# Beyond classical circuit design
# lecture 6

## Synthesis

# Further Reading

Alain J. Martin: *Synthesis of Asynchronous VLSI Circuits.* Tech report California Institute of Technology, 1991.

Alain J. Martin and Mika Nyström: *Asynchronous techniques for system-on-chip design.* Proceedings of the IEEE Volume 94, Issue 6:1089 - 1120, June 2006.

Edmund M. Clarke, Orna Grumberg and Doron Peled: *Model Checking.* MIT Press, 1999.

# What we had...

- Bare Handshakes -> Communication
- Isochronic fork assumption

- Coming up: Synthesis
  - CSP -> PR
  - PR -> transistors

# Synthesis

**In:** CHP

**Out:** Circuit (= Production rules + constraints)

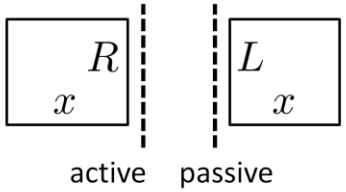
The snythesis problem

## Example: 1-bit channel

Send $x \in \{0, 1\}$

active send,

passive receive




CHP:

sender: $Ps : *[\ldots R!x \ldots]$

receiver: $Pr : *[\ldots L?x \ldots]$

We want to synthesize a 1-bit communiction channel between a sender and receiver (push based).

# Example: 1-bit channel

Choice: dual-rail encoding.

$$R!x \quad rd := x; [v(ra)]; rd := neutral; [n(ra)]$$

$$[x \rightarrow rd.1 \uparrow || \bar{x} \rightarrow rd.0 \uparrow]; [ra];$$
$$(rd.1 \downarrow || rd.0 \downarrow); [\bar{ra}]$$

Start with the abstract channel CHP as we learned it.
Our first choice, the encoding. -> modified CHP.
insert valid and neutral predicates for the chosen code.

## Example: 1-bit channel

$$L?x \qquad [v(ld)]; y := ld; la\uparrow; [n(ld)]; la\downarrow$$

$$[ld.1 \vee ld.0]; [ld.1 \rightarrow x\uparrow \,||\, ld.0 \rightarrow x\downarrow]; la\uparrow;$$
$$[\neg(ld.1 \vee ld.0)]; la\downarrow$$

$$[ld.1 \rightarrow x\uparrow \,||\, ld.0 \rightarrow x\downarrow]; la\uparrow;$$
$$[\neg(ld.1 \vee ld.0)]; la\downarrow$$

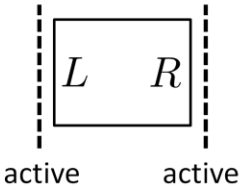
2nd CHP: the first predicate can be removed. It is redundant because the select is blocking anyway.

We finally obtained low-level CHP for sender and receiver. We will see at another example of how to fully get to PRs from CHP.

# Example: active-active sequencer

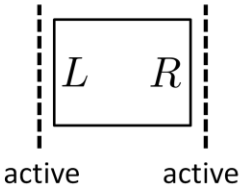
two active interfaces L, R.

CHP: $P1 : *[L; R]$



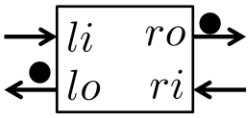
# Example: active-active sequencer

two active interfaces L, R.



