EENG 426/CPSC 459/ENAS 876
Silicon Compilation

Handshaking expansions

Computer Systems Lab

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

Fall 2018

Yale

AVLSI

Handshaking expansions

CHP: high level constructs such as send and receive

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

Production rules: circuit description

$$a \wedge b \mapsto c\downarrow$$
$$\neg a \vee \neg b \mapsto c\uparrow$$

Handshaking expansions: intermediate form

Yale

AVLSI

Handshaking expansions

Handshaking expansions are CHP programs, with the following restrictions:

- only Boolean-valued variables
- no communication actions
- only constants on the RHS of assignments

Yale

AVLSI

Handshaking expansions

Replace

$$x := y$$

with

$$[y \longrightarrow x\uparrow \;[]\; \neg y \longrightarrow x\downarrow]$$

Variables of a process are classified into:

- internal (local variables, not shared)
- input (shared, only read by the process)
- output (shared, written by the process)

Yale

AVLSI

Handshake protocols

We first consider the case when we have bare communication actions (no data being sent/received).

Synchronization is implemented by using two wires.

Two-phase handshake: (initially all variables are false)

$$X: \quad xo\uparrow; [xi]$$
$$Y: \quad [yi]; yo\uparrow$$

X: *active* communication protocol
Y: *passive* communication protocol

$xo = yi; yo = xi$

Yale

AVLSI

Handshake protocols

If all variables are true initially:

$$X: \quad xo\downarrow; [\neg xi]$$
$$Y: \quad [\neg yi]; yo\downarrow$$

Both protocols synchronize the two actions.

Since the final state of one implementation is the initial state of the other, we can *alternate* the two implementations.

Handshake protocols

Problem: it is not always possible to alternate the two implementations.

$$[B \longrightarrow X$$
$$[]\neg B \longrightarrow \textbf{skip}$$
$$]$$

General solution: use the *same* implementation for the two.

(Two-phase handshake protocol)

$$X: \quad xo := \neg xo; [xi = xo]$$
$$Y: \quad [yi \neq yo]; yo := \neg yo$$

This implementation is costly.

Yale

AVLSI

Handshake protocols

Four-phase handshaking:

$$X: \quad xo\uparrow;\ [xi];\ xo\downarrow;\ [\neg xi]$$
$$Y: \quad [yi];\ yo\uparrow;\ [\neg yi];\ yo\downarrow$$

X: *active* communication protocol
Y: *passive* communication protocol

The waits are simplfied, which results in better circuits.

Handshaking expansions

Example:

$$*[L; R]$$

becomes:

$$*[[li]; lo\uparrow; [\neg li]; lo\downarrow; ro\uparrow; [ri]; ro\downarrow; [\neg ri]]$$

or:

$$*[lo\uparrow; [li]; lo\downarrow; [\neg li]; [ri]; ro\uparrow; [\neg ri]; ro\downarrow]$$

or:

$$*[[li]; lo\uparrow; ro\uparrow; [ri]; [\neg li]; lo\downarrow; ro\downarrow; [\neg ri]]$$

Yale

AVLSI

Implementing probes

Example:

$$*[\,[\overline{C} \longrightarrow x\uparrow; C \;\; []\, \overline{D} \longrightarrow x\downarrow; D \;\;]\,]$$

becomes:

$$*[\,[ci \longrightarrow x\uparrow; \underbrace{[ci]}_{omit}; co\uparrow; [\neg ci]; co\downarrow$$

$$\quad []\, di \longrightarrow x\downarrow; \overbrace{[di]}^{omit}; do\uparrow; [\neg di]; do\downarrow$$

$$\,]\,]$$

or:

$$*[\,[ci \longrightarrow x\uparrow; co\uparrow; [\neg ci]; co\downarrow$$

$$\quad []\, di \longrightarrow x\downarrow; do\uparrow; [\neg di]; do\downarrow$$

$$\,]\,]$$

Yale

AVLSI

Implementing probes

Example:

$$*[[\overline{L} \longrightarrow R; L]]$$

becomes:

$$*[[li \longrightarrow ro\uparrow; [ri]; ro\downarrow; [\neg ri]; lo\uparrow; [\neg li]; lo\downarrow]]$$

If port X is probed, we implement it using a passive communication protocol.

