TCP/IP in hardware using SME

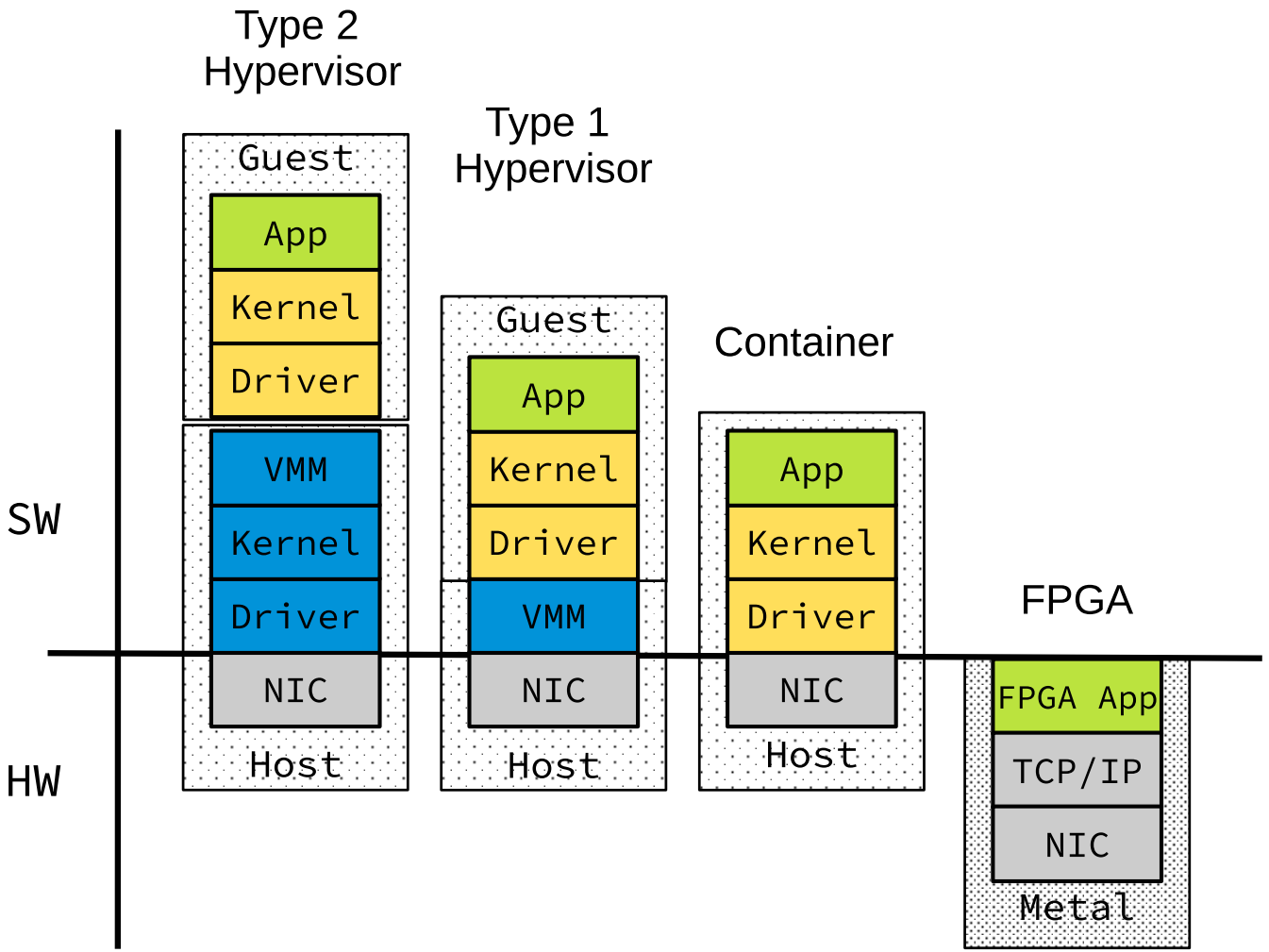
Mark Jan Jacobi & Jan Meznik