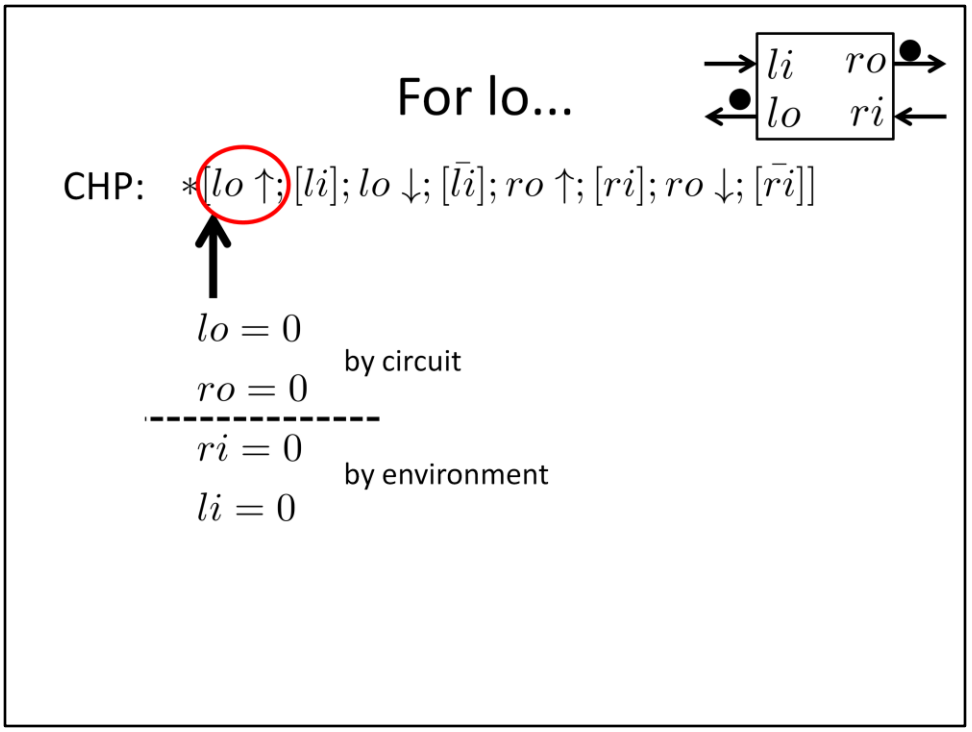

CHP: $P1 : *[L; R]$

Choice: 4-ph hs

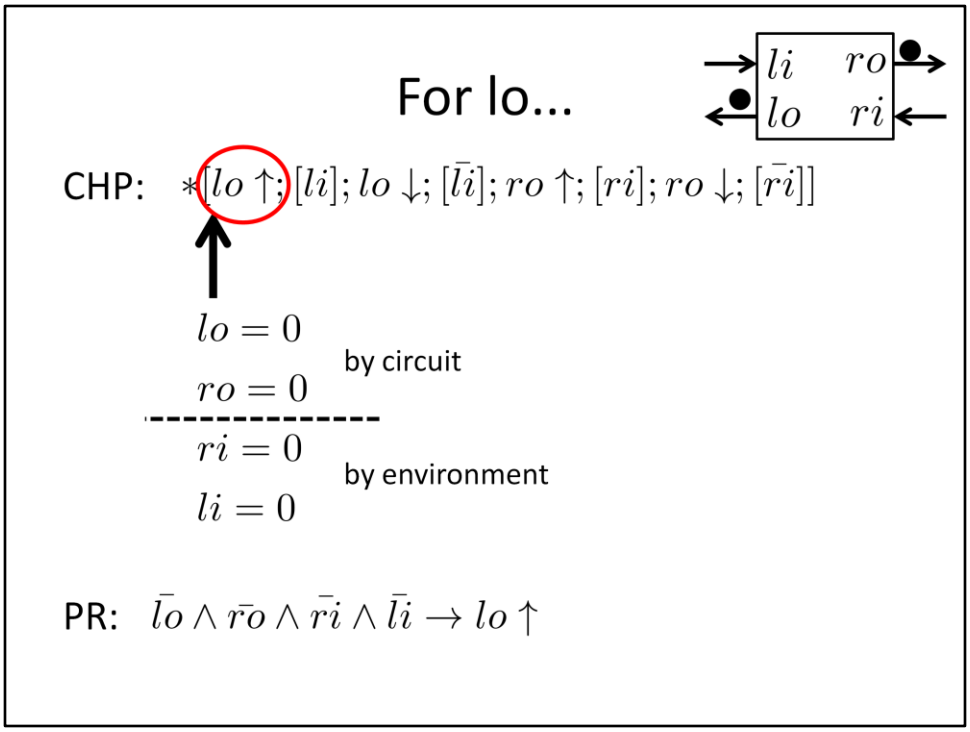



$$*[lo\uparrow; [li]; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; ro\downarrow; [\bar{ri}]]$$

Here: choice was for 4-phase handshaking as a sequencing protocol.

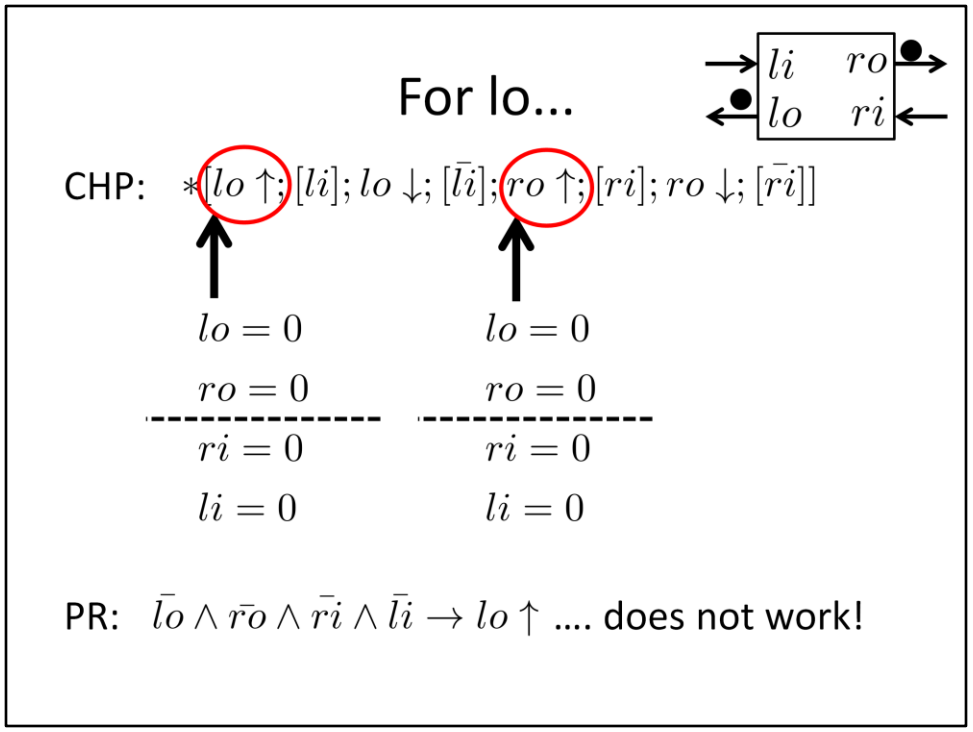



Synthesis process:
start with first command and note the variable states for it. Variables are both set from the circuit and from the environment.

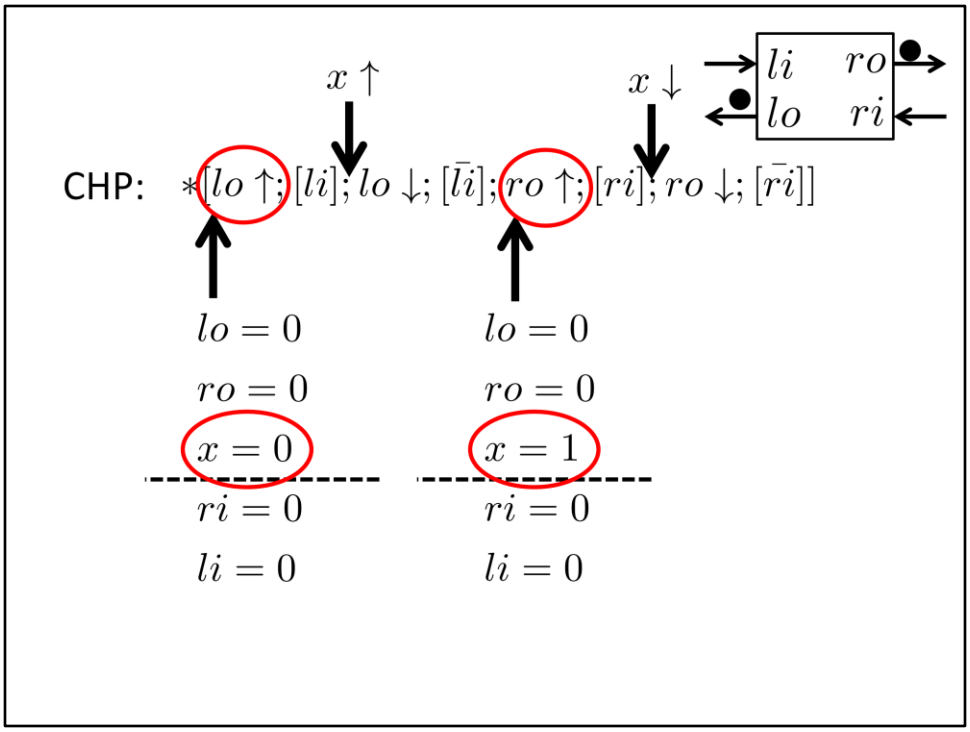



by circuit section: variable states derived from the CHP of the circuit alone
by environment section: variable states derived from the environmental behavior + circuit. e.g. [not ri] at end of CHP. Before executing lo-up still ri=0 since environment does not rise lo without a lo-up transition.

