EENG 426/CPSC 459/ENAS 876
Silicon Compilation

Syntax-directed translation

Computer Systems Lab

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

Fall 2018

Yale AVLSI

Syntax-directed translation

Goal:

- Convert CHP program into production rules
- Translation will be
  - Correct
  - Inefficient except for the simplest examples

Syntax-directed:

- Construction is by *structural induction* on the syntax of the CHP program

Yale AVLSI

CHP summary

$x := E$, **skip**

$S_1 ; S_2$

$[G_1 \longrightarrow S_1 [] \cdots [] G_n \longrightarrow S_n]$
$[G_1 \longrightarrow S_1 | \cdots | G_n \longrightarrow S_n]$

$*[G_1 \longrightarrow S_1 [] \cdots [] G_n \longrightarrow S_n]$
$*[G_1 \longrightarrow S_1 | \cdots | G_n \longrightarrow S_n]$

$S_1 \| S_2$

$X!e \quad X?v$

Yale AVLSI

Implementing communication

Synchronization with handshaking expansions.

Four-phase handshaking:

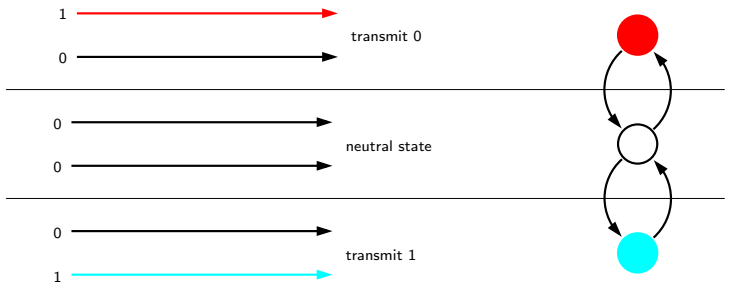
X : $xo\uparrow ; [xi] ; xo\downarrow ; [\neg xi]$
Y : $[yi] ; yo\uparrow ; [\neg yi] ; yo\downarrow$

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

Yale AVLSI

Implementing communication

Dual-rail encoding for data: a standard delay-insensitive (DI) code



Yale

AVLSI

Invariant

To run a statement S, we will assume:

- Input *control* channel C
- Channel consists of two wires (ci, co)
  - When ci goes high, S can start
  - When S is finished, co goes high
  - . . . and the four phase handshake is finished

Yale

AVLSI

Invariant

To evaluate a one-bit expression, we assume:

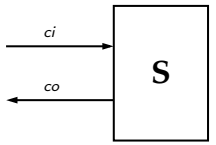
- Input *control* wire
- A dual-rail output that specifies the value
- Variables will have a dual-rail value
- Expressions will built using structural induction

(Primitives: AND, OR, NOT, variable)

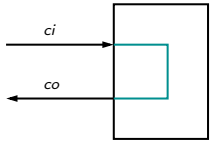
Yale

AVLSI

Translating **skip**



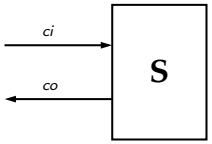
≡



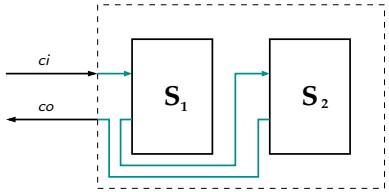
Yale

AVLSI

Translating $S_1; S_2$



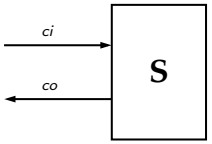
≡



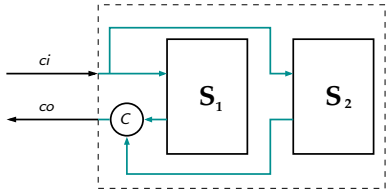
Yale

AVLSI

Translating $S_1 \parallel S_2$



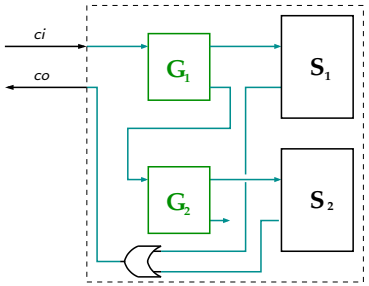
≡



Yale

AVLSI

Deterministic selection statements

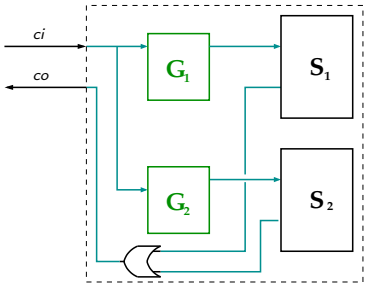


(In this version, only local variables are used)

Yale

AVLSI

Deterministic selection statements



(In this version, only local variables are used)

Yale

AVLSI