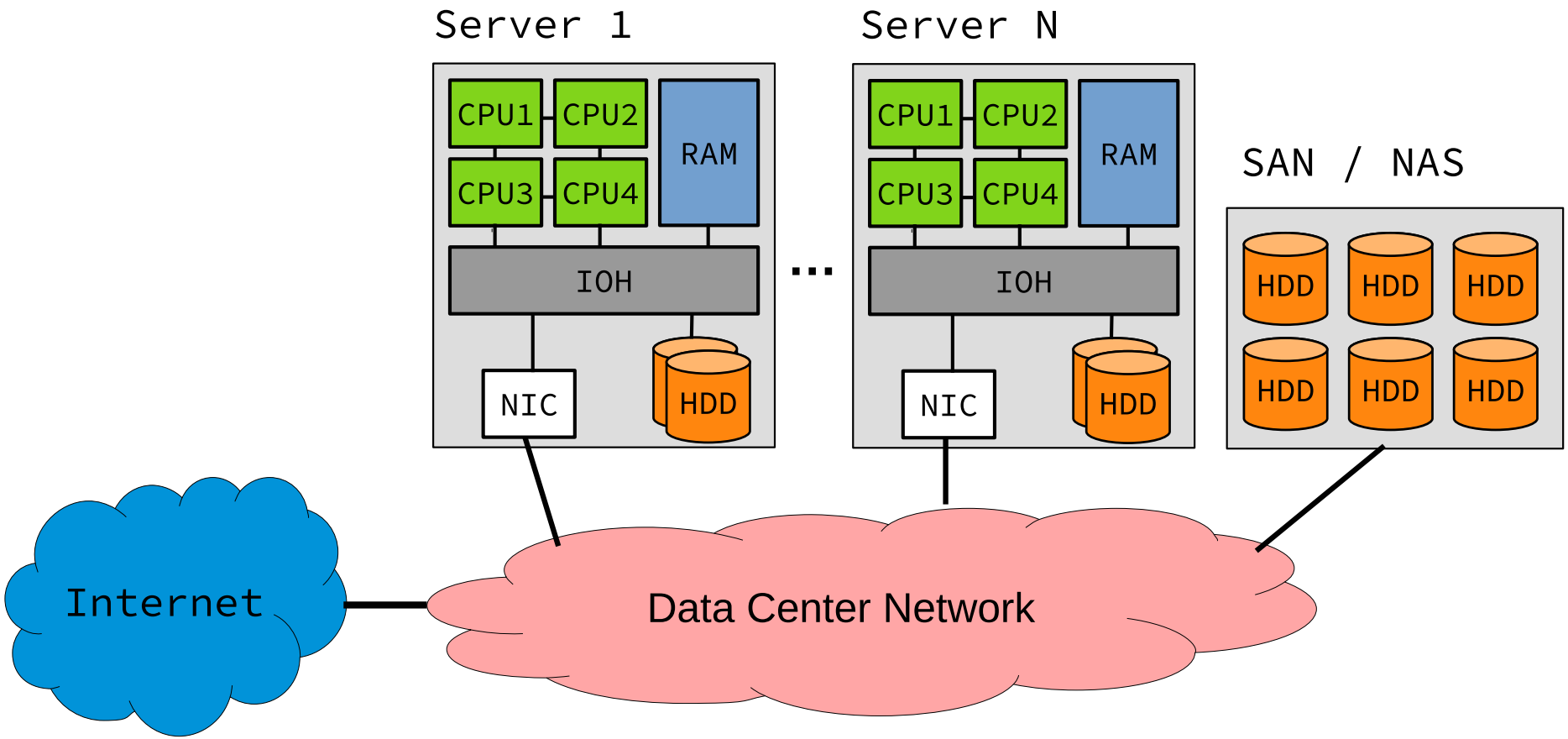
KU

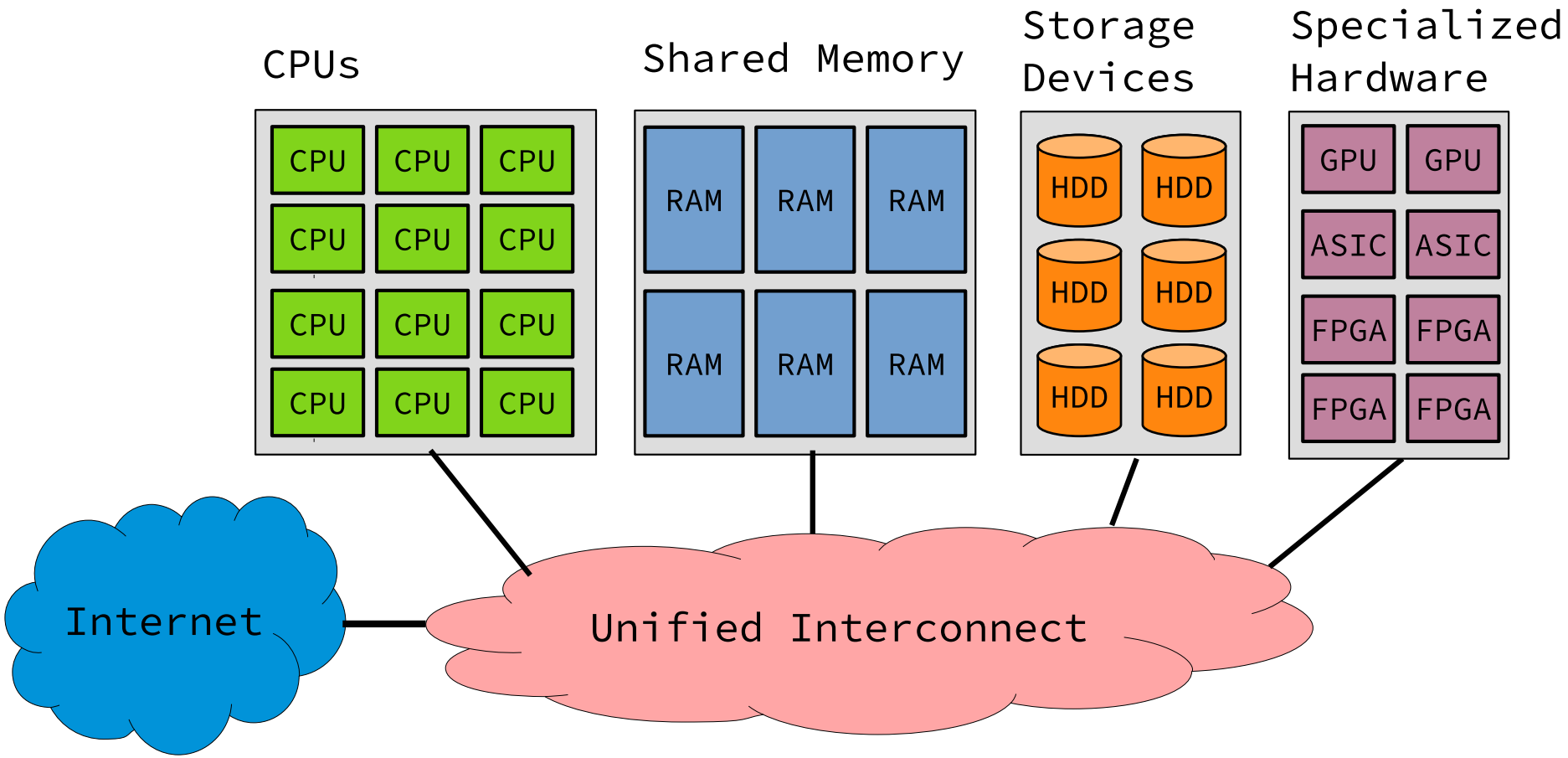
September 10, 2019

Background and Motivation

FPGAs are making their way into data centers to boost the computing power and the overall power efficiency.