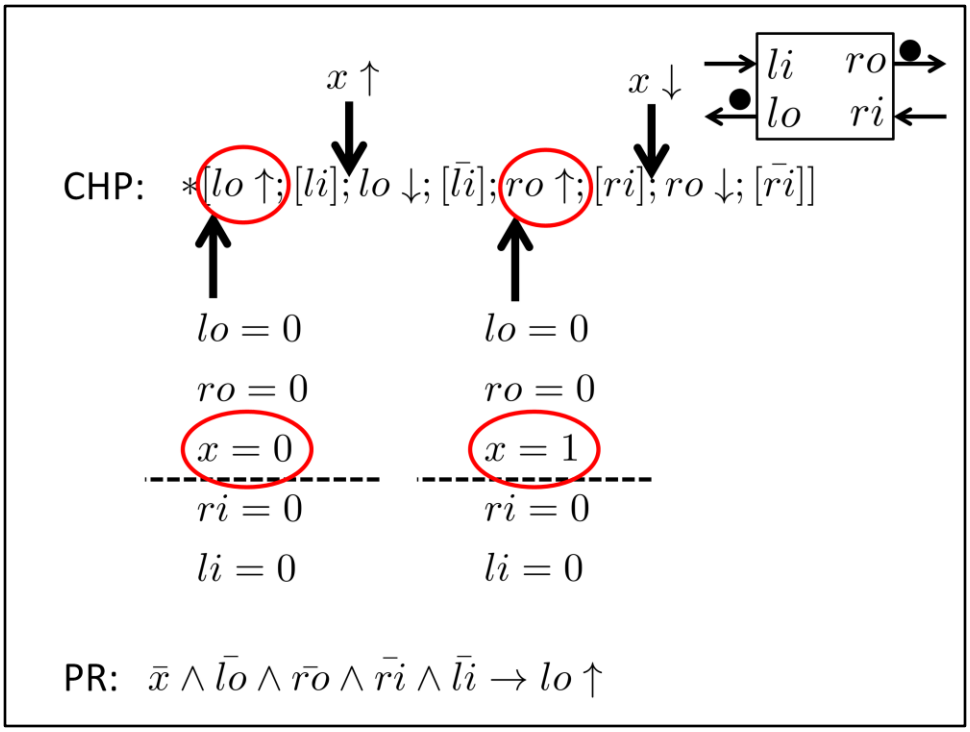
write a production rule from the state guarantees we get for the action. Here: predicate depending on *all* variables.



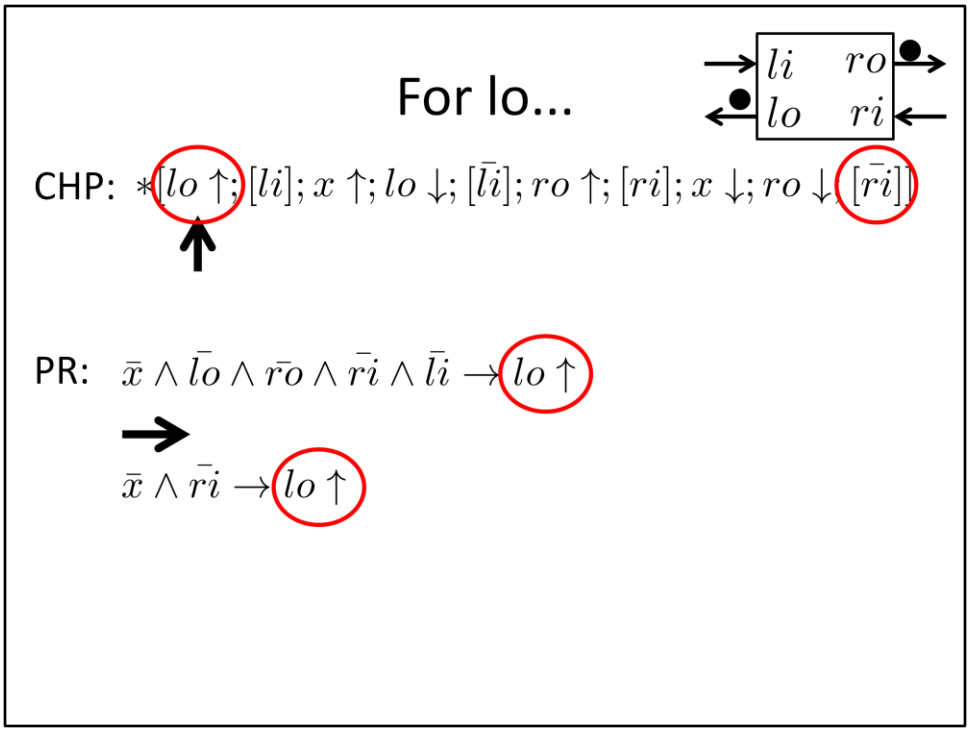

Problem: the action for ro-up needs the same predicate as lo-up.
But [!]: lo-up and ro-up should be executed at different times according to the CHP.
This cannot be expressed by the variables in the CHP!
-> we need to add helper variables.



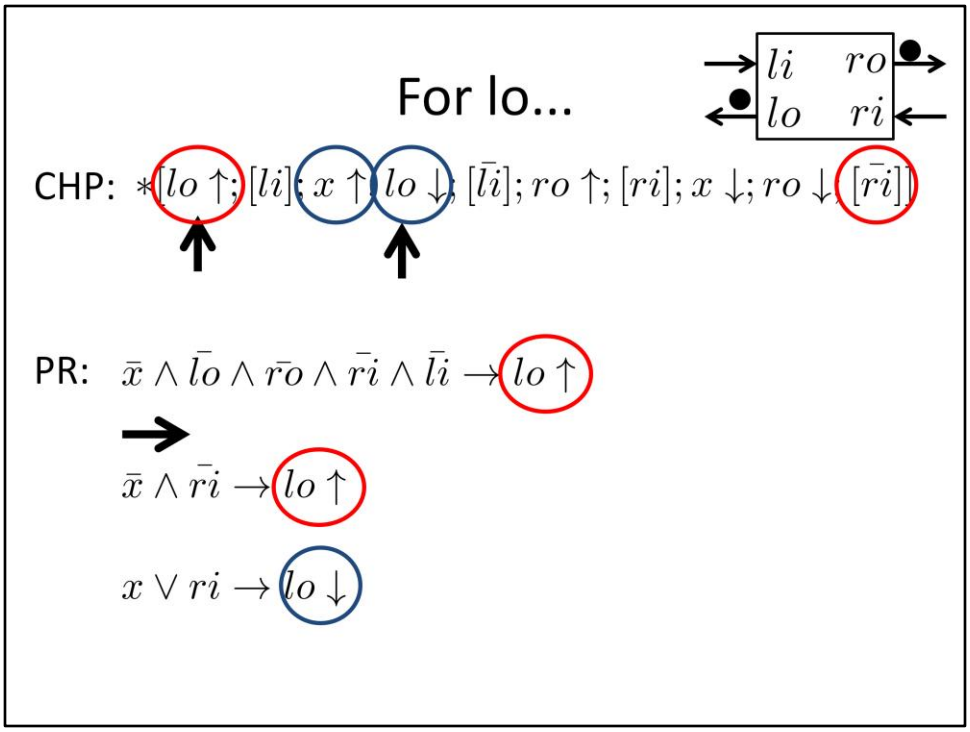

adding the helper variable x.



