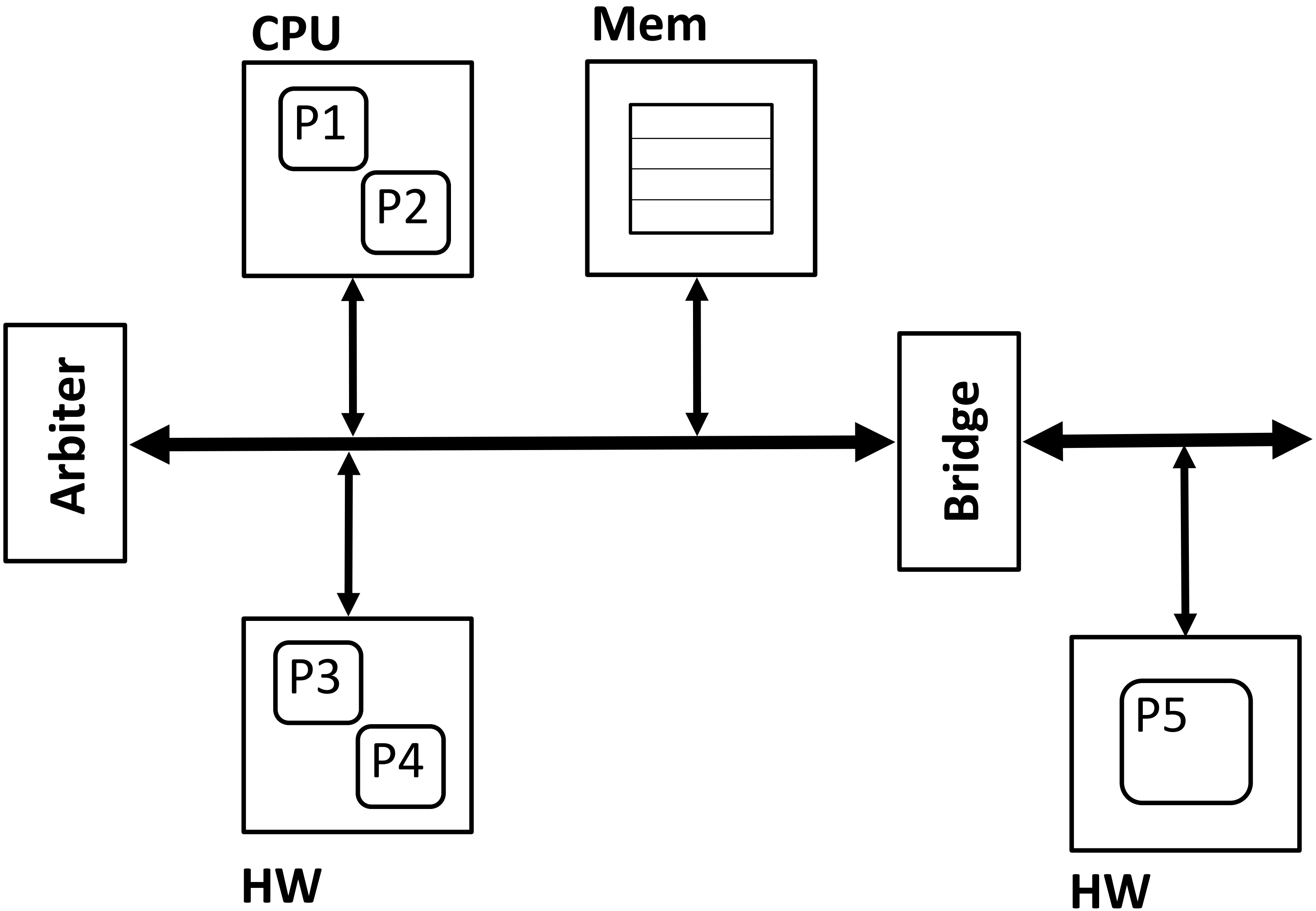
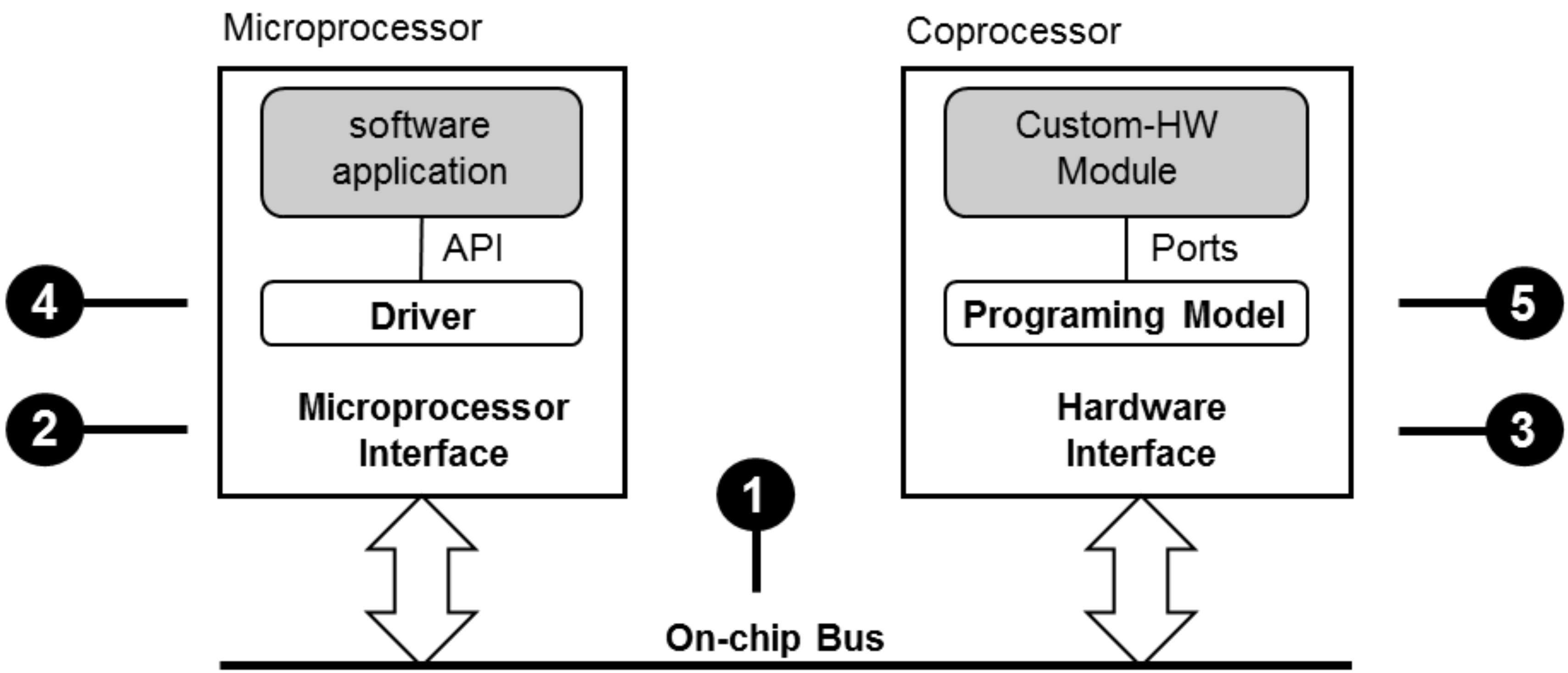
System-on-Chip Design
HW/SW Interfaces and Communications