Yale

AVLSI

Reshuffling

Since the first part of the four-phase protocol synchronizes the two actions, we can postpone the last part of the protocol.

- This transformation is called **reshuffling**
- Reshuffling changes the order in which signals change
  $\Rightarrow$ different circuit!
- Circuit efficiency can be significantly altered

Reshuffle with care: you might introduce deadlock!

Yale

AVLSI

Lazy-active handshake protocol

Lazy-active protocol:

$$X: \quad [\neg xi]\,; xo{\uparrow}; [xi]\,; xo{\downarrow}$$

The wait for xi to be **false** can always be postponed until the next time the handshake protocol is executed.

Analysis of reshuffling

We can analyze the effect of a reshuffling at the CHP level of abstraction!

A 4-phase handshake is two 2-phase handshakes—i.e., two synchronizations.

- Write all handshaking using 2-phase CHP: only two-phase handshakes allowed. $L_{4\phi} \triangleright L^+_{2\phi}; L^-_{2\phi}$
- Relation between two 2-phase handshakes L and R?
  - Test against two environments: $L; R$ and $R; L$!
  - If both work, then parallel
  - If only one works, in sequences
  - If neither work, then "$\star$"

Analysis of reshuffling

Passive, passive:

$$[li]\,;\, lo{\uparrow};\, [ri]\,;\, ro{\uparrow} \quad \triangleright \quad L^+;\, R^+$$

$$[li \wedge ri]\,;\, lo{\uparrow},\, ro{\uparrow} \quad \triangleright \quad L^+ \star R^+$$

$$[ri]\,;\, ro{\uparrow};\, [li]\,;\, lo{\uparrow} \quad \triangleright \quad R^+;\, L^+$$

Yale

AVLSI

Analysis of reshuffling

Active, active:

$lo\uparrow; [li]; ro\uparrow; [ri] \quad \triangleright \quad L^+; R^+$

$lo\uparrow, ro\uparrow; [li \wedge ri] \quad \triangleright \quad L^+ \| R^+$

$ro\uparrow; [ri]; lo\uparrow; [li] \quad \triangleright \quad R^+; L^+$

Yale

AVLSI

Analysis of reshuffling

Active, passive:

$$lo{\uparrow}; [li \wedge ri]; ro{\uparrow} \quad \triangleright \quad L^+; R^+$$

$$lo{\uparrow}; [ri]; ro{\uparrow}; [li] \quad \triangleright \quad L^+ \| R^+$$

$$[ri]; lo{\uparrow}; [li]; ro{\uparrow} \quad \triangleright \quad L^+ \star R^+$$

$$[ri]; ro{\uparrow}, lo{\uparrow}; [li] \quad \triangleright \quad R^+; L^+$$

Yale

AVLSI

Reshuffling examples

$$*[\,[li]\,;\,ro{\uparrow};\,[ri]\,;\,ro{\downarrow};\,[\neg ri]\,;\,lo{\uparrow};\,[\neg li]\,;\,lo{\downarrow}\,]$$

$\triangleright$

$$*[\,L^+ \star (R^+; R^-); L^-\,]$$

$$*[\,[li]\,;\,ro{\uparrow};\,[ri]\,;\,lo{\uparrow};\,[\neg li]\,;\,ro{\downarrow};\,[\neg ri]\,;\,lo{\downarrow}\,]$$

$\triangleright$

$$*[\,L^+ \star R^+; L^- \star R^-\,]$$

$$*[\,[li]\,;\,lo{\uparrow};\,[\neg li]\,;\,lo{\downarrow};\,ro{\uparrow};\,[ri]\,;\,ro{\downarrow};\,[\neg ri]\,]$$

$\triangleright$

$$*[\,L^+; L^-; R^+; R^-\,]$$

Yale

AVLSI

Implementing bullets

$$L \bullet R \quad \triangleright \quad L^+; R^+; L^-; R^-$$
$$\triangleright \quad L^+; R^+; R^-; L^-$$

Interleave the parts of the handshaking expansion so that neither L nor R can complete unless the other has begun.