Server 1
Server N
CPU1
CPU2
CPU3
CPU4
RAM
IOH
NIC
HDD
...
SAN / NAS
Internet
Data Center Network


CPUs
Shared Memory
Storage Devices
Specialized Hardware
CPU
RAM
HDD
GPU
ASIC
FPGA
Internet
Unified Interconnect

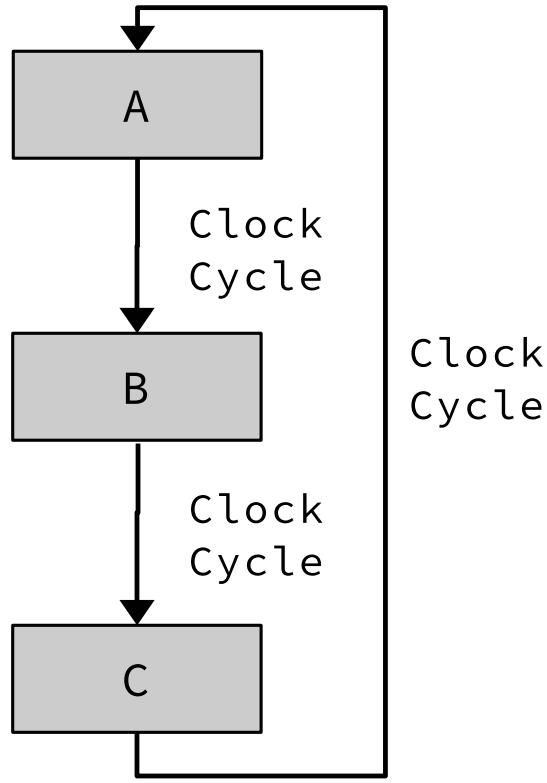
Implementation

- Processes
  - State machines
- Buffers
  - Memory segments
  - Dictionary
- Interface signal control
  - Buffer-Producer
  - Compute-Producer
- Interface control
  - Usage
  - Limitations