Hao Zheng
Comp Sci & Eng
U of South Florida

System Structural Model

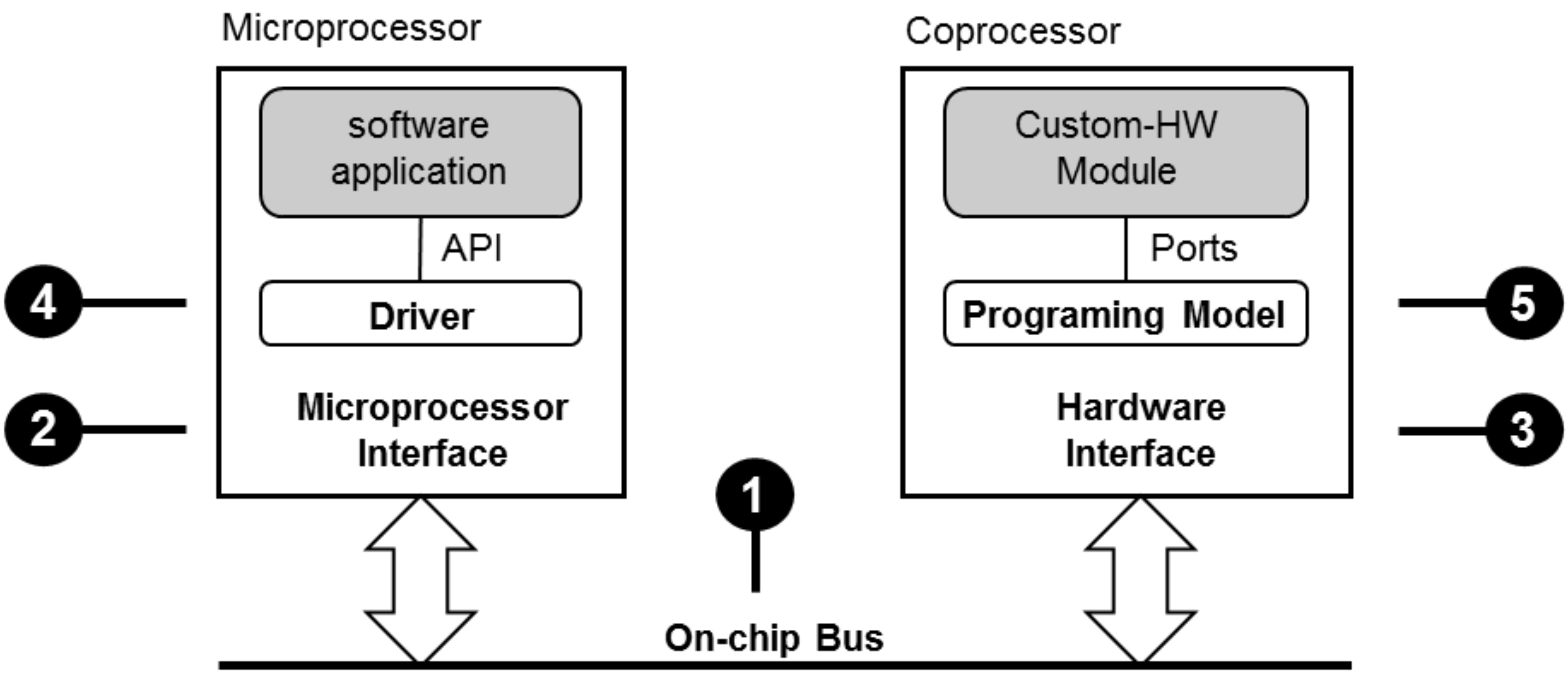


Basic Elements of HW/SW Interfaces



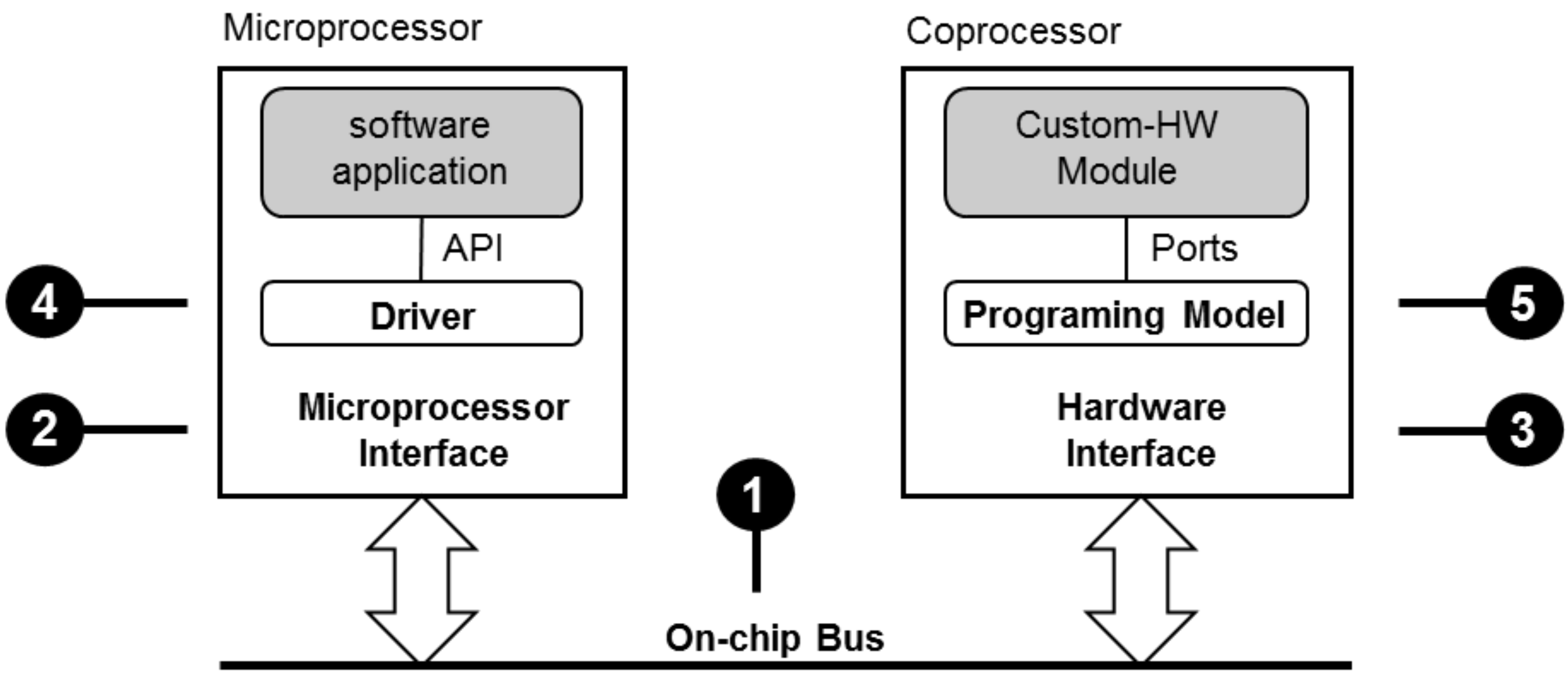
1. On-chip communication fabrics, ex. buses

