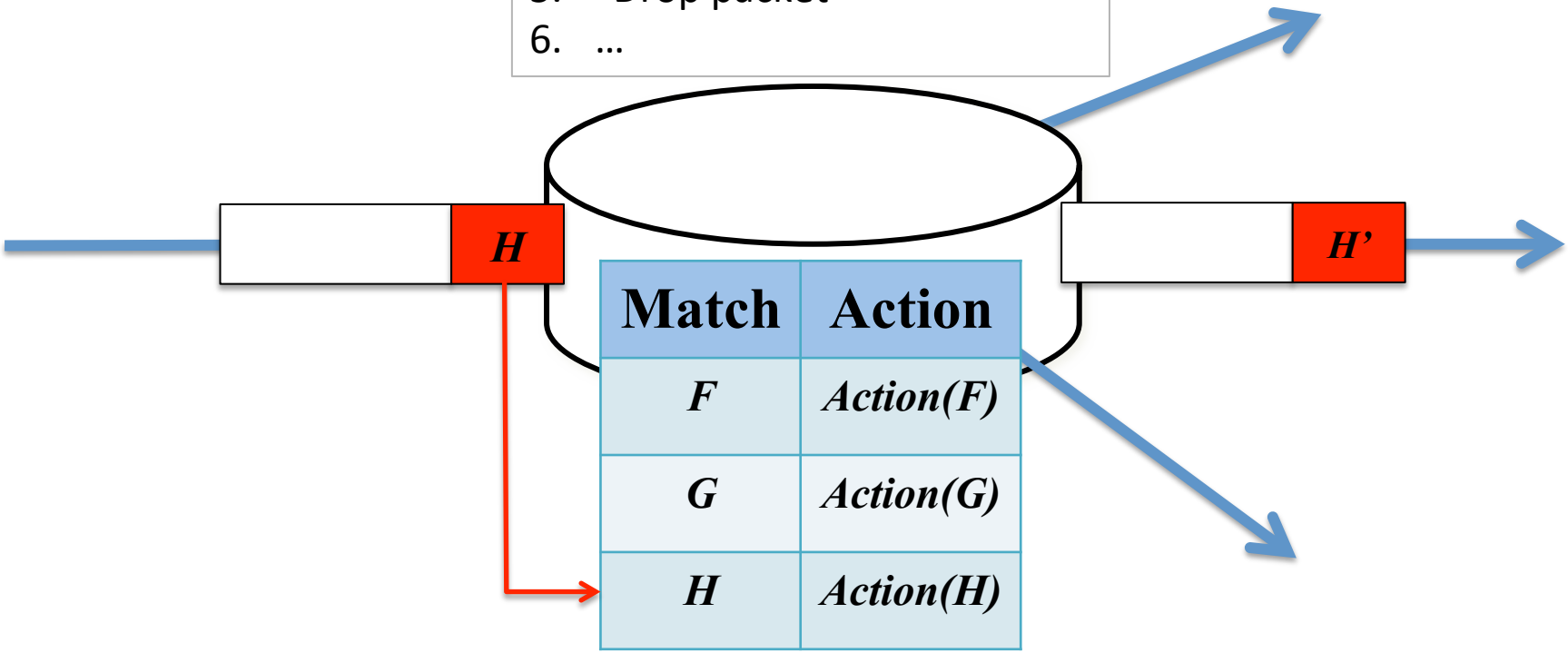
1. Constraints of traditional switching chips.
   - Each table tied to a specific protocol
   - Fixed sequence processing pipeline