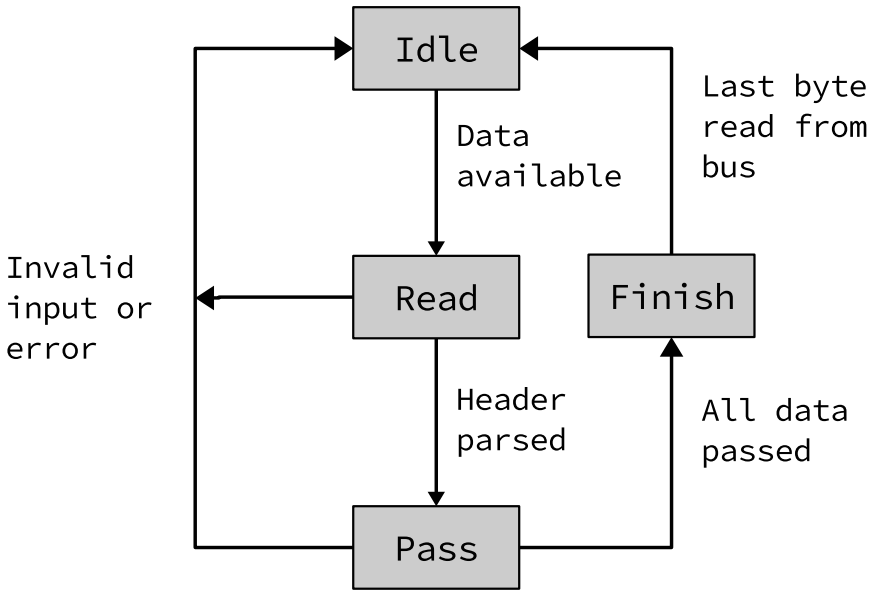
State machines

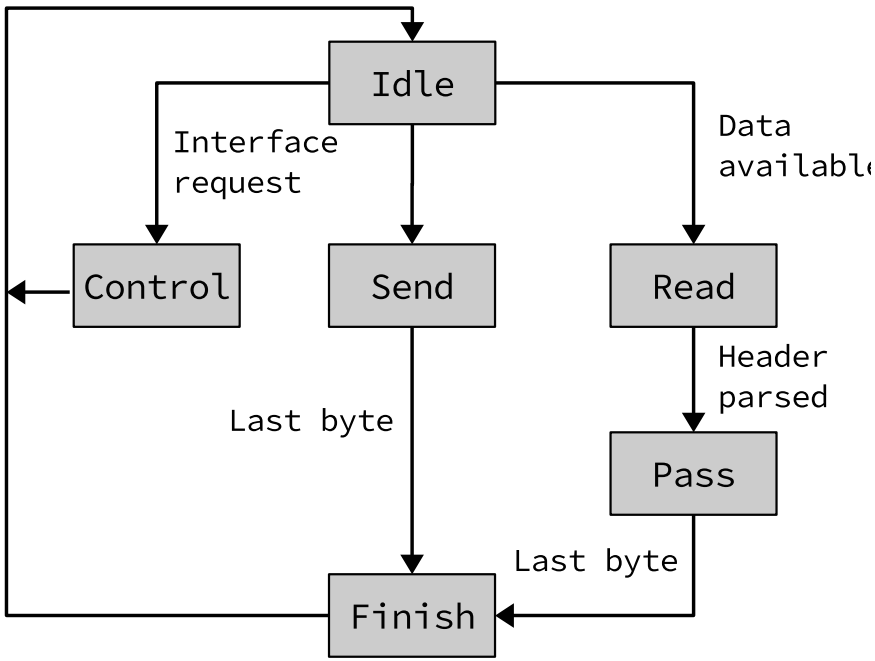
```
1  public class SomeProcess :
     ↪ StateProcess
2  {
3    private override async
       ↪ Task
       ↪ OnTickAsync()
4    {
5      a();
6      await ClockAsync();
7      b();
8      await ClockAsync();
9      c();
10     await ClockAsync();
11   }
12 }
```

```
1  public class SomeProcess :
     ↪ SimpleProcess
2  {
3  // Initial state
4  state = A;
5
6  protected override void
     ↪ OnTick()
7  {
8    switch(state) {
9      case A:
10       a();
11       state = B;
12     case B:
13       b();
14       state = C;
15     case C:
16       c();
17       state = A;
18   }
19 }
```