Basic Elements of HW/SW Interfaces



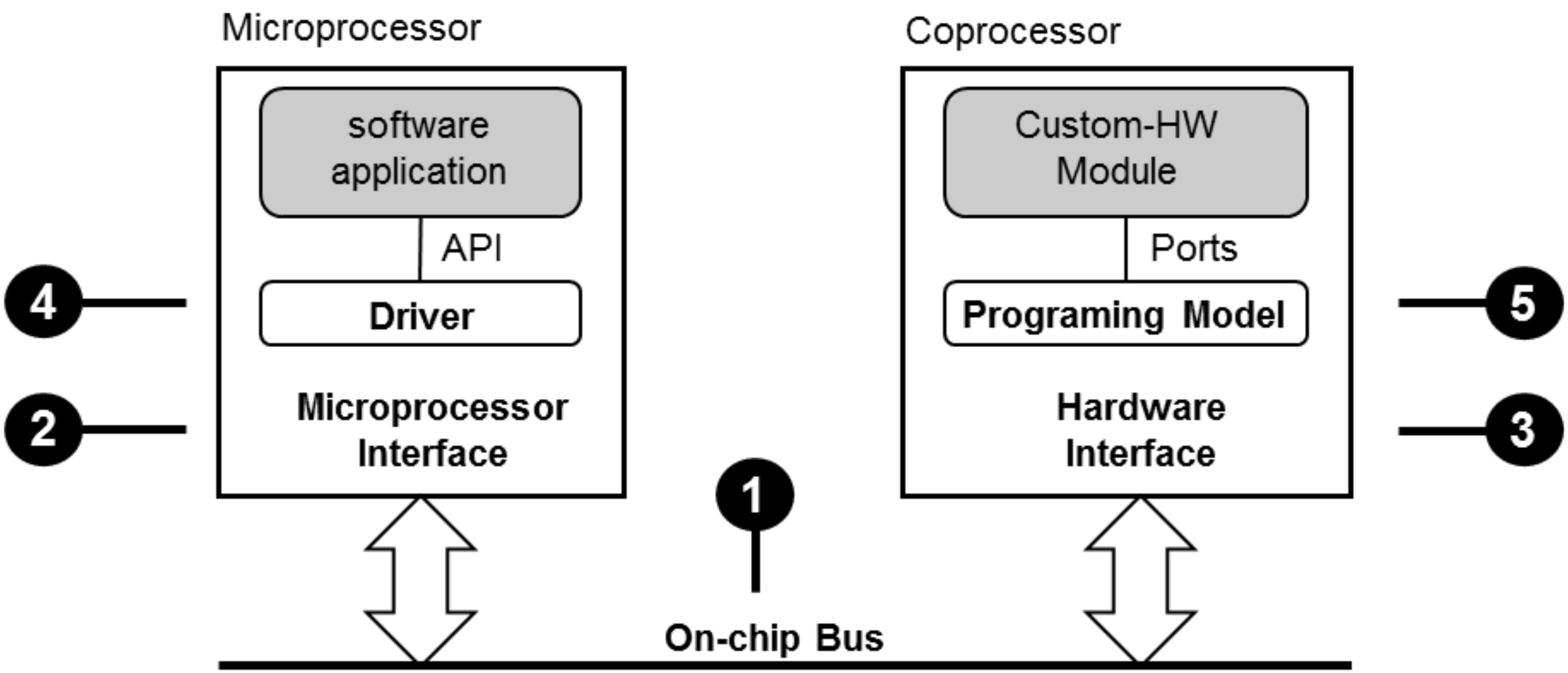
2. CPU interface for SW to communicate with custom HW.

Basic Elements of HW/SW Interfaces



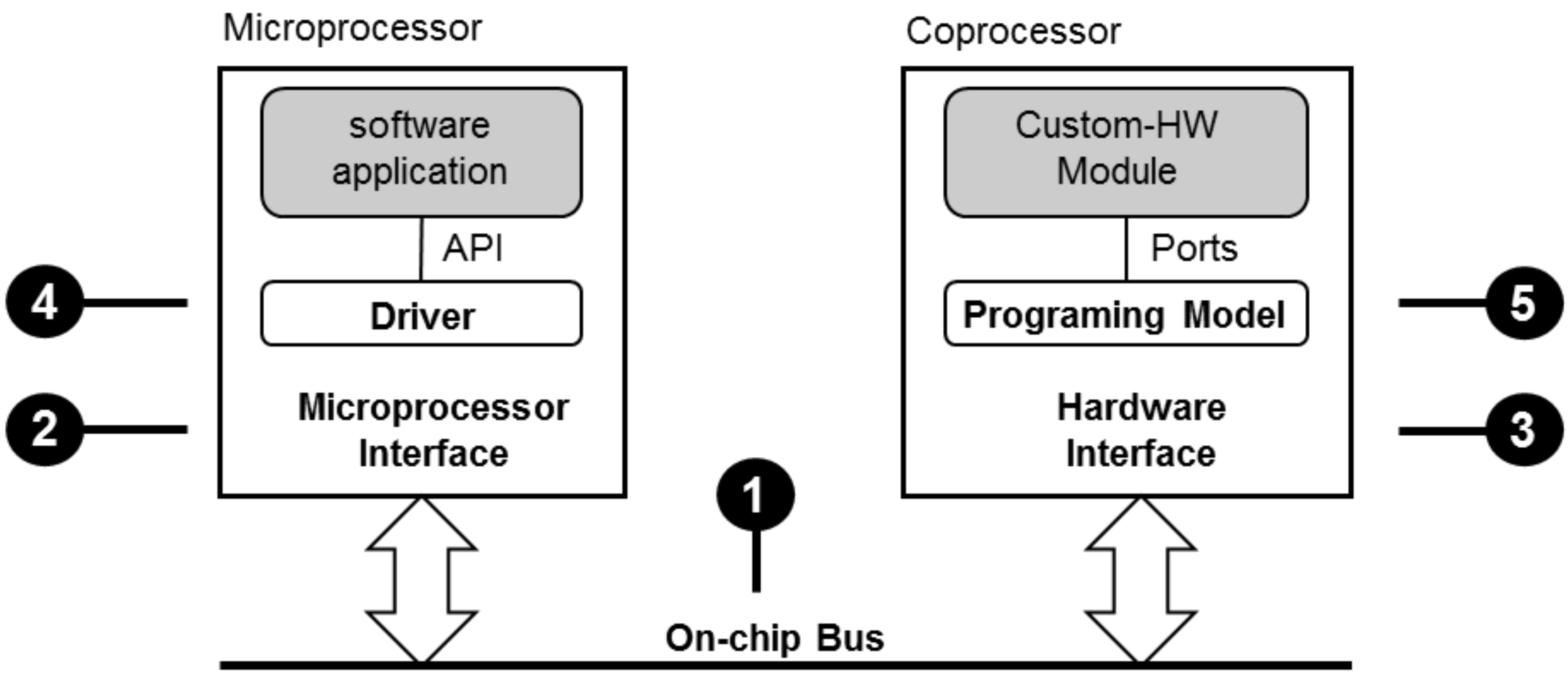
3. HW interface for custom HW to communicate with CPU.

Basic Elements of HW/SW Interfaces



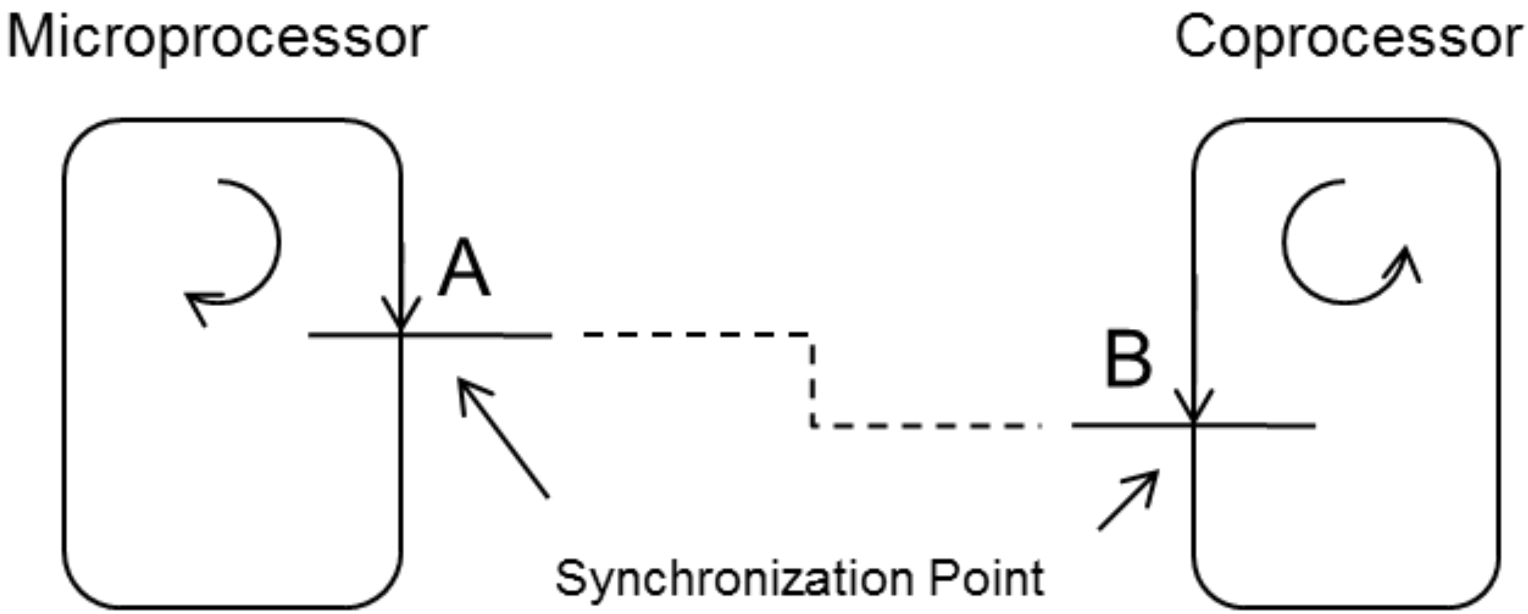
4. SW driver converts SW IO operations to operations supported by CPU interface.