2. Maps to existing switching chips.
   - For quick adoption

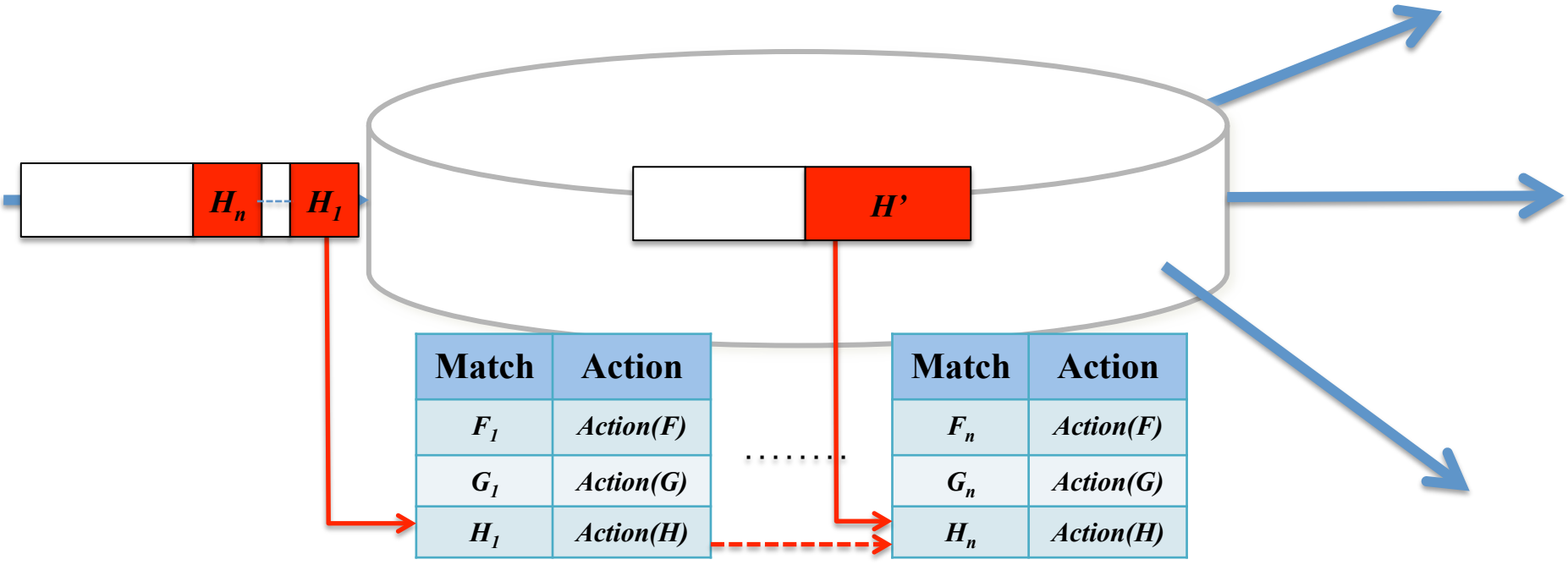
Match-Action Forwarding Abstraction

Action Primitives

1. "Forward to ports 4 & 5"
2. "Push header Y after bit 12"
3. "Pop header bits 8-12"
4. "Decrement bits 13-18"
5. "Drop packet"
6. …



Multiple Table Match-Action



What would an "OpenFlow-optimized" switching chip look like?

Parse any existing header

Parse any custom header

Match:

- Huge protocol independent tables (Millions of entries)
- Many tables in a pipeline (8 or more)
- Tables can be used efficiently

Action:

- Protocol independent
- Instruction primitives for processing headers

Design Exercise

with Texas Instruments

- 64 x 10GE OpenFlow-optimized ASIC
- Industry-standard 28nm design process
- Parse existing + custom packet headers
- 32 stages of “Match + Action”
- Large tables
  - 1M x 40b TCAM
  - 370Mb SRAM (hash table & statistics counters)
- VLIW action processing