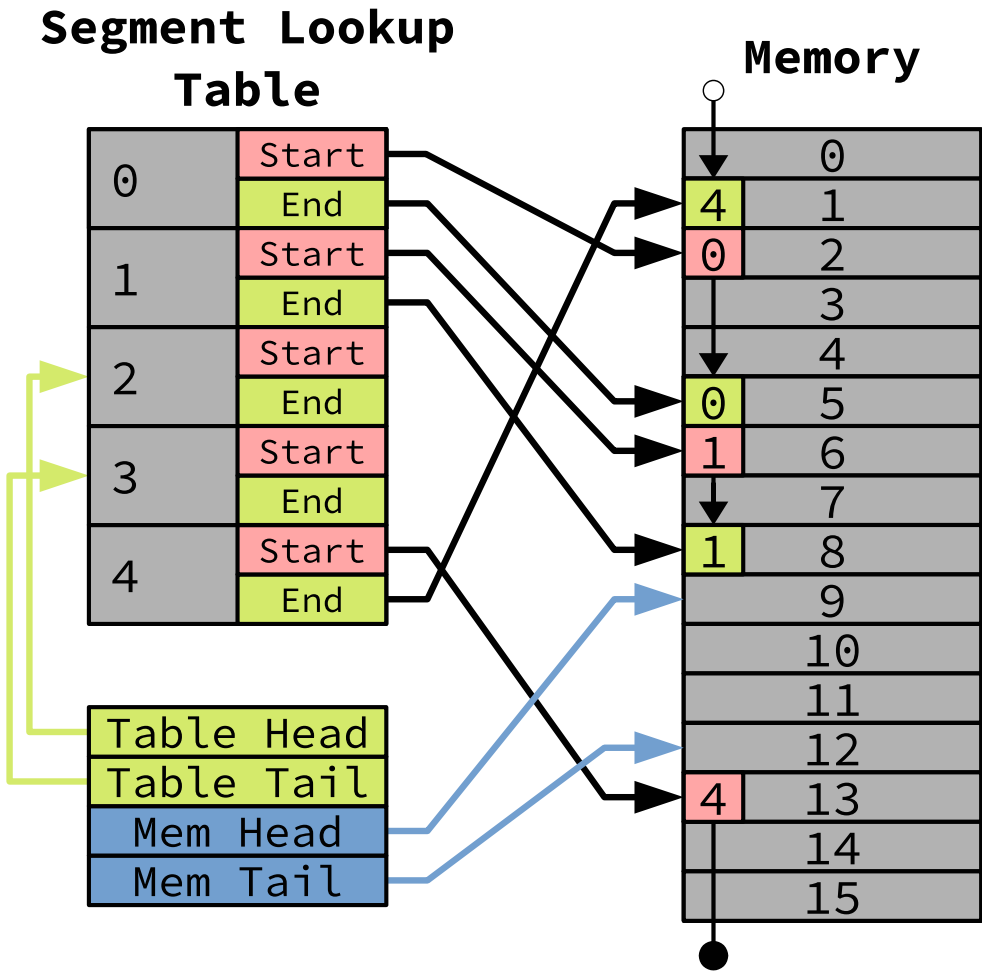


Examples

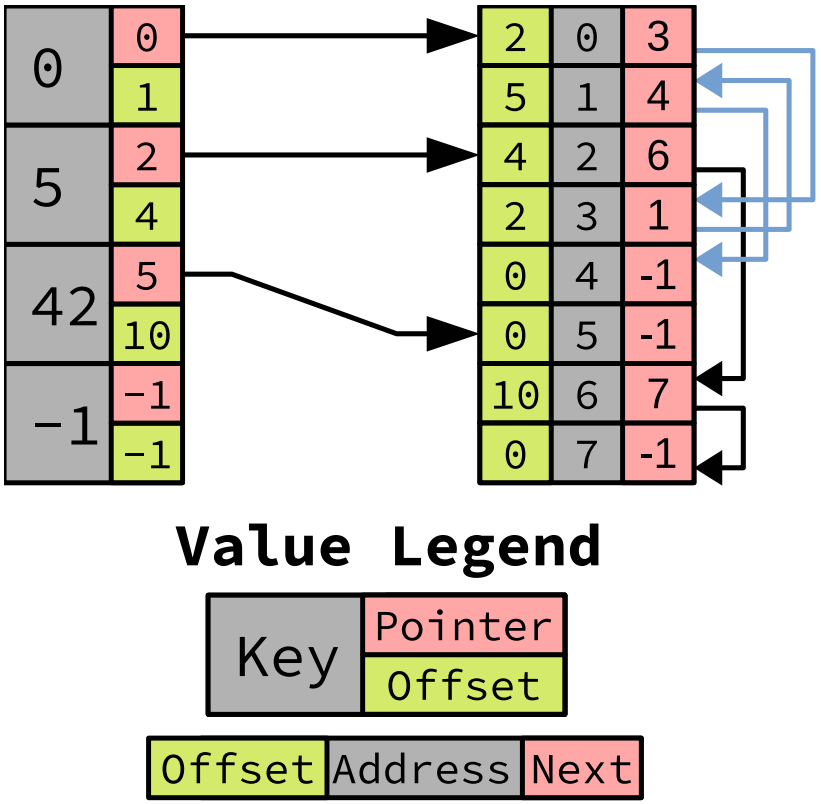

Idle
Data available
Last byte read from bus
Invalid input or error
Read
Finish
Header parsed
All data passed
Pass