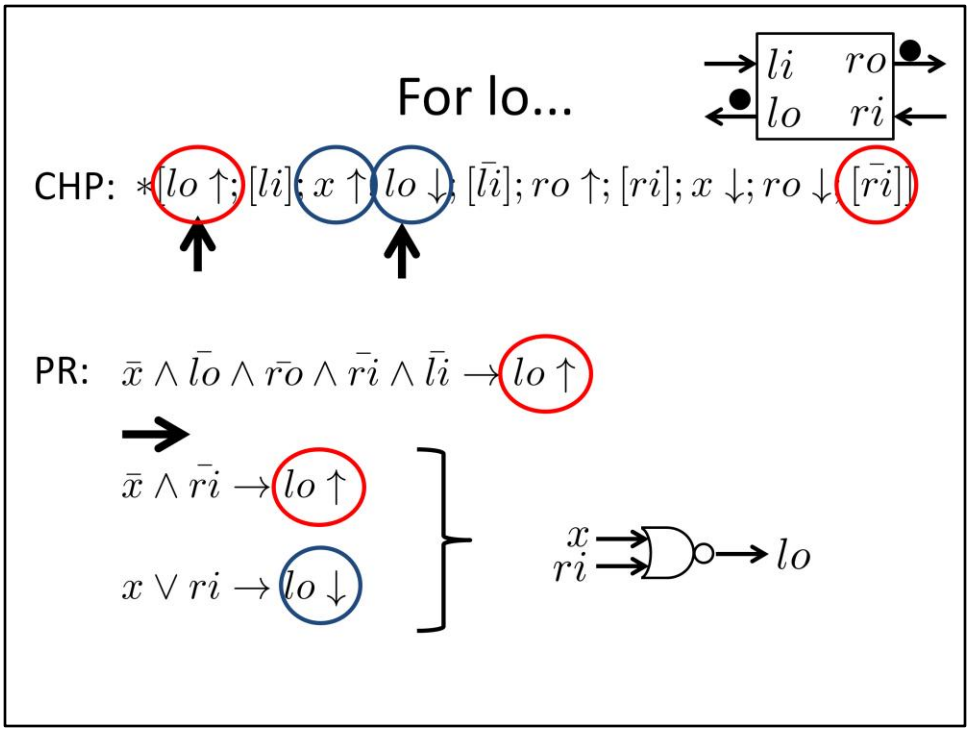

New PR with the helper variable.



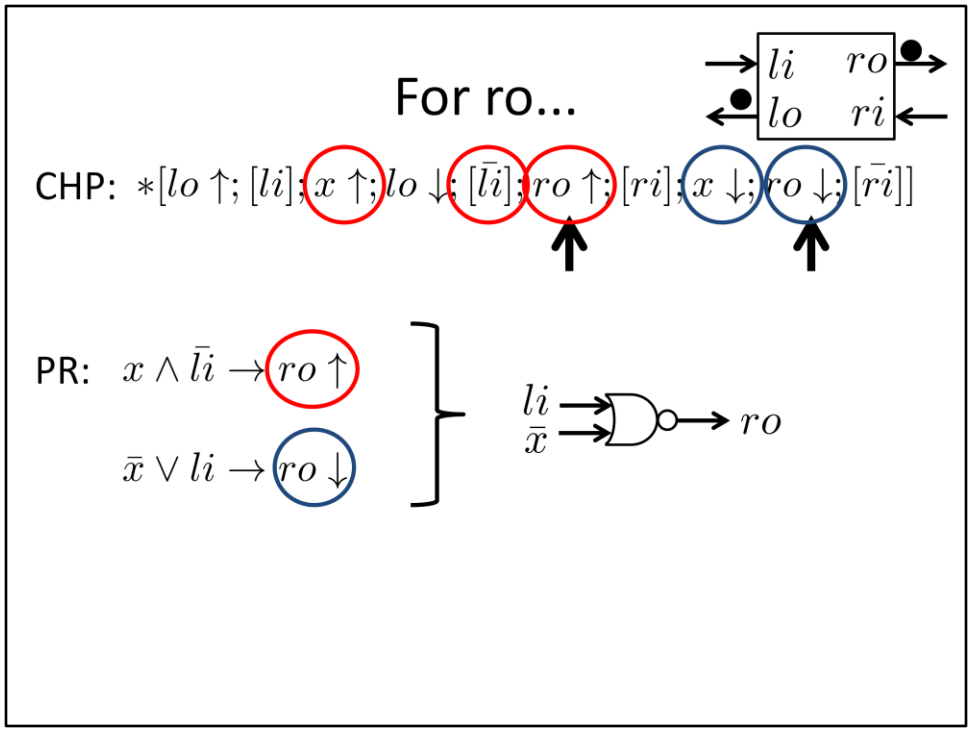

opimizing. minimum guard sufficient to decide when to trigger lo up. neg ri & helper variable are sufficient.




now write the guard when it should go down.

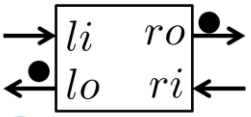



implementation as combinational gate is favorable since small and fast.
-> try to make gates combinational instead of state holding if possible.
here: 2OR.



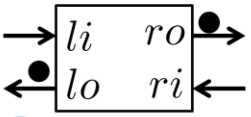

now for the ro signal
again a combinational gate

For x...