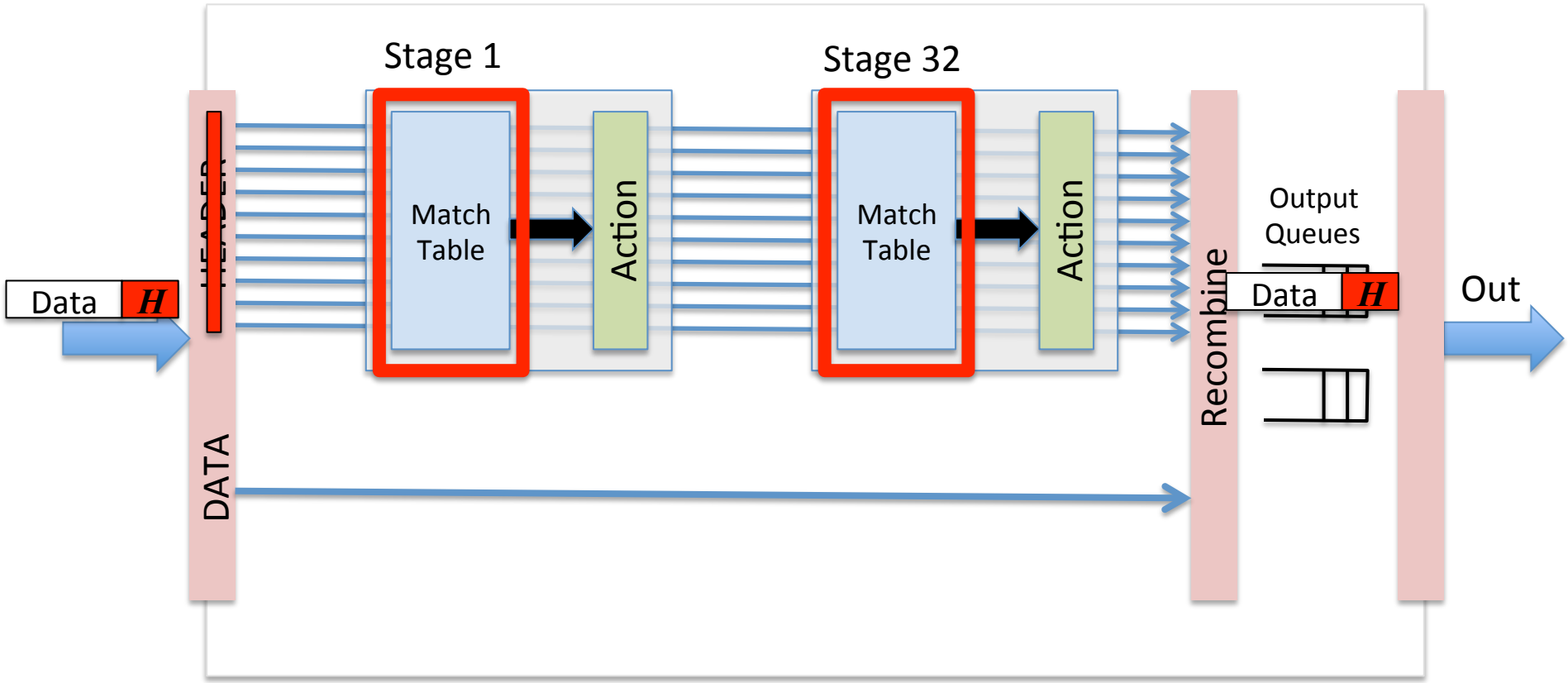
Taken as given

- Fast interface to local CPU
- Usual features: buffers, counters, etc.
- Open interface, not hidden behind NDA.