Basic Elements of HW/SW Interfaces



5. Programming model where SW running CPU uses to control custom HW module.

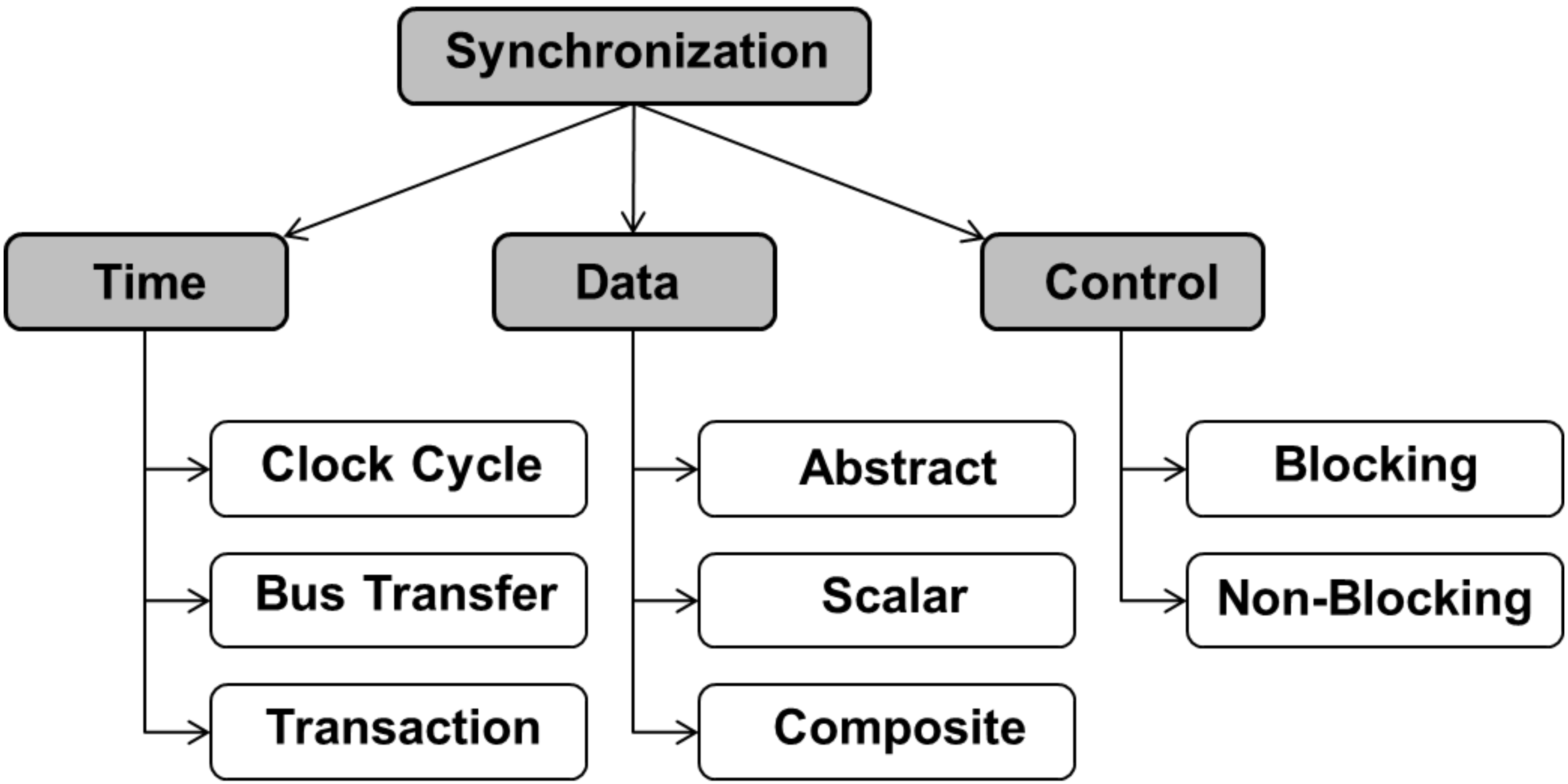
Synchronization Schemes



Synchronization is necessary for effective communications, i.e. data transferred between CPU and HW correctly.

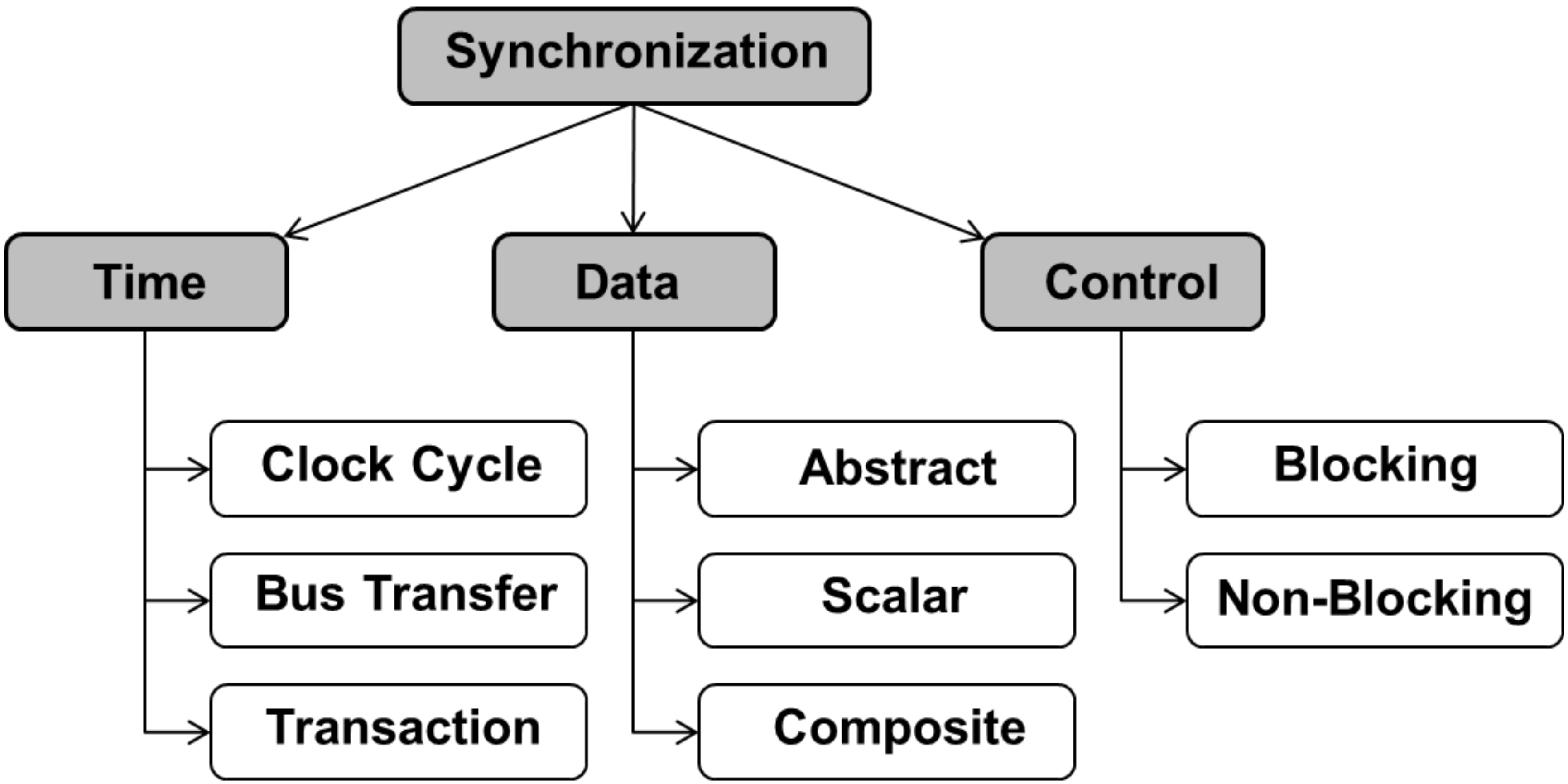
Synchronization is part of interface implementation.