CHP: $*[lo\uparrow;[li];x\uparrow;lo\downarrow;[\bar{li}];ro\uparrow;[ri];x\downarrow;ro\downarrow;[\bar{ri}]]$

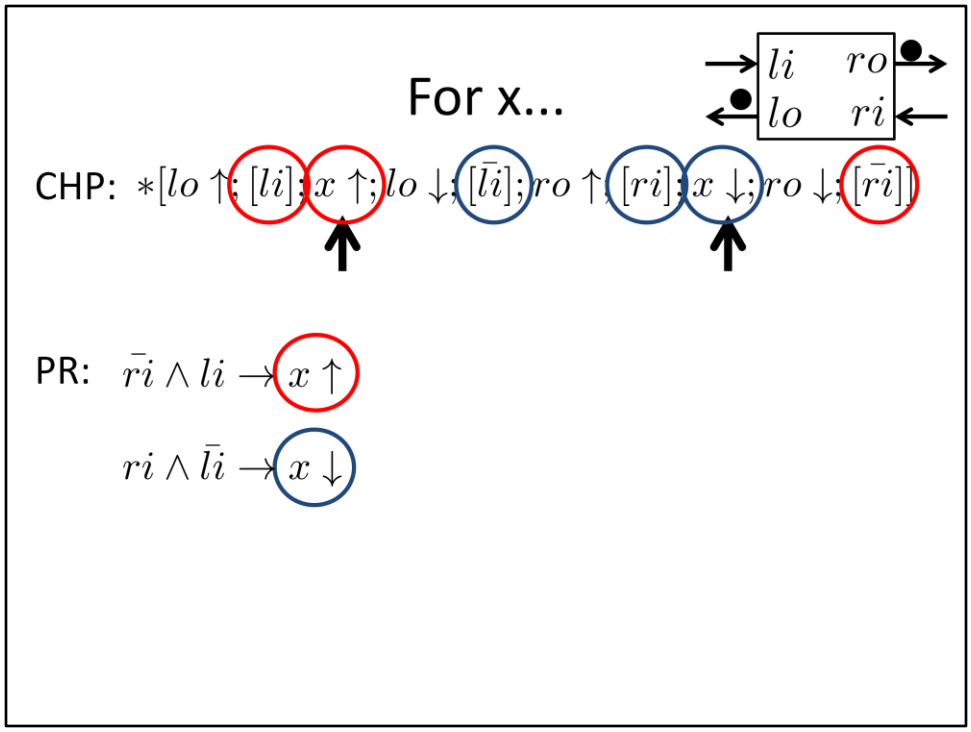
For x...



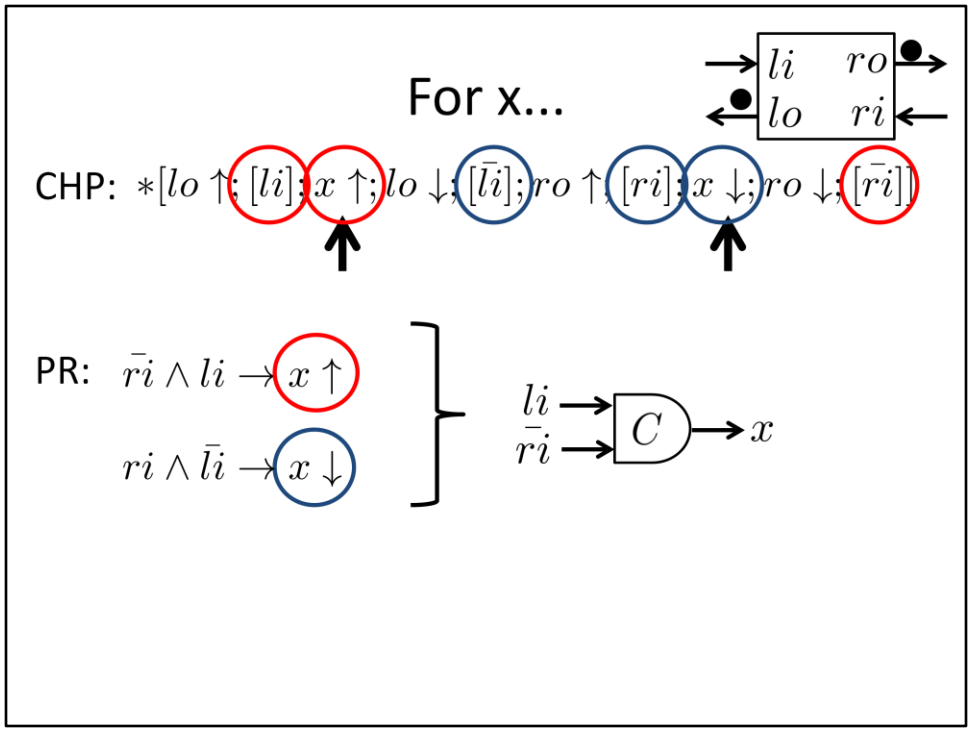

CHP: $*[lo\uparrow;[li];x\uparrow;lo\downarrow;[\bar{li}];ro\uparrow;[ri];x\downarrow;ro\downarrow;[\bar{ri}]]$

… What gate is this?

this time it doesn't look like we can do with a combinational gate.

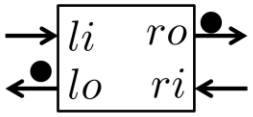



Writing it this way seems more favorable.



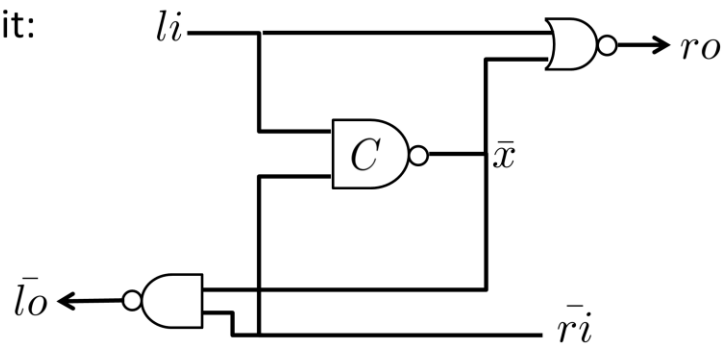

-> not combinational but a C-Element.

## Putting it together



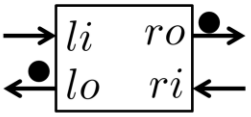

CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

Circuit:



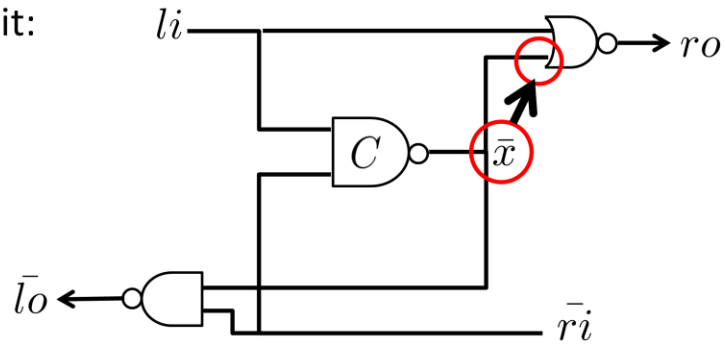

This is our result from synthesis. But can we simply put it together?