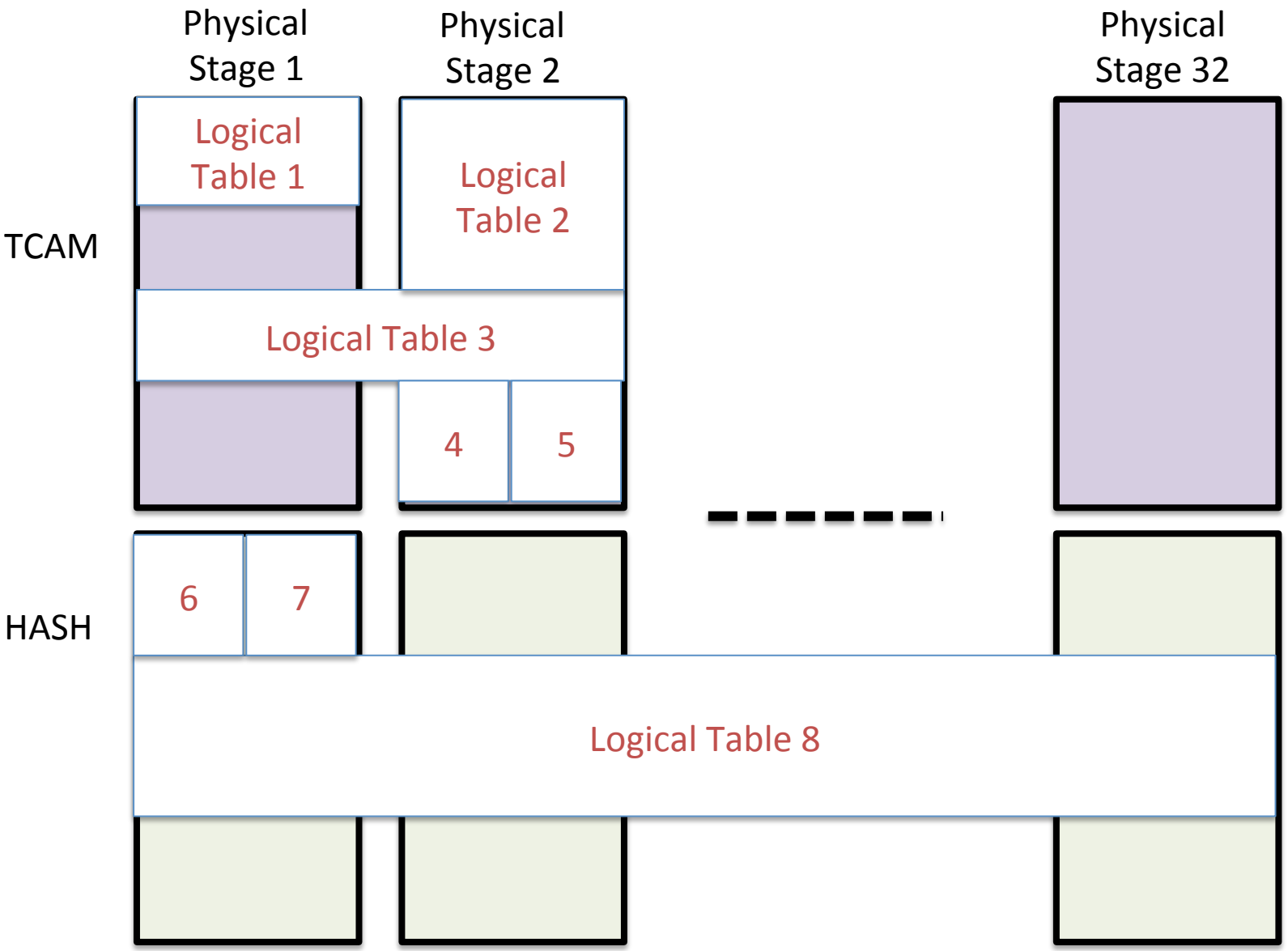
Question

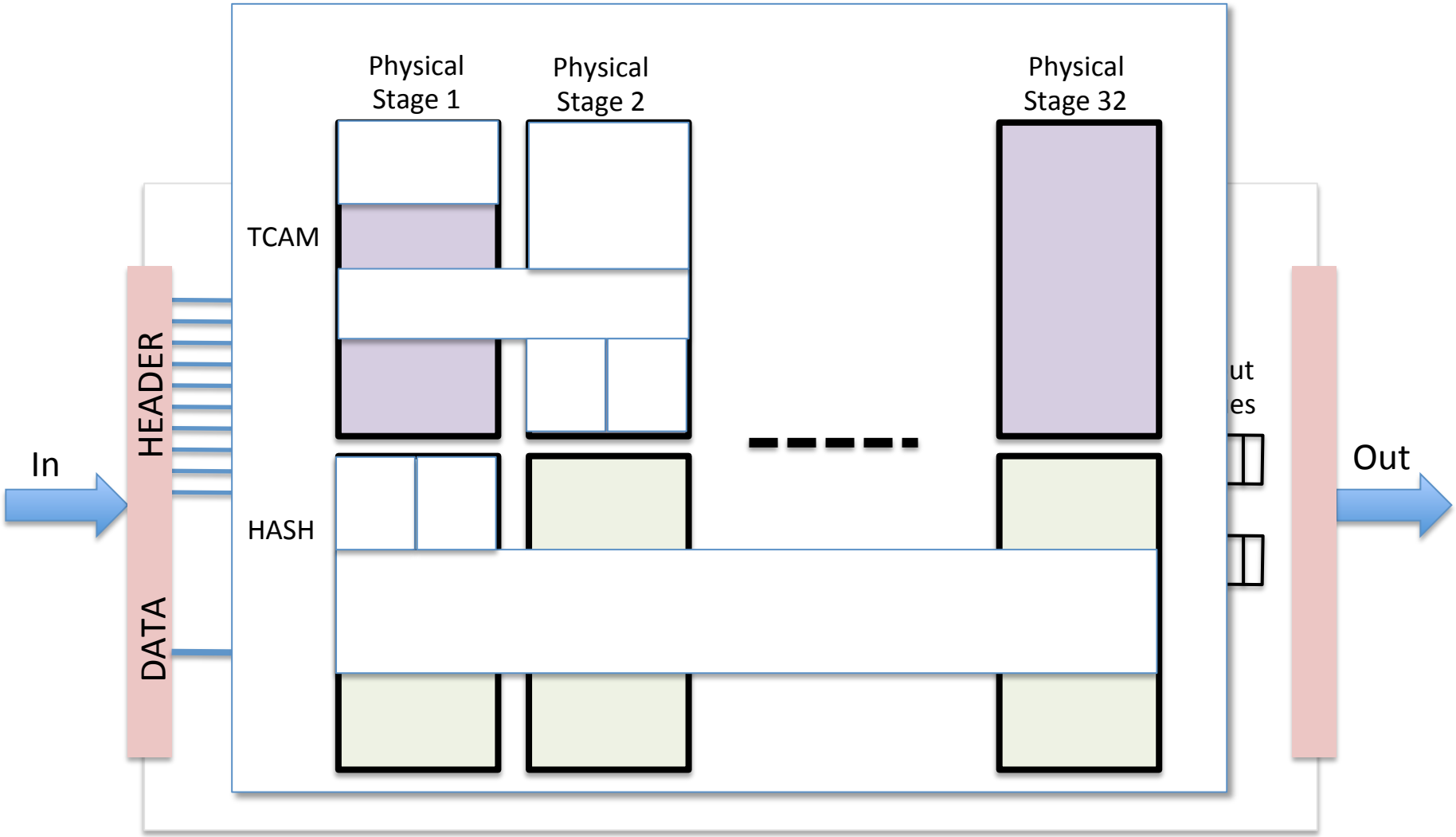
How much *extra* area and power compared to a traditional, fixed function switch chip?