Synchronization Schemes



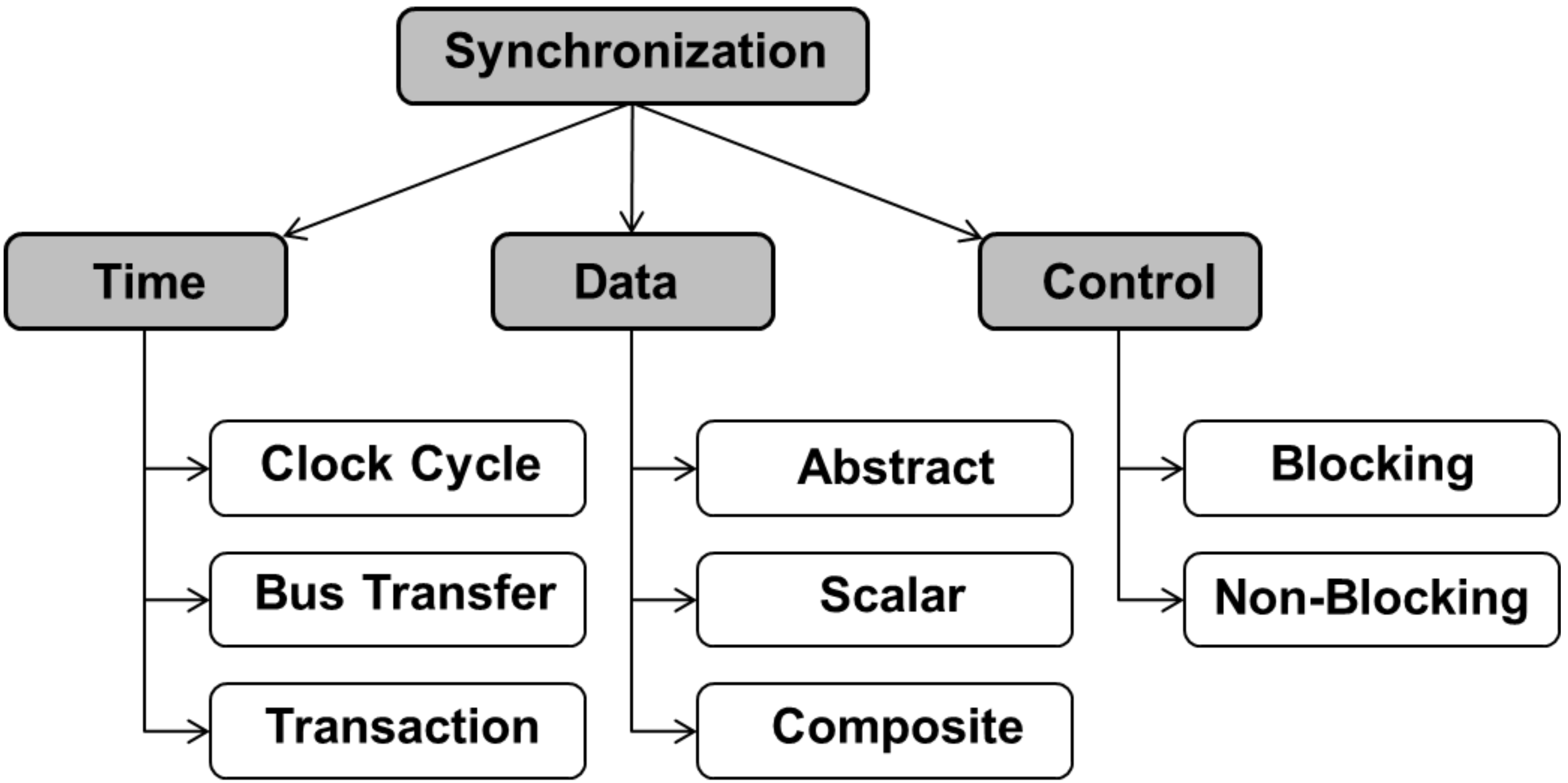
Time: how synchronization is defined over time.

Synchronization Schemes



Data: how data is represented in synchronization.

Synchronization Schemes



Control: how synchronization is implemented locally in individual modules..

Semaphores

- Used to control of accesses to shared resource.
- Two ops on semaphore S:
  - P(S): acquire S.
  - V(S): release S.
- How can we ensure an order between thread 1 & 2?

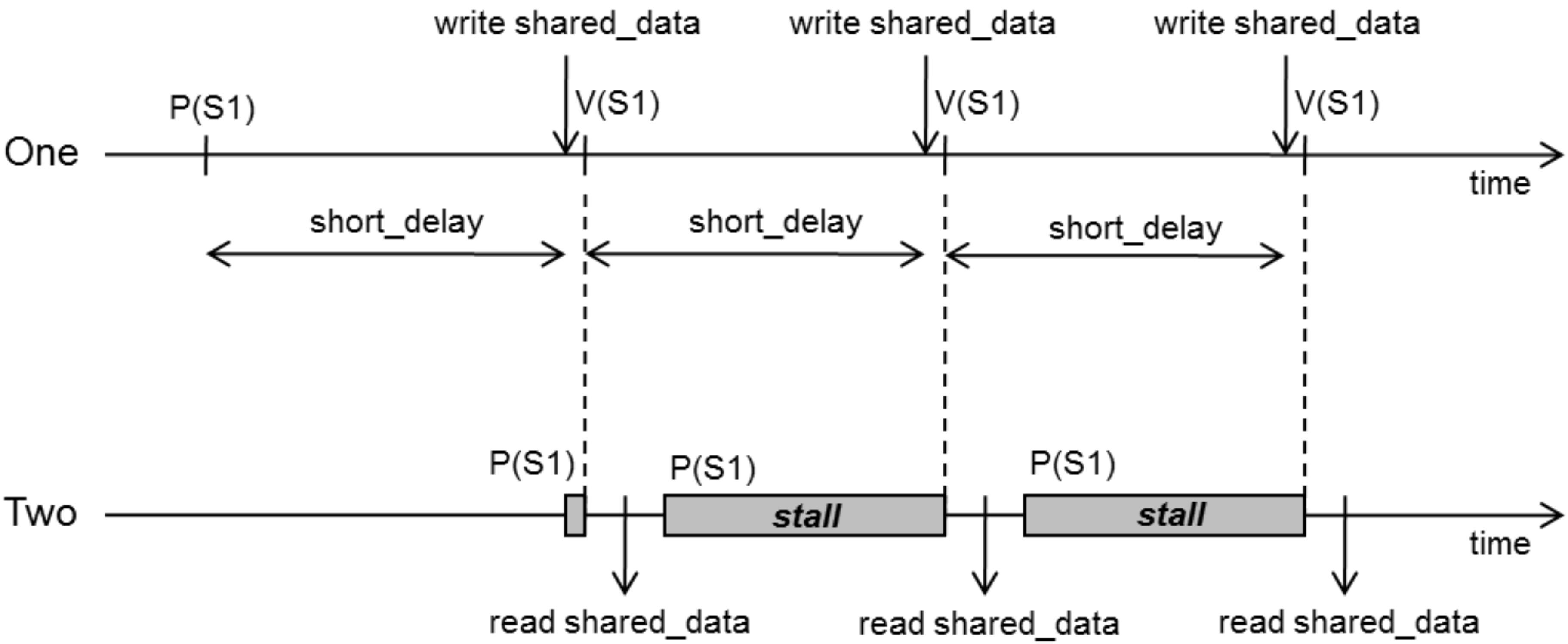
thread 1:	thread 2:
…	…
P(S);	P(S);
x++;	x = x – 2;
V(s);	V(s);
…	…