Idle
Interface request
Data available
Control
Send
Read
Header parsed
Last byte
Pass
Last byte
Finish

Memory segments

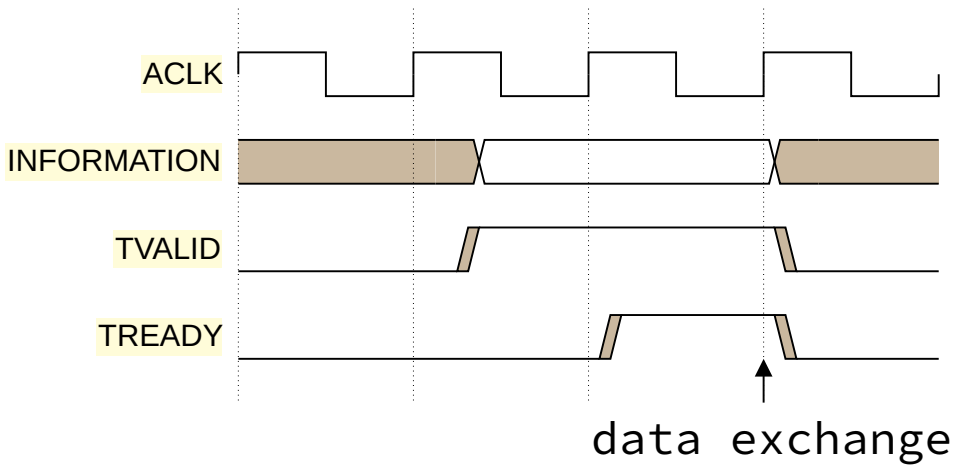


Memory dictionary



Inspired by AXI4

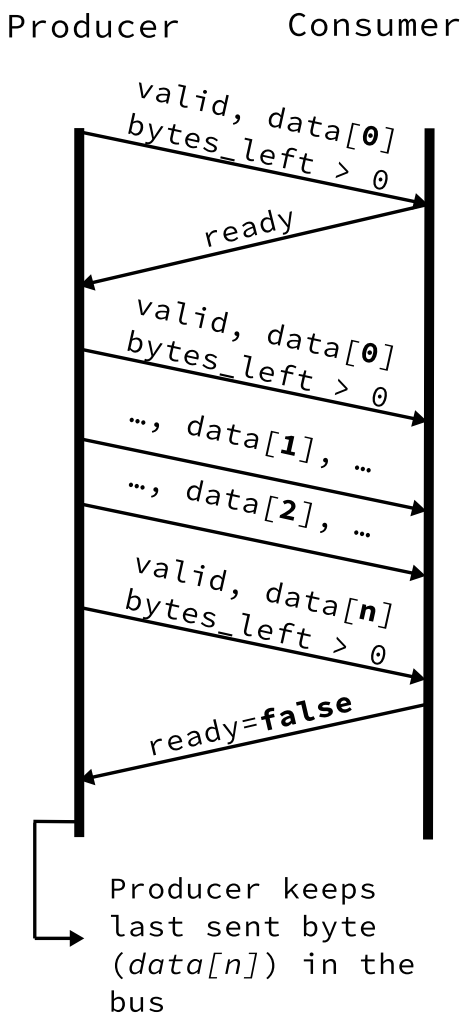
- Single clock offset when sending data.
- Indicate end of stream with `bytes_left`.