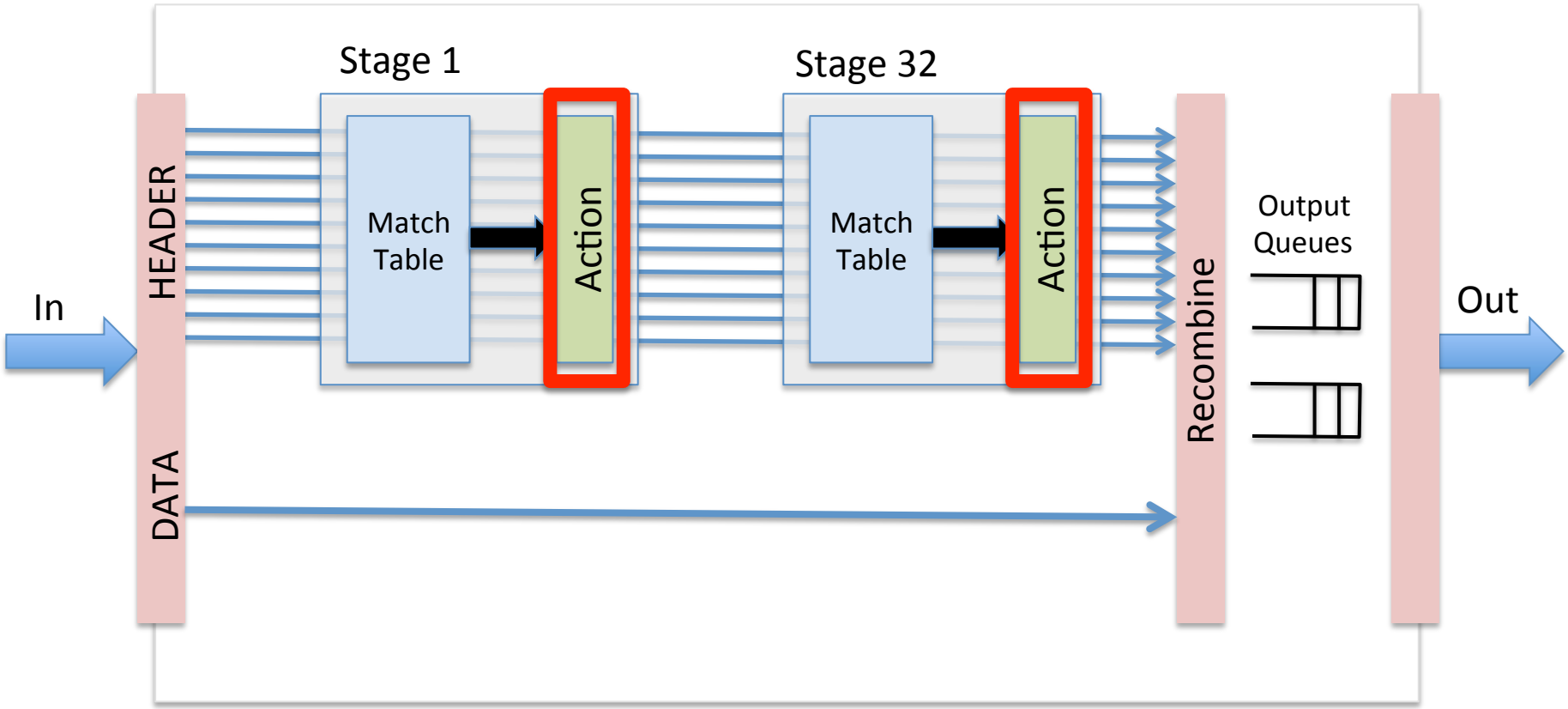
RISC-like architecture



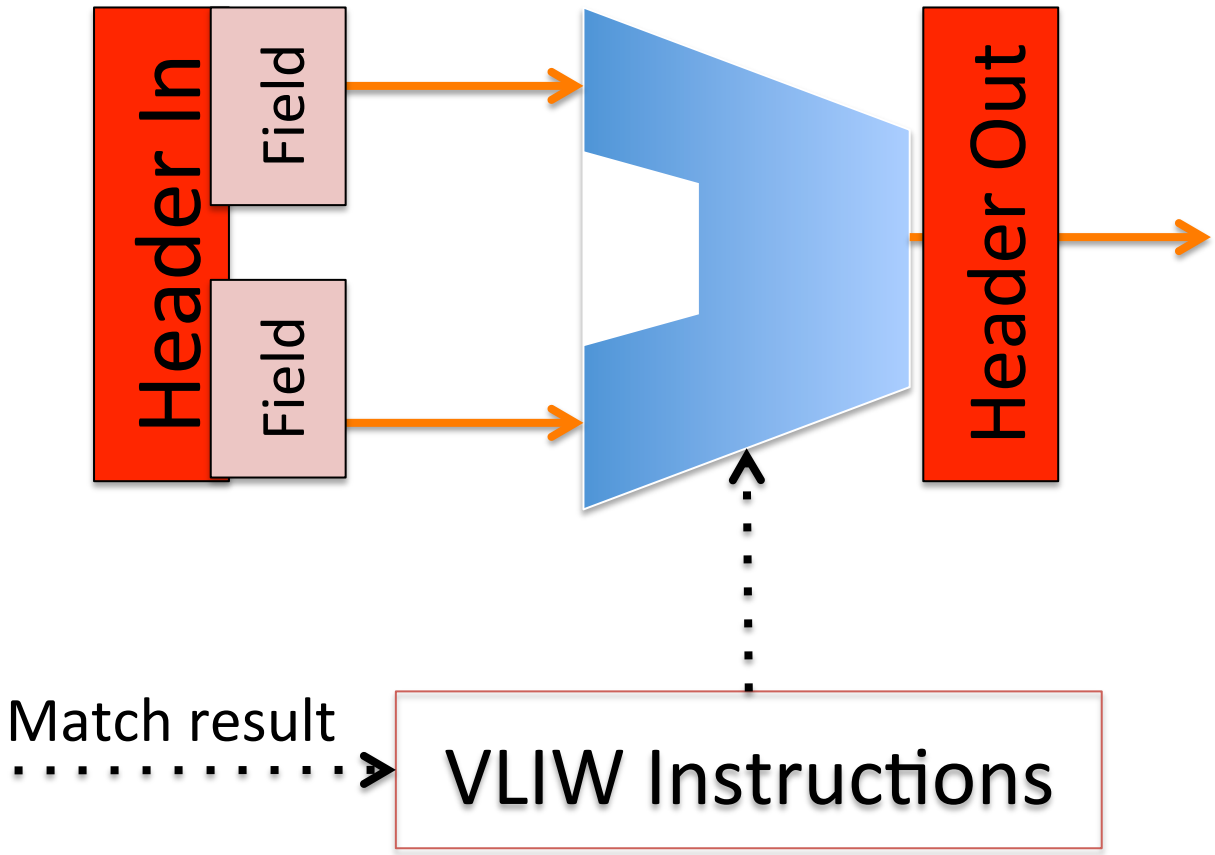
Match Tables