Semaphores

```
int shared_data;
semaphore S;
```

```
entity one {
  P(S);
  while (1) {
    short_delay();
    shared_data = ...;
    V(S);
}}
```

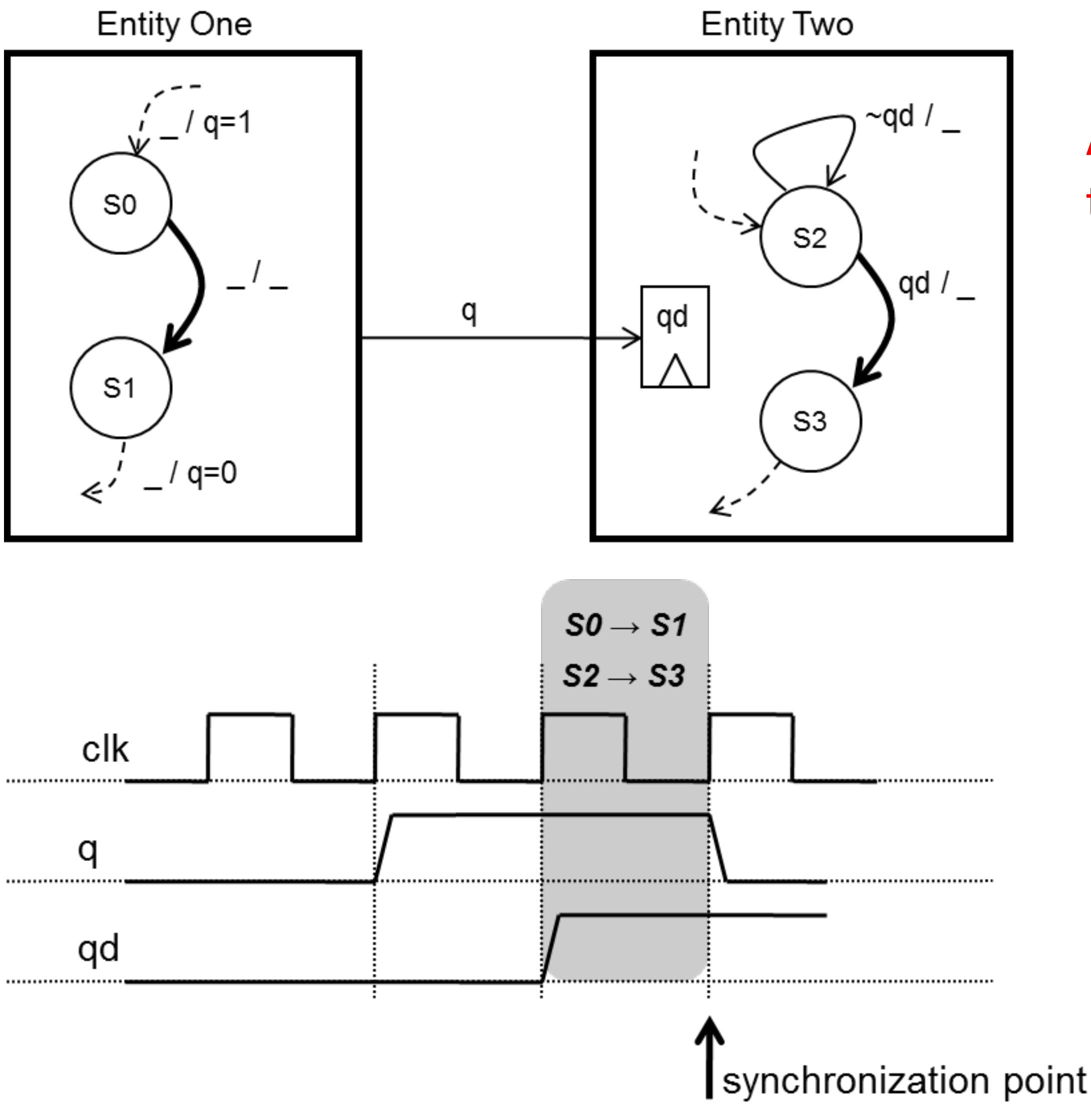
```
entity two {
  short_delay();
  while (1) {
    P(S);
    rd = shared_data;
}}
```



Semaphores

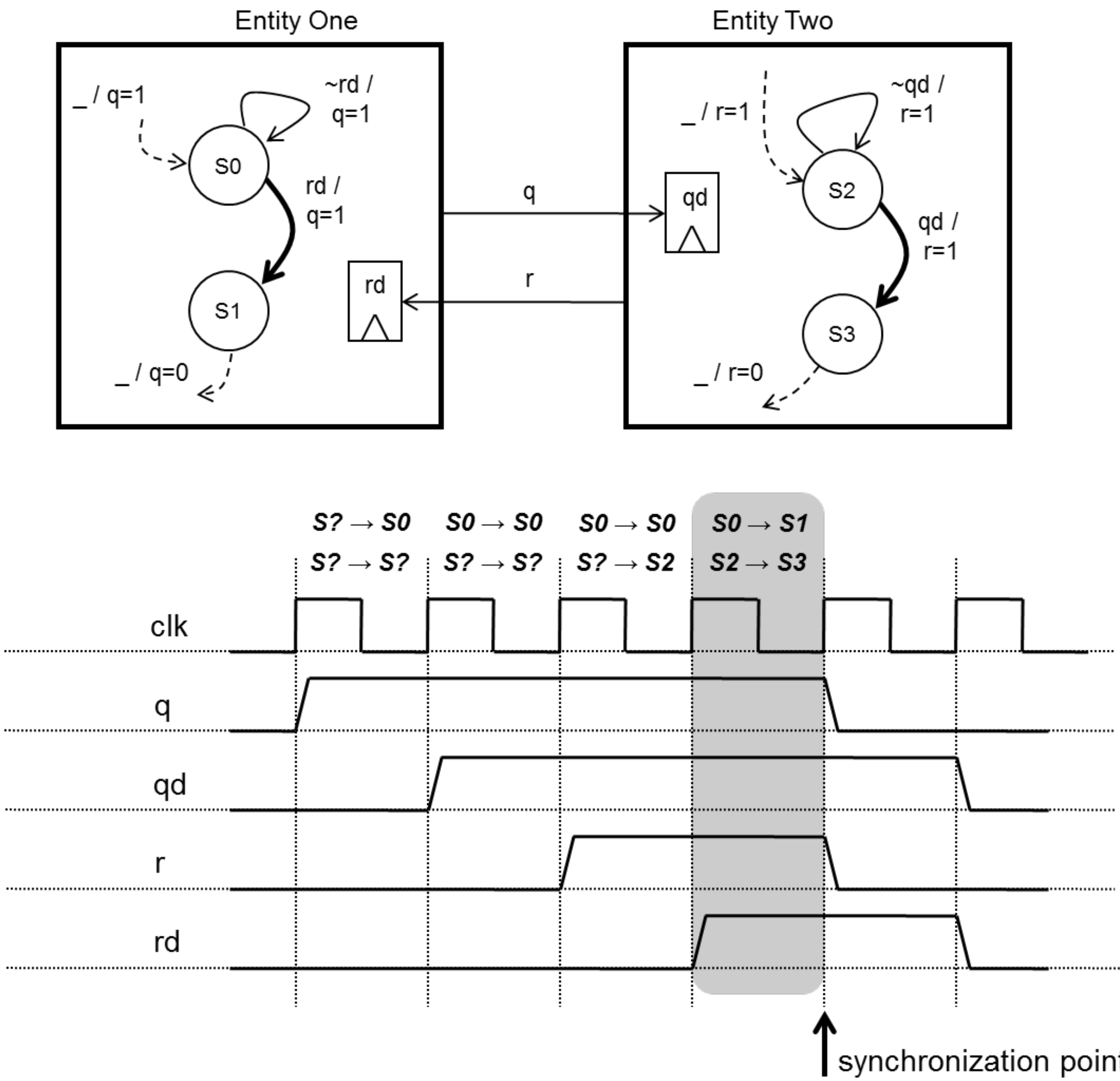
- Semaphores can only guarantee exclusive access to shared resources.
  - Difficult to control precise data transfer
  - Multiple semaphores can be used, but not elegant.
- **Handshaking**: a signaling protocol between two entities to coordinate data transfers.
  - Can handle entities with different speeds.