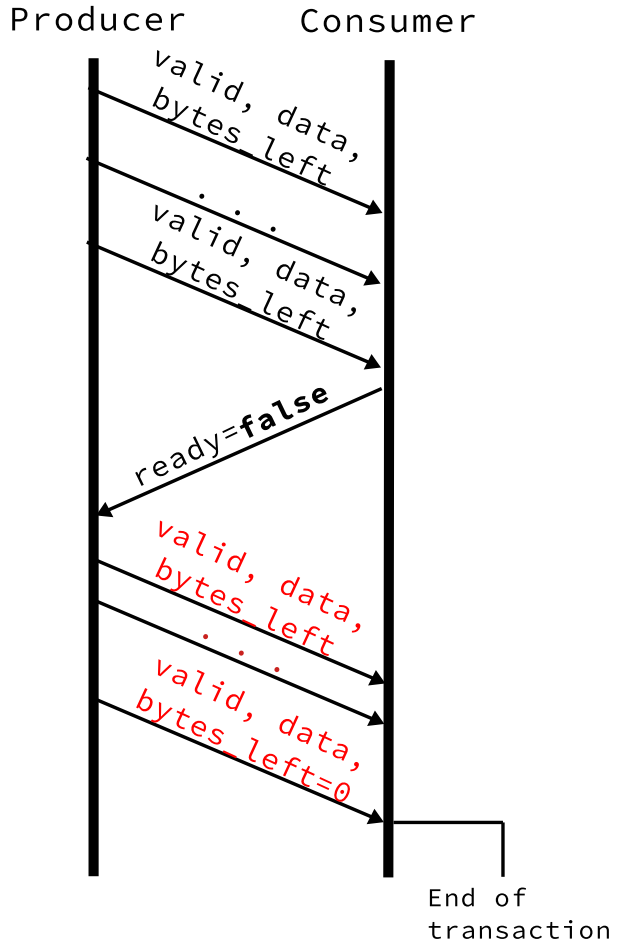
Implementation

Interface signal protocol

Buffer-Producer (BP)



Compute-Producer (CP)



The interface structures

```
enum InterfaceFunction : byte
{
    INVALID = 0,
    // BIND = 1,
    LISTEN = 2,
    CONNECT = 3,
    ACCEPT = 4,
    CLOSE = 7,
    // ...
    OPEN = 255,
}

struct InterfaceData
{
    public int socket;
    public uint ip;
    public byte protocol;
    public ushort port;
}
```

```
interface InterfaceBus : IBus
{
    bool valid;
    byte interface_function;
    InterfaceData request;
}

interface InterfaceControlBus : IBus
{
    bool valid;

    byte exit_status;
    byte interface_function;
    InterfaceData request;
    InterfaceData response;
}
```

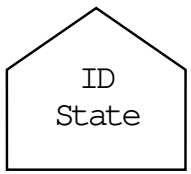
Limitations

- One request at a time.
- Arbitrary delay between request and response.

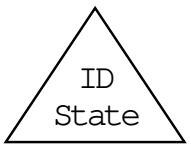
Evaluation

- Setup
  - Graph file simulator
- Test
- Validation
  - Latency
  - Outgoing packet validation
  - Internet Protocol Suite compliancy as per RFC 1122

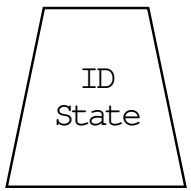
Graph simulation node types



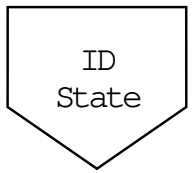
Data In



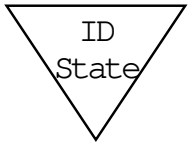
Send



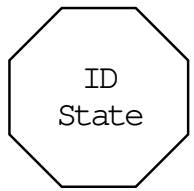
Command



Data Out



Receive



Wait

Questions

? Some random citation so it does not complain[Andrew S Tanenbaum(2013)]

Bibliography


Todd Austin Andrew S Tanenbaum.
Structured computer organization.
Pearson, Boston, 2013.
ISBN 978-0-273-76924-8.

end

end