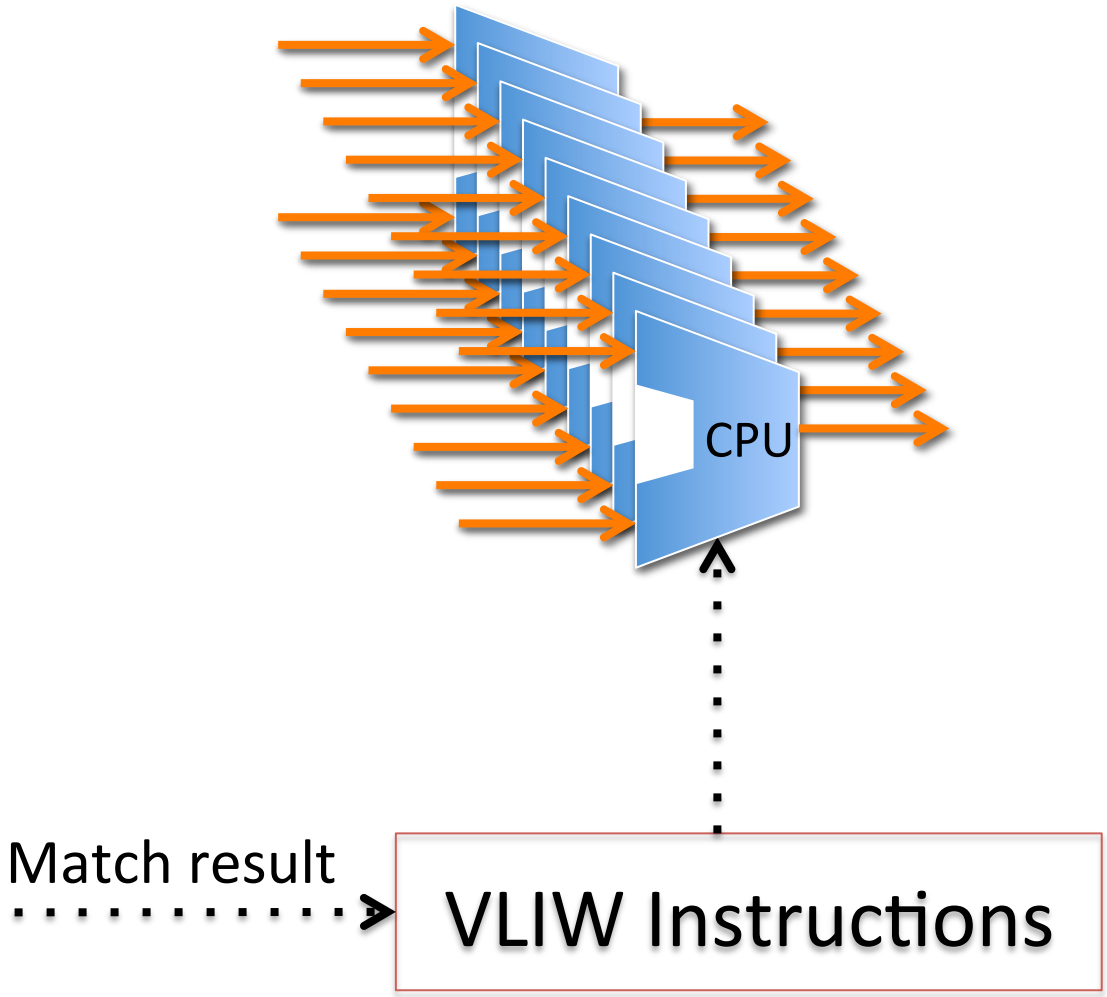

Physical
Stage 1
Physical
Stage 2
Physical
Stage 32
TCAM
HASH
HEADER
DATA
In
Out
ut
es