One-Way Handshake

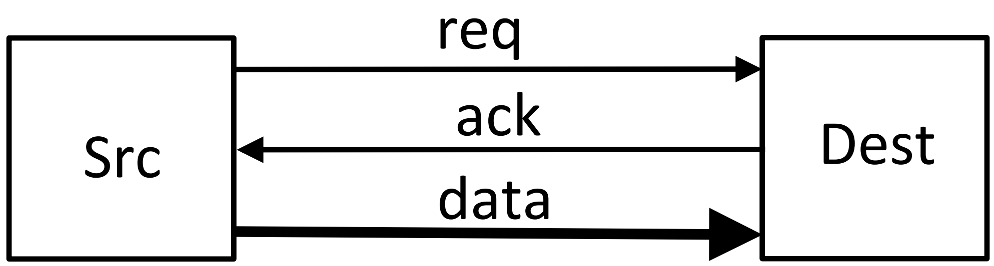


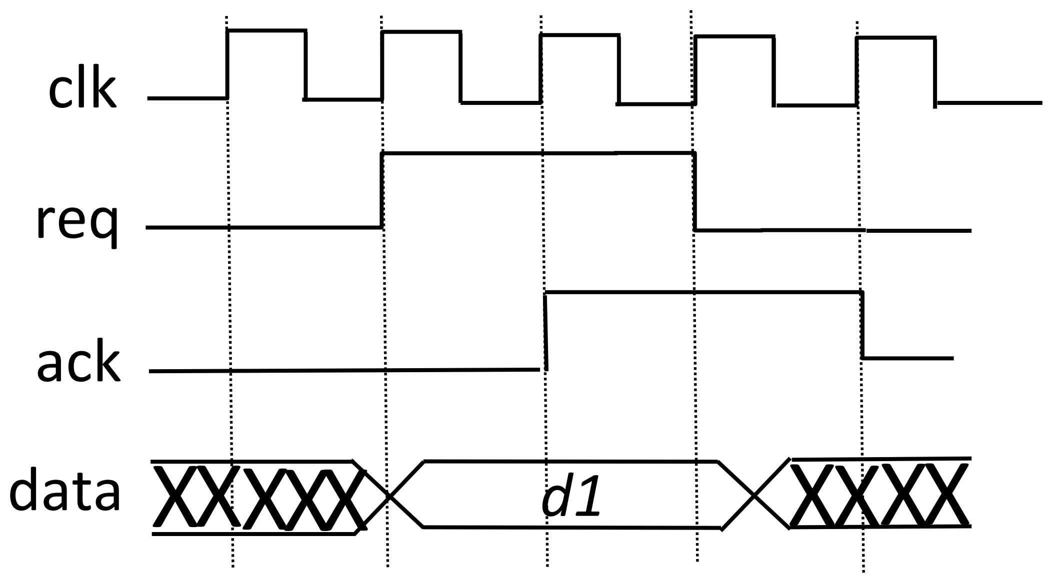
Assume that entity two is slower.