## Mind wires...



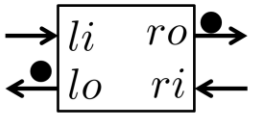

CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

Circuit:



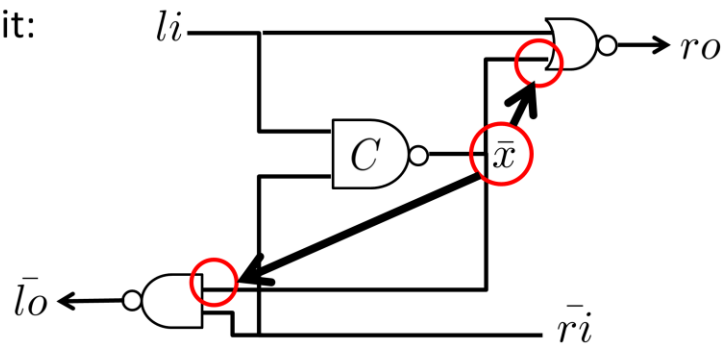

we accounted for delays with our PRs

## Mind forks...



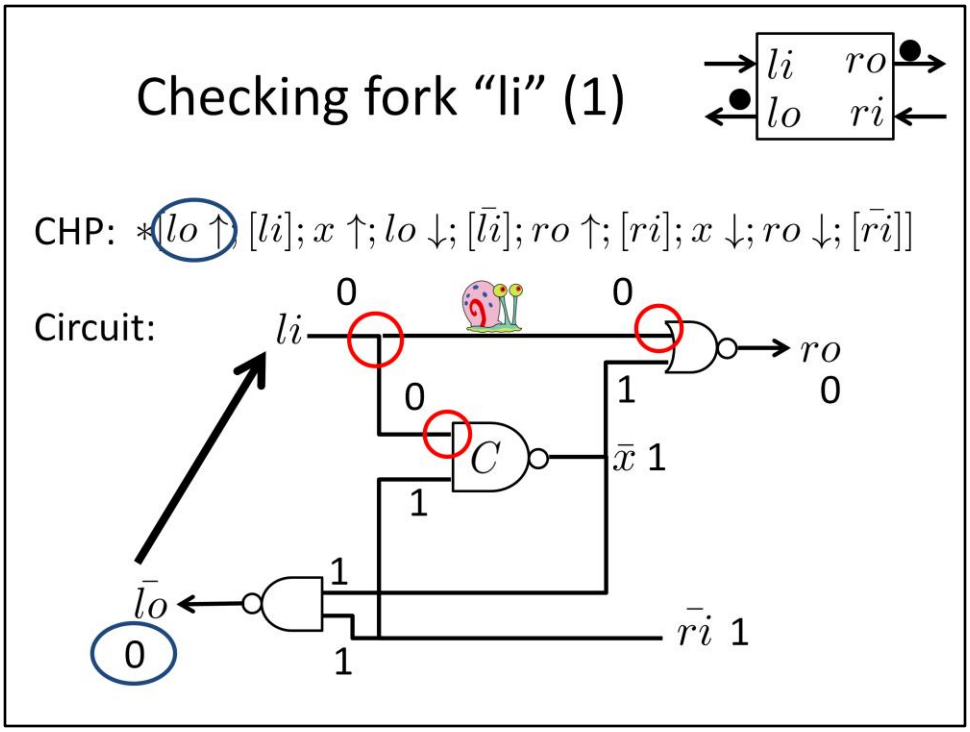

CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

Circuit:

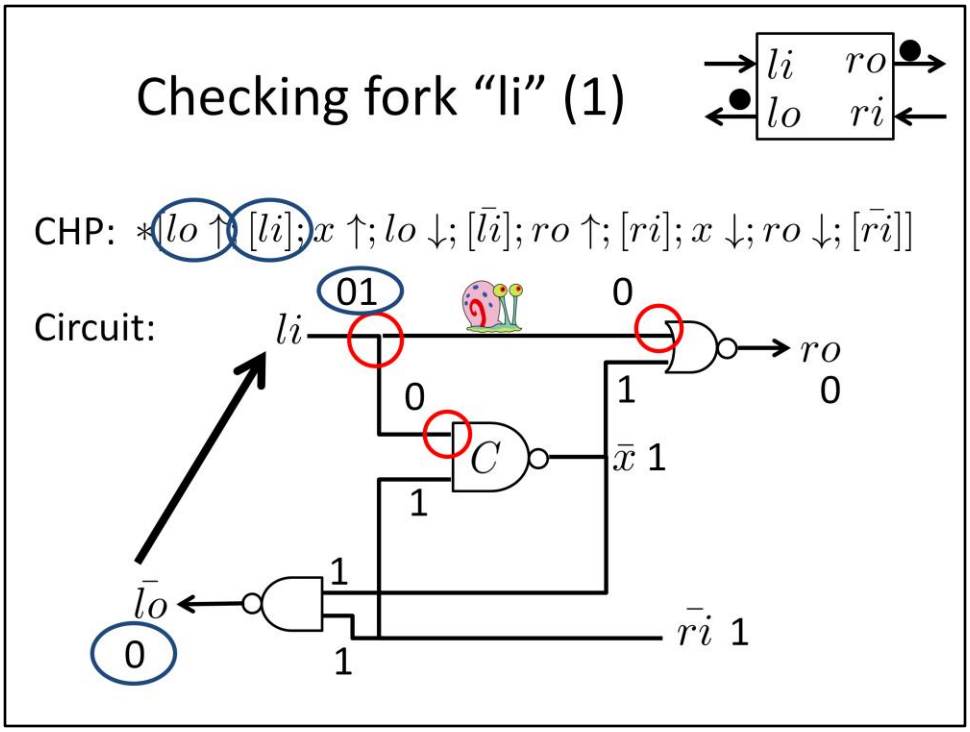



But: we assumed during synthesis that the views are the same. once x is updated, it is viewed the same at all gate inputs.