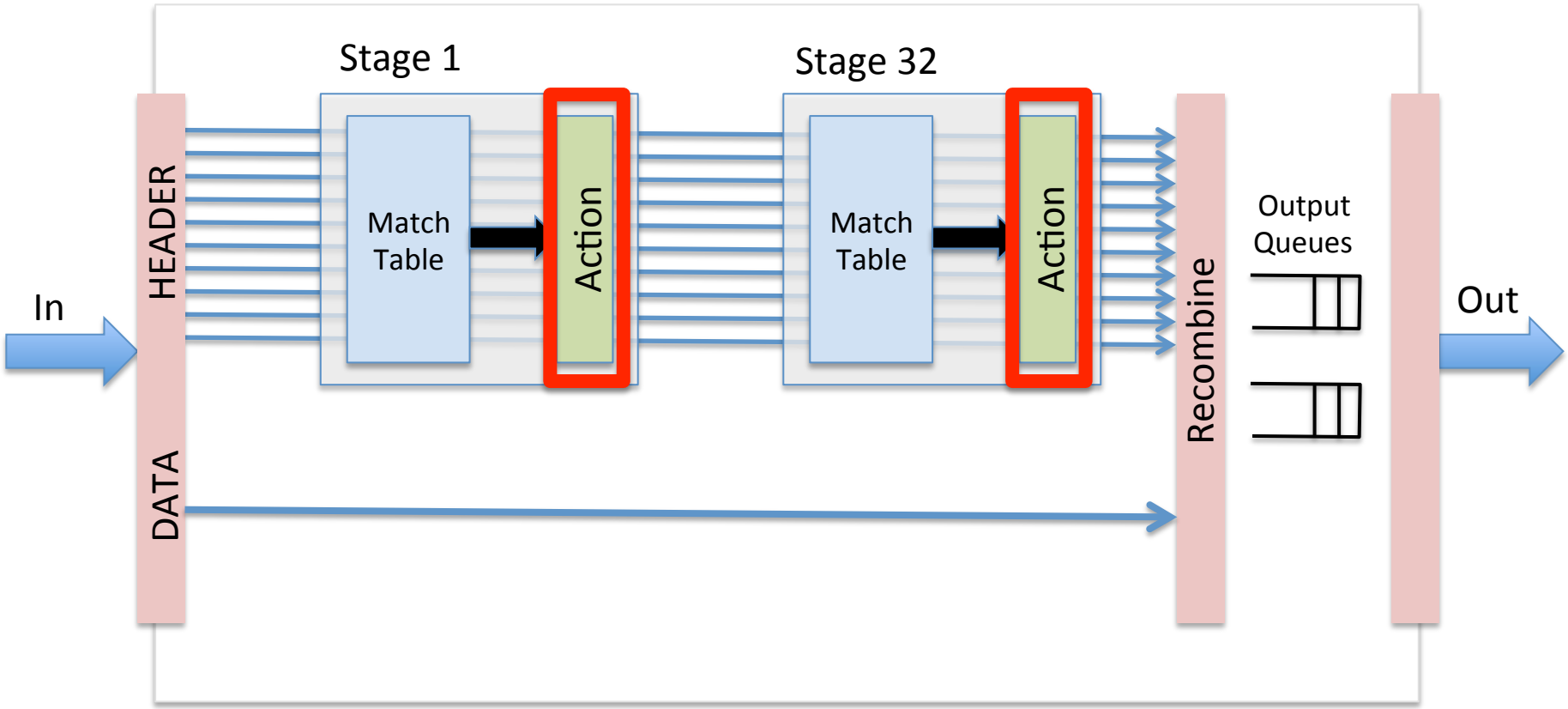

Stage 1
Stage 32
HEADER
DATA
In
Match Table
Action
Match Table
Action
Recombine
Output Queues
Out