Two-Way Handshake



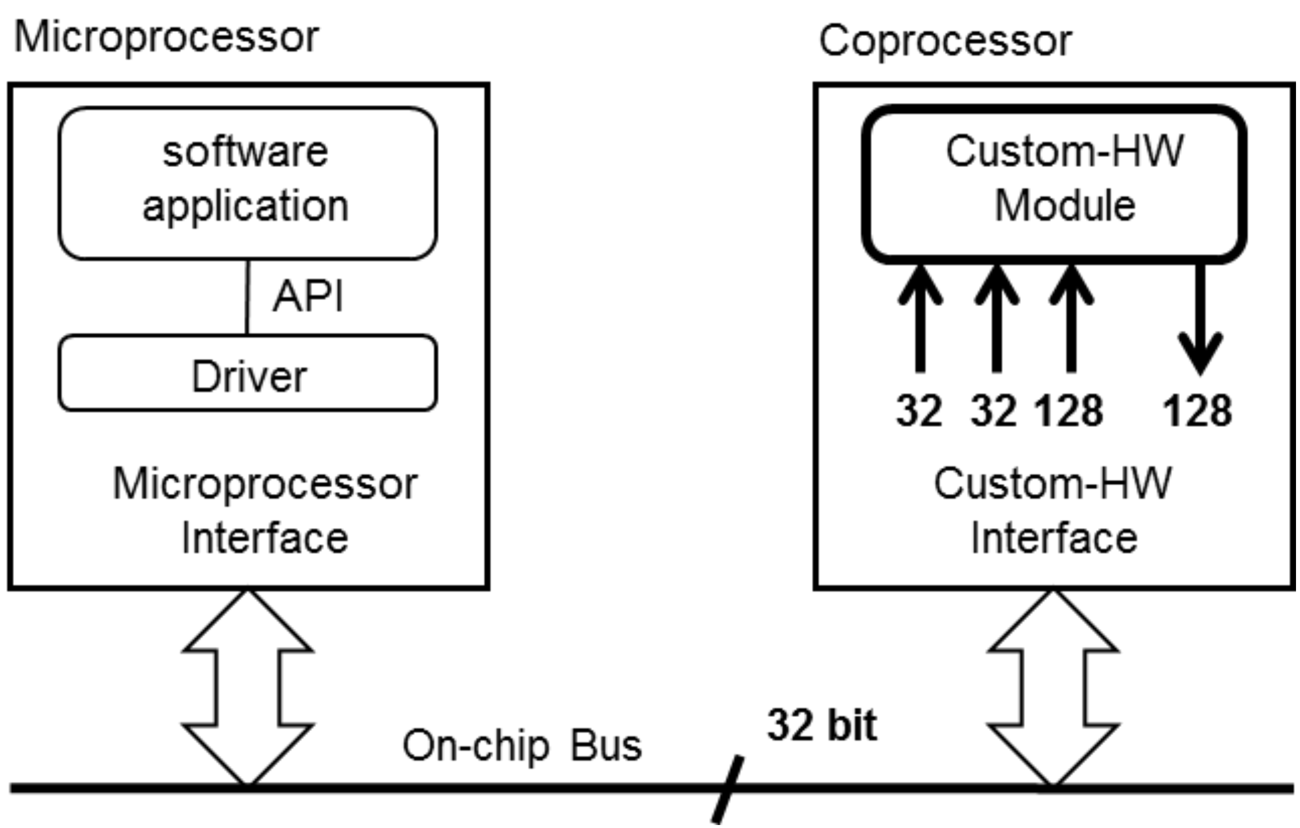
Two-Way Handshake for Data Transfer



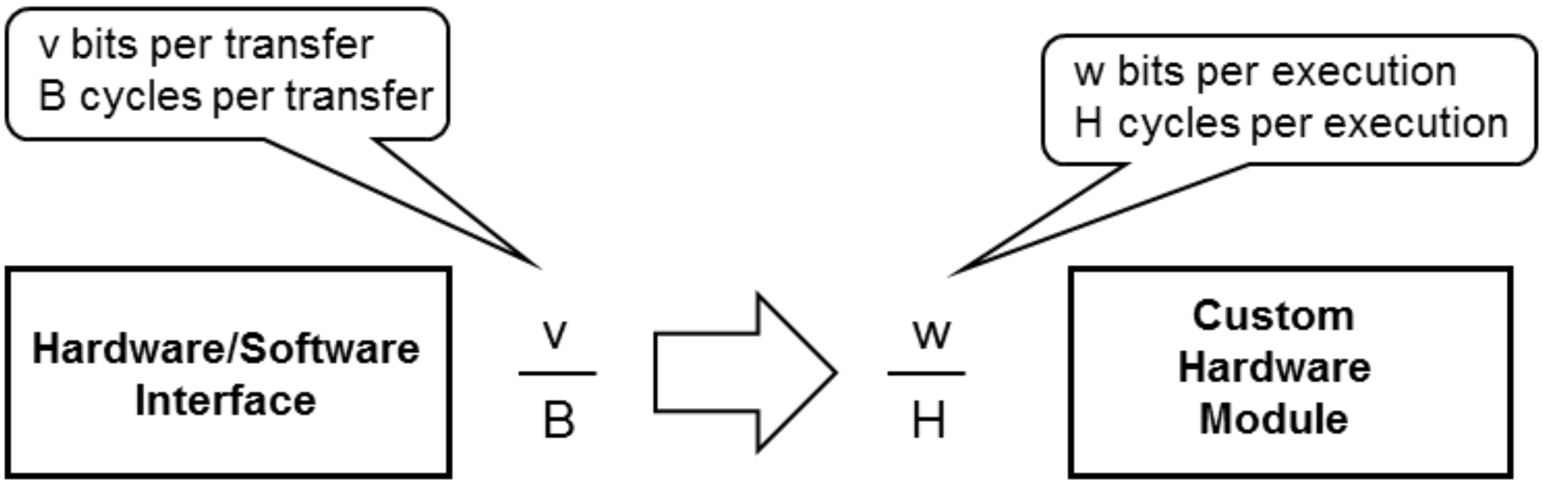


Communication Constrained vs Computation Constrained

System performance should consider both computation performance and communication overhead.



Communication Constrained vs Computation Constrained

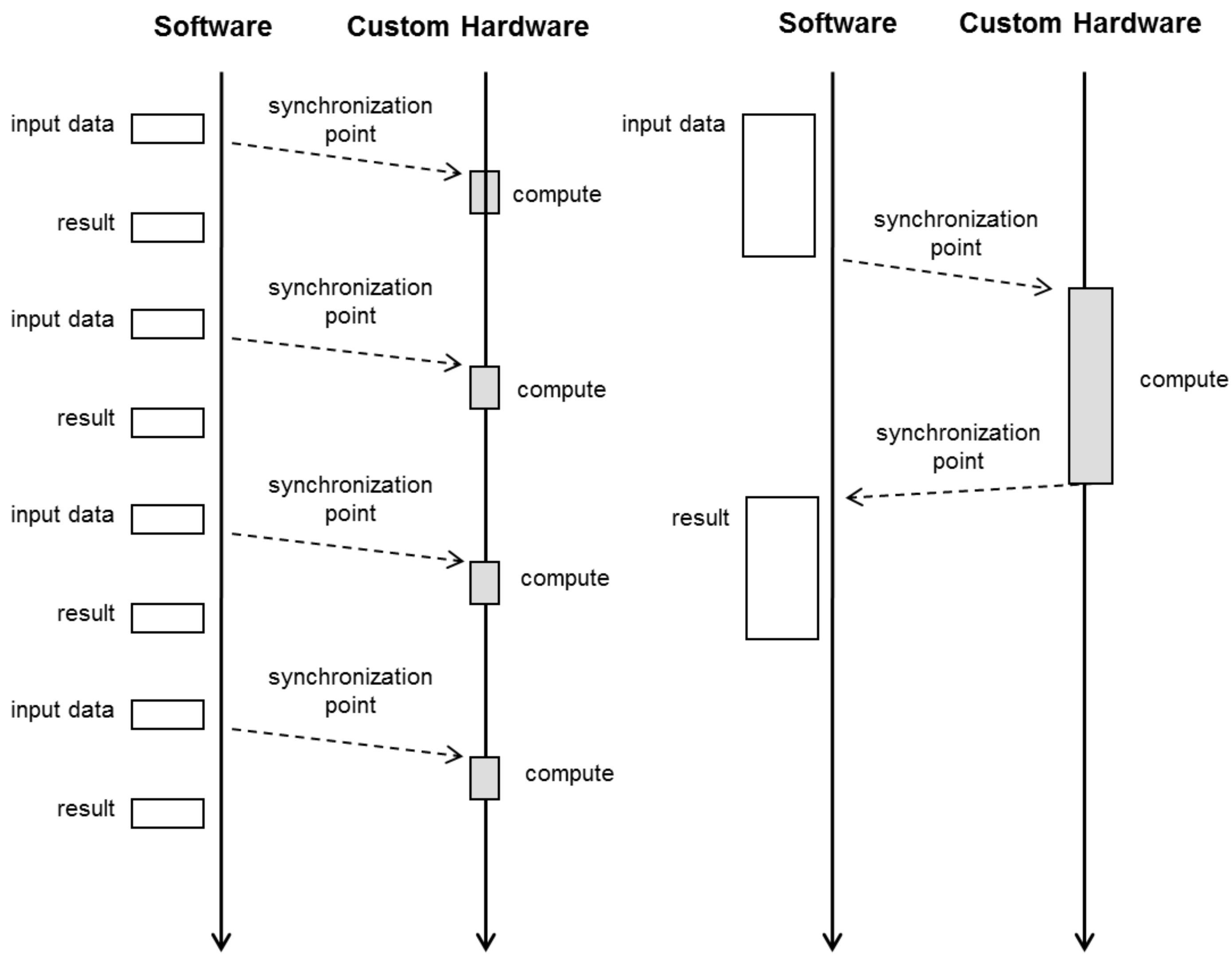


Computation Constrained	$\frac{v}{B} > \frac{w}{H}$
Communication Constrained	$\frac{v}{B} \leq \frac{w}{H}$

Tight and Loose Coupling

Coupling: the level of interactions between two components.

Degree of coupling affects choice and implementation of synchronization.



Dedicated vs Shared Interfaces

Nature of coupling affects the organization of HW/SW interfaces

Factor	Coprocessor interface	Memory-mapped interface
Addressing	Processor-specific	On-chip bus address
Connection	Point-to-point	Shared
Latency	Fixed	Variable
Throughput	Higher	Lower

↑ tight coupling

↑ loose coupling

Reading Guide

- Chapter 9, the *CoDesign* book.