

Asynchronous Computing

2. Communication + Storage *versus* Computation + Flow control

Marly Roncken and Ivan Sutherland
 Asynchronous Research Center (ARC)
 Maseeh College of Engineering and Computer Science
 Portland State University
 July 2018

slide 1 of 20

Building blocks

joint
 computation + flow control

link
 communication + storage

in **out**

Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 2 of 20

Building blocks

Example: sequential finite state machine

joint: compute

link in *link out*

joint: copy

link feedback_in *link feedback_out*

Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 3 of 20

Joint: action

WHEN to act:
in is full
 and
out is empty

joint: copy

full *link in* *link out* empty

WHAT to do:

- copy data
- drain *in*
- fill *out*

joint: copy

empty *link in* *link out* full

Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

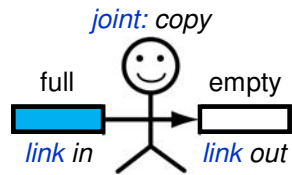
slide 4 of 20

Joint: computation time

WHEN to act:

in is full
and
out is empty

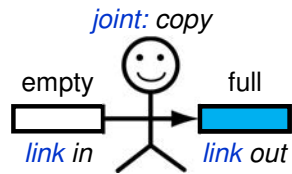
decision time



WHAT to do:

- copy data
- drain *in*
- fill *out*

action time

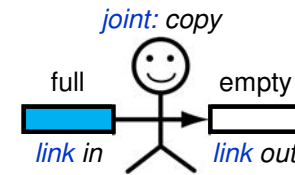


Joint: guarded command specification

WHEN to act:

in is full
and
out is empty

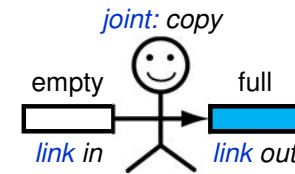
guard



WHAT to do:

- copy data
- drain *in*
- fill *out*

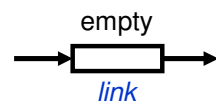
command



Link: guarded command (1/2)

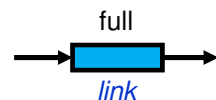
WHEN to act:

fill with data
and
empty



WHAT to do:

- store data + full
- transport to other end

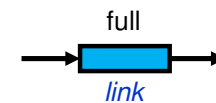


Link contribution to time = transport delay

Link: guarded command (2/2)

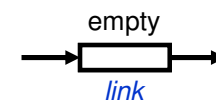
WHEN to act:

drain
and
full



WHAT to do:

- store empty
- transport to other end

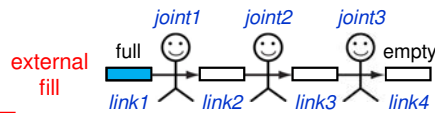


Link contribution to time = transport delay

Systems of building blocks (1/3)

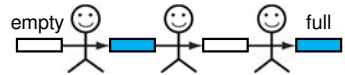
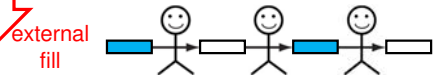
WHEN to act:

in is full
and
out is empty



WHAT to do:

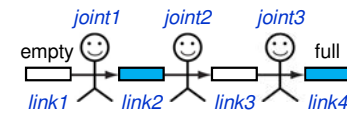
- copy data
- drain *in*
- fill *out*



Systems of building blocks (2/3)

WHEN to act:

in is full
and
out is empty



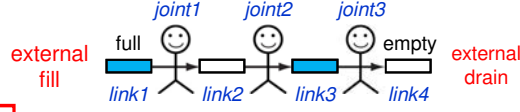
WHAT to do:

- copy data
- drain *in*
- fill *out*

Systems of building blocks (3/3)

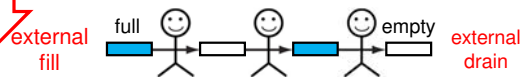
WHEN to act:

in is full
and
out is empty

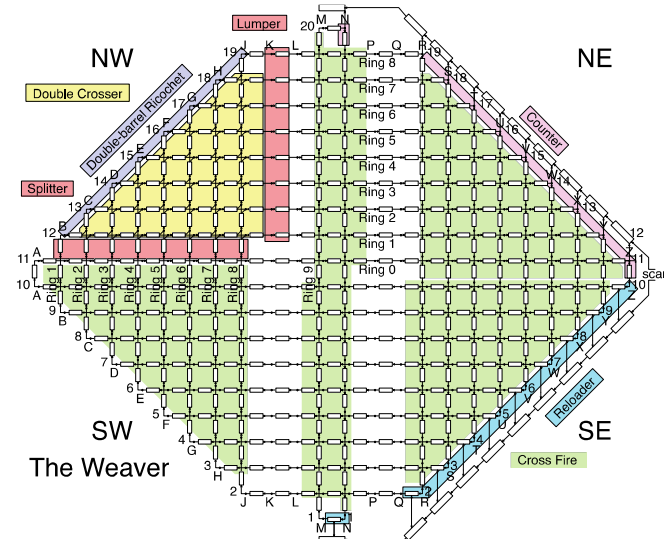


WHAT to do:

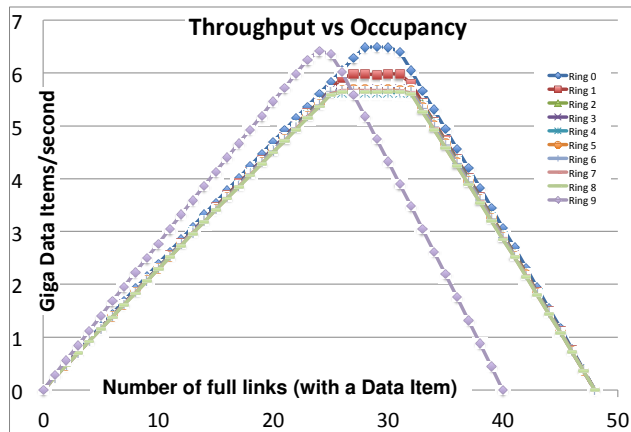
- copy data
- drain *in*
- fill *out*



etcetera



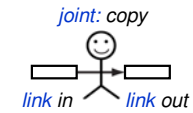
Performance without clocks (Weaver)



Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 13 of 20

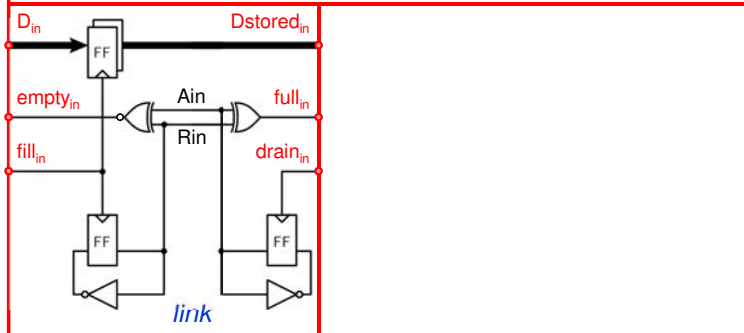
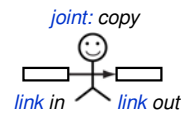
Link-Joint: circuit implementation (1/4)



Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 14 of 20

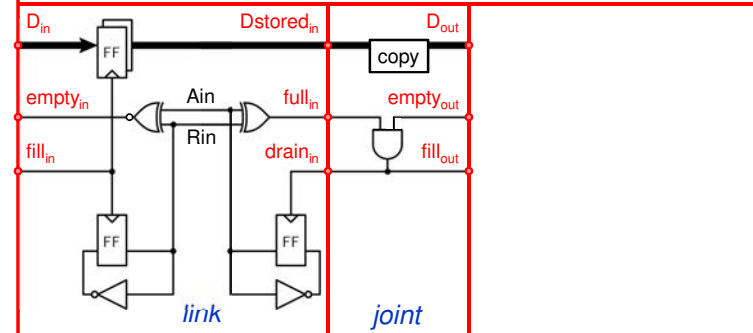
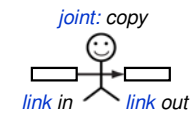
Link-Joint: circuit implementation (2/4)



Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 15 of 20

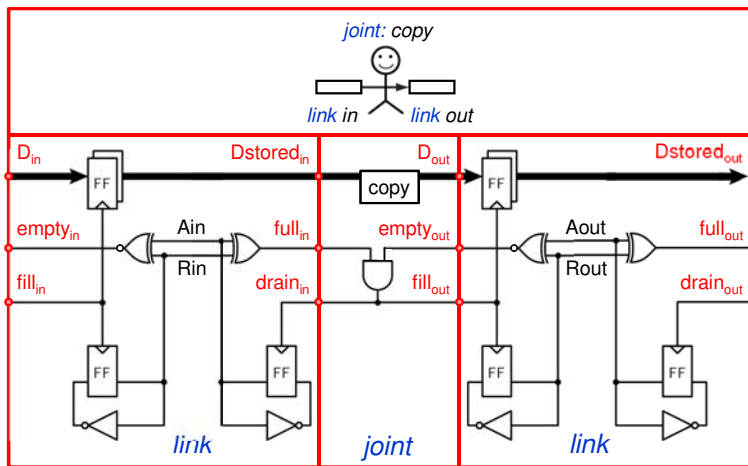
Link-Joint: circuit implementation (3/4)



Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 16 of 20

Link-Joint: circuit implementation (4/4)



Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 17 of 20

Link-Joint: design rules

for robust delay-insensitive interfaces

- **Joint must**
 - use and drain **only FULL** input links
 - use and fill **only EMPTY** output links
 - execute a guarded command as an **atomic action**
 - **arbitrate** if more than one guarded command is enabled
- **Link must**
 - execute a guarded command as an **atomic action**

Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 18 of 20

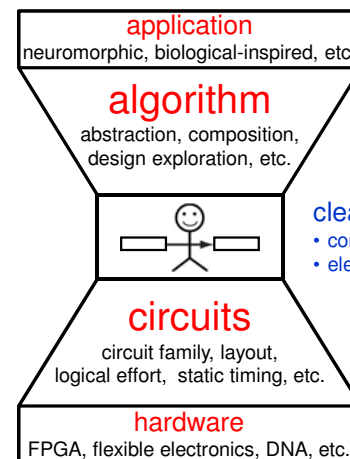
Summary

- **Communication and computation are equal partners**
 - from the bottom up
 - **links** do communication and storage
 - **joints** do computation and flow control
- **Interfaces matter**
 - design them for collaboration and re-use
 - **full-empty** interface
 - works for computer scientists and electrical engineers
 - unifies self-timed circuit families

Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 19 of 20

Vision



clean interface between

- computer scientists
- electrical engineers

After: Kees van Berkel, Handshake Circuits, Fig. 1.1, Cambridge University Press, 1993.

Asynchronous Computing — 2. Computation + Flow control versus Communication + Storage

slide 20 of 20