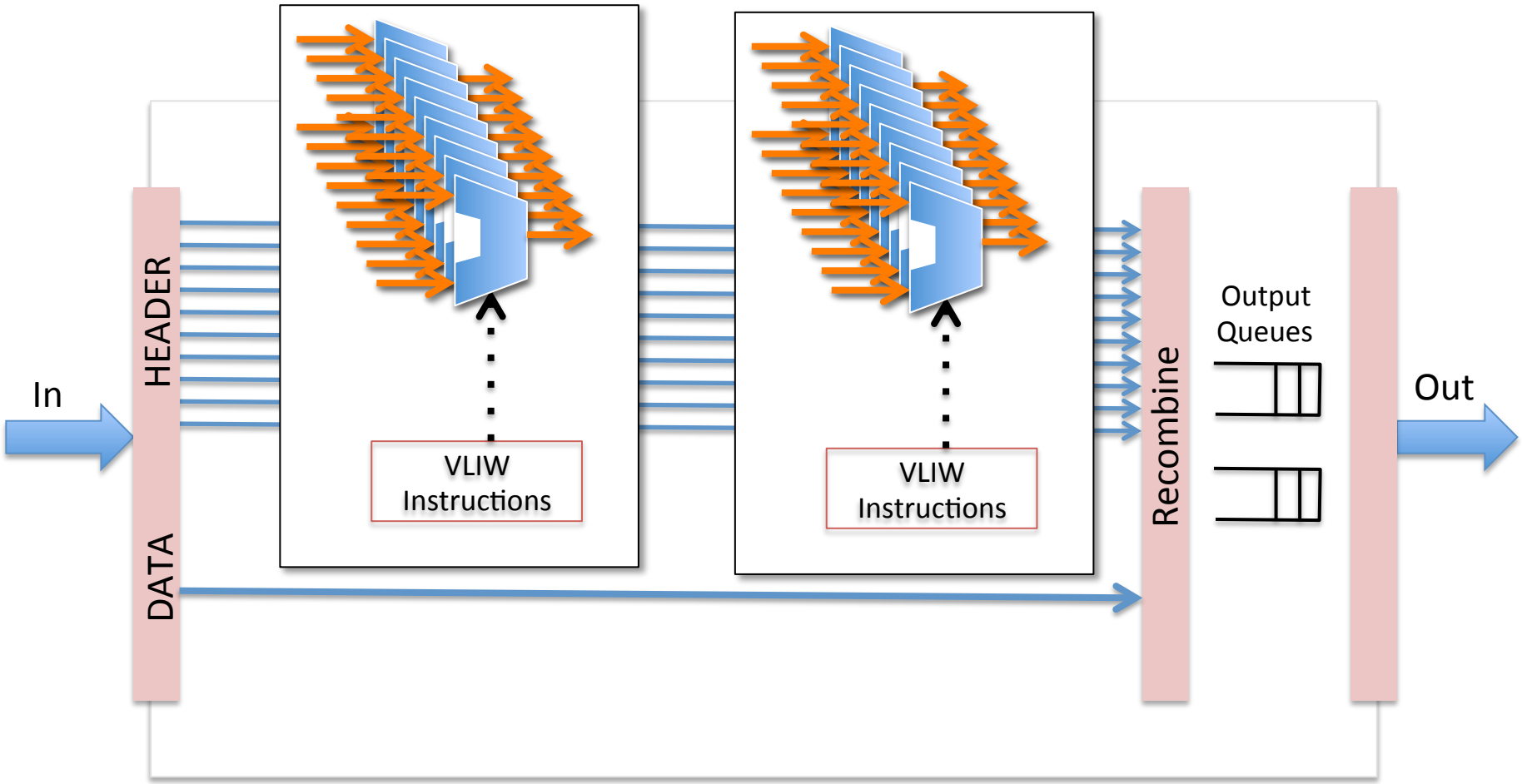
Action Processor



100s of VLIW CPUs




Stage 1
Stage 32
HEADER
DATA
In
Match Table
Action
Match Table
Action
Recombine
Output Queues
Out


In
HEADER
DATA
VLIW
Instructions
VLIW
Instructions
Recombine
Output
Queues
Out

Takeaways

Extreme flexibility

- Custom packet formats
- Repurpose in the field

Large TCAM no longer a problem

- 1M+ entries
- Used very efficiently

All for < 15% extra area

What does this tell us about OpenFlow?

What is possible

Protocol-independent processing

Pipeline: Parse + Multiple “Match-Action” Stages

Field configurable pipeline

But wait….

Won’t OpenFlow
commoditize my business?

Codeword for:

Is it going to erode my enormous profit margins?

Competition based on

1. Swi[CPU speed]acity, power, price
2. Differentiating features in addition to Open[Base instruction set]

Enabling great software.

CPU business is healthy, profitable, innovative.

The switch chip business can be too.

<end>