

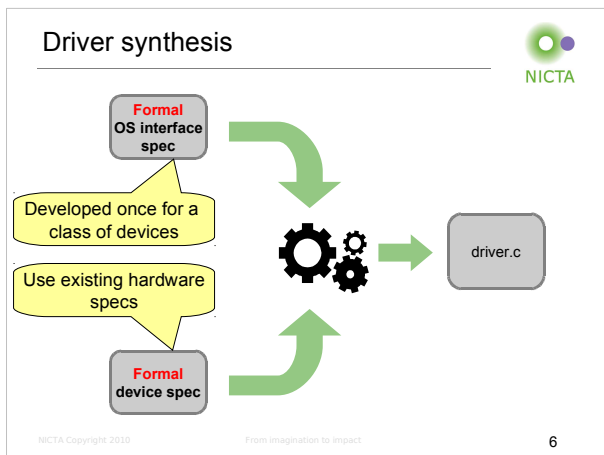
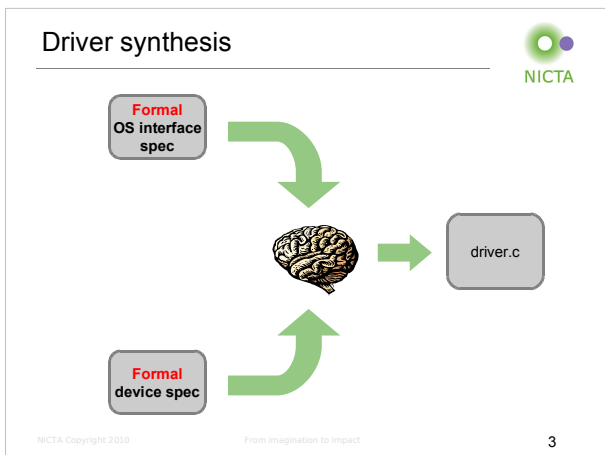
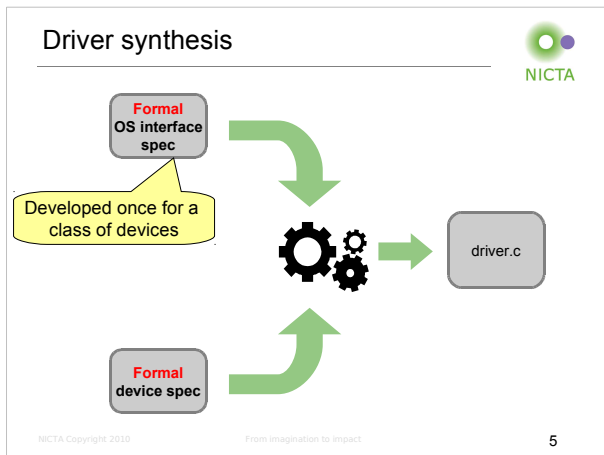
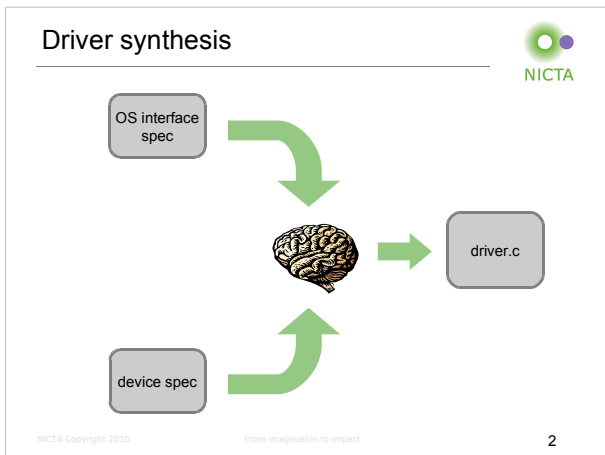
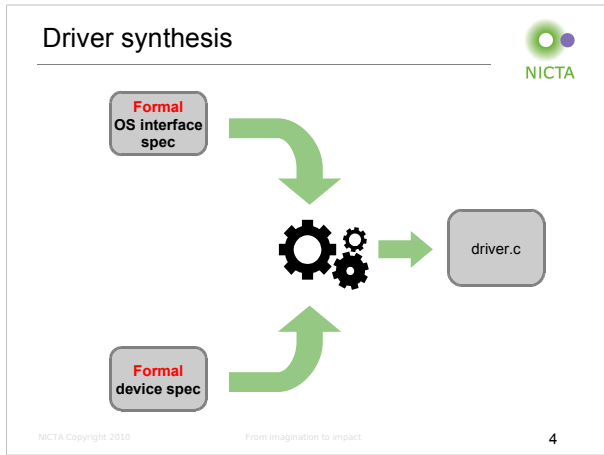
NICTA

# Challenges in Automatic Device Driver Synthesis

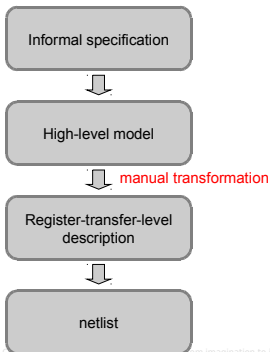
Leonid Ryzhyk Adam Walker

Australian Government  
Department of Broadband, Communications  
and the Digital Economy  
Australian Research Council

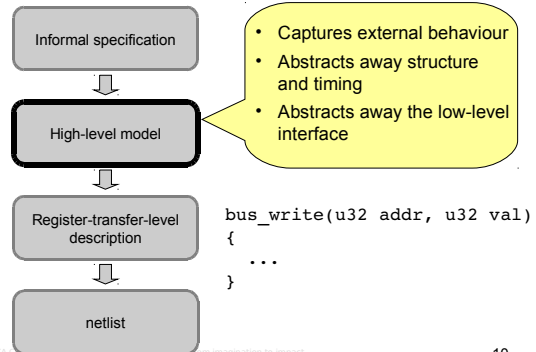
NICTA Funding and Supporting Members and Partners  
ANU UNSW Industry & Investment