That is: we assumed *all* forks to have exactly the same delay. To be isochronic.
Do we really need all the be isochronic. Lets check…

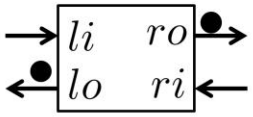



assume slow upper li teeth



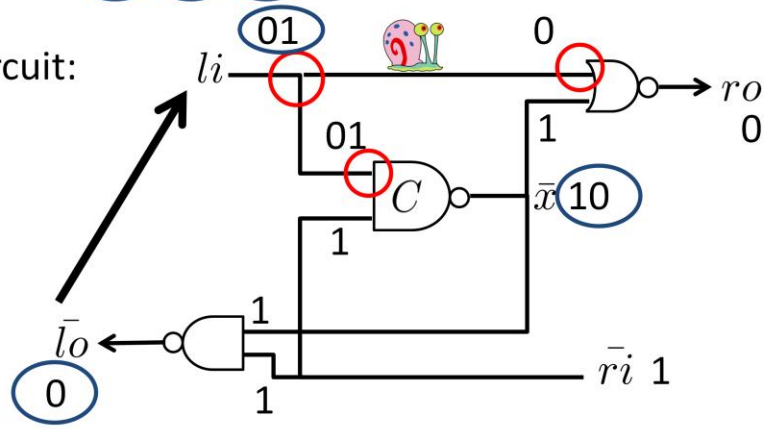

An example execution.

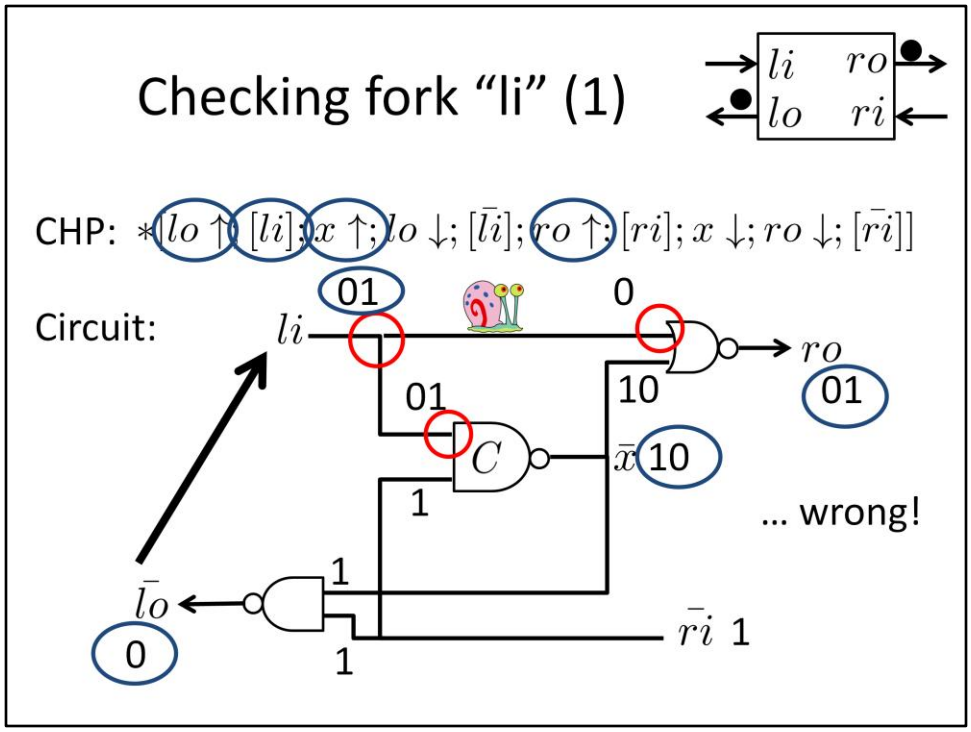
# Checking fork “li” (1)




CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

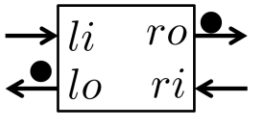
Circuit:





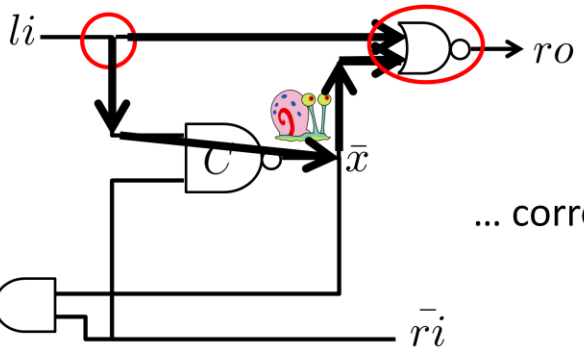

violation to our CHP specification.

## + one-sided Constraint




CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

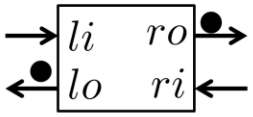
Circuit:




... correct.

prevent this from happening by one-sided delay constraint

## + one-sided Constraint



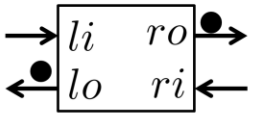

CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

Proving correct:

e.g. by induction.

base case: start with initial states and prove ordering of events from there for the first loop

## + one-sided Constraint



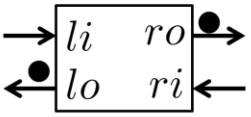

CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

proof:

examine ordering in time "<".

1) **lo-up < [li]:** guaranteed by environment: Initially li = 0. Can be set to li = 1 only by environment. Environment does this only after lo = 1.

## + one-sided Constraint

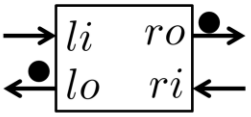



CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

proof:

2) **[li] < x-up:** x = 1 can happen only after both C-Element inputs are 1. This can happen only after li = 1 for the first time.

## + one-sided Constraint




CHP: $*[lo\uparrow; [li]; x\uparrow; lo\downarrow; [\bar{li}]; ro\uparrow; [ri]; x\downarrow; ro\downarrow; [\bar{ri}]]$

proof:

3) **x-up < lo-down:** this can happen only if one of the NAND inputs becomes 0 for the first time.

We first show that lo-down cannot happen because of the input connected to not ri becoming 0: .....

[hw project]

# Hw project

- Prove: PR circuit implements CHP circuit:

  all executions generated by PR circuit fulfil CHP properties.

