EENG 426/CPSC 459/ENAS 876
Silicon Compilation

Synchronization

Computer Systems Lab

http://csl.yale.edu/~rajit

Fall 2018

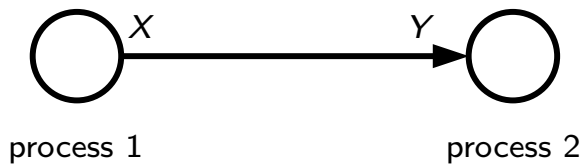
Yale

AVLSI

Communication

Basic paradigm:

- Processes send and receive messages on communication ports.
- Connected ports form a communication channel.
- Messages cannot be received before they are sent
  $\Rightarrow$ synchronization



$X!e$: send e on port X
$Y?v$: receive into v from port Y

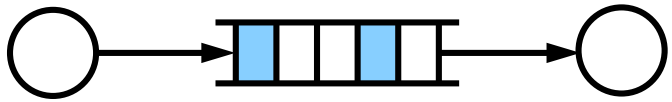
Synchronization slack

To any command X we can attach a counter $\mathbf{c}X$.

$$\mathbf{c}X \equiv \#\text{ of completed } X\text{-actions}$$

We know that $\mathbf{c}S - \mathbf{c}R \geq 0$.

The maximum difference $\mathbf{c}S - \mathbf{c}R$ is called the *slack* of the communication channel.



slack corresponds to
buffering in the channel

There are three possibilities:

- infinite slack
- positive, finite slack
- zero slack

Yale

AVLSI

Semantics of synchronization

We will mostly deal with slack zero channels. For these, we know that:

$$\mathbf{c}S = \mathbf{c}R$$

The completion of the Nth send action coincides with the completion of the Nth receive action.

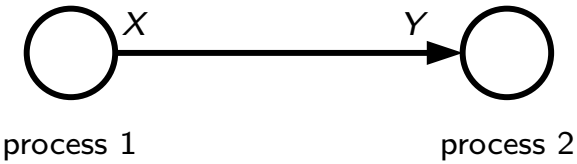
What happens if a send action is reached, and the matching receive action has not yet been reached?

Probes

If the send/receive is reached before the matching receive/send, then the operation blocks.

$\mathbf{q}X \equiv$ is there a pending X-action?

We use the *probe* to determine if the other end of the channel is ready.



$\overline{X}$ denotes the probe of X

Rules: $\overline{X} \Rightarrow \mathbf{q}Y$

$\mathbf{q}Y \Rightarrow \Diamond\overline{X}$

Probes can only occur in guards.

Probes

Stability:

- If $\overline{X}$ is true, it remains true until the next X-action. The true value of a probe is stable.
- If $\overline{X}$ is false, it can change from false to true at any point. The false value of a probe is unstable.

Suspension:

If a communication action is probed, the process will never suspend at the communication action.

$$\overline{X} \longrightarrow \ldots \; X?$$

Yale

AVLSI

Communication

Matching sends and receives implement a form of *distributed assignment*.

$$(X!e \| X?v) \quad \equiv \quad v := e$$

Convention: use the same name to indicate that two ports form a channel.
We usually parameterize a process by its channels:

$$P(A, B) \equiv *[\ \ A?x; B!x\ \ ]$$

Yale

AVLSI

Bullet

One more composition operator:

$$S_1 \bullet S_2$$

Both statements must be communication actions.

$$\mathbf{c}S_1 = \mathbf{c}S_2$$

Useful in development, because it tightly synchronizes different actions.

Operator precedence: bullet, comma, semicolon, parallel

Yale

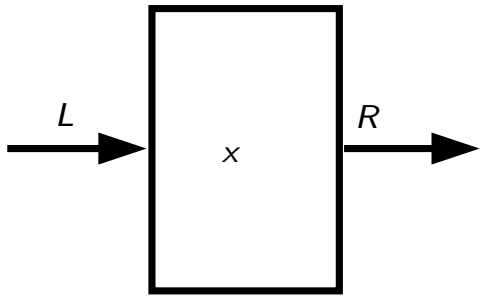
AVLSI

One-place buffer

A one-place buffer:

$$*[\ L?x;\ \ R!x\ ]$$

Can be thought of as slack on a channel, since slack corresponds to buffering.



How do we construct an N-place buffer?

Merge

Two input channels, one output channel. The process merges the two input streams onto the output stream.

$$
\begin{array}{l}
*[\,[\overline{X} \longrightarrow X?a; Z!a \\
\quad |\, \overline{Y} \longrightarrow Y?a; Z!a \\
\quad]\,]
\end{array}
$$

Alternatively,

$$
\begin{array}{l}
*[\,[\overline{X} \longrightarrow X?a \\
\quad |\, \overline{Y} \longrightarrow Y?a \\
\quad]; \;\; Z!a \\
\quad]
\end{array}
$$

Yale

AVLSI