- Use: NuSMV

  model checker

  see e.g. nusmv.fbk.eu/NuSMV/tutorial

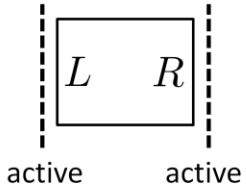
# Hw project

- Input:
  - automaton specification (PRs) + fairness condition
  - Linear Temporal Logic (LTL) formula to verify
- Output:
  - formula holds vs. does not hold + counterexample

# active -> passive

active L, active R,

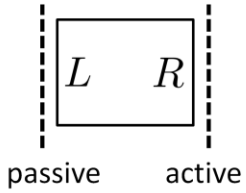
CHP: $P1 : *[L; R]$



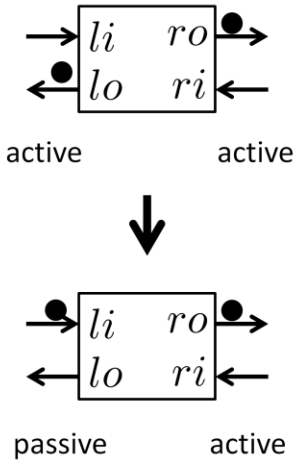

→

passive L, active R,

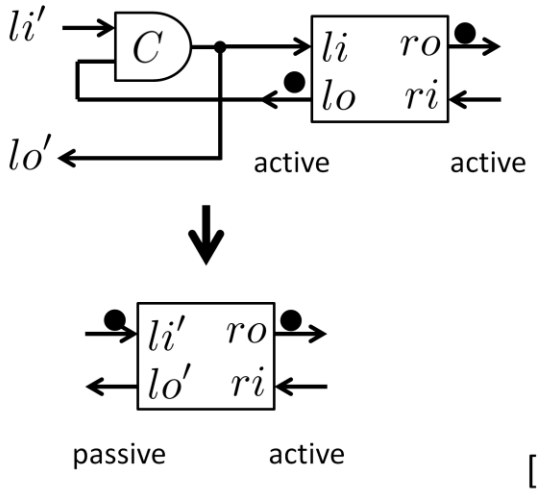
CHP:



# 4ph-hs active -> passive



# 4ph-hs active -> passive




[hw]

# Beyond classical circuit design
# lecture 6.5

# Gate internals

# Further Reading

Simon M. Sze, Kwok K. Ng: *Physics of Semiconductor Devices. 3rd edition.* Wiley, 2006.

Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic: *Digital Integrated Circuits. A Design Perspective. 2nd edition.* Prentice Hall, 2003.

# Why look inside?

- Why is stability important?
  -> Runts/metastability
- What if it does not hold?
  -> Dealing with them.

- Optimization for speed/power/size/robustness...

# In principle…

Schrödinger equation in space and time

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = H\psi(x,t)$$

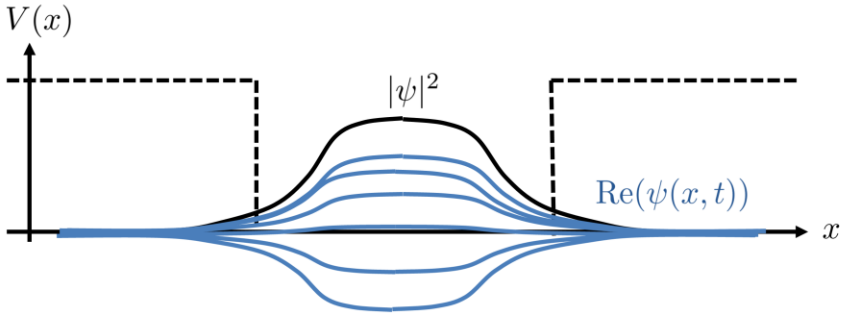
in one dimension

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = -\frac{\hbar\partial^2}{2m\partial x^2}\psi(x,t) + V(x,t)\psi(x,t)$$

# In principle…

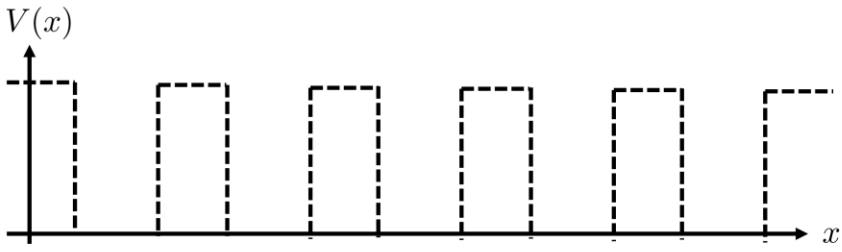
assuming steady state

$$\psi(x,t) = e^{i\omega t}\psi(x)$$



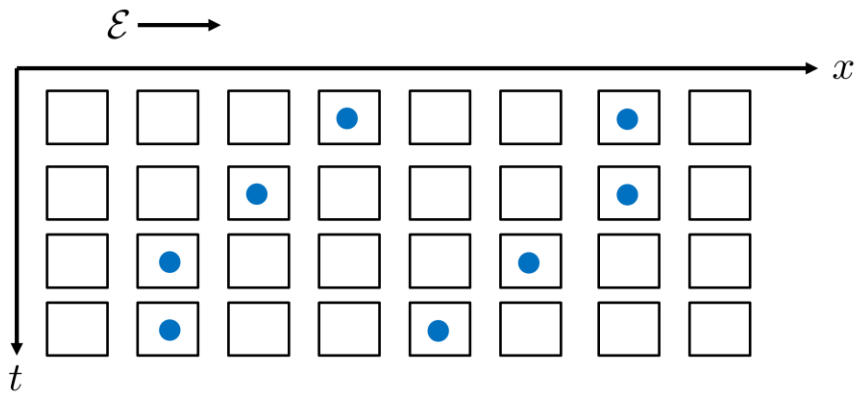
# In principle…

simplified model: single e- in periodic potential



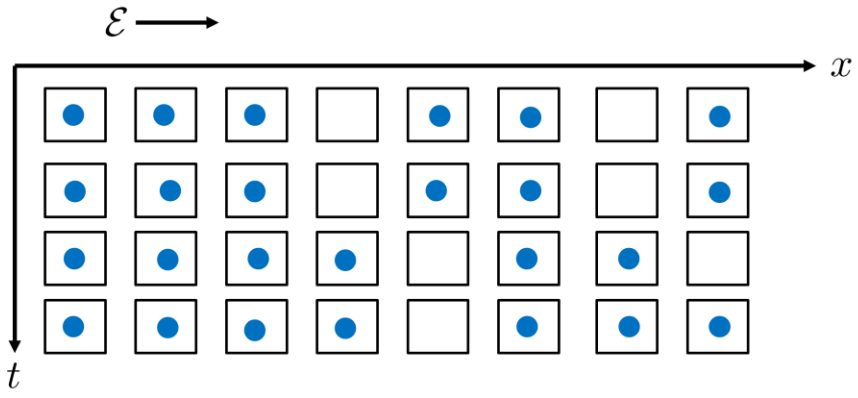
# Carriers

$e^-$ electron



# Carriers

$e^-$ electron



# Carriers

$e^-$ electron $h^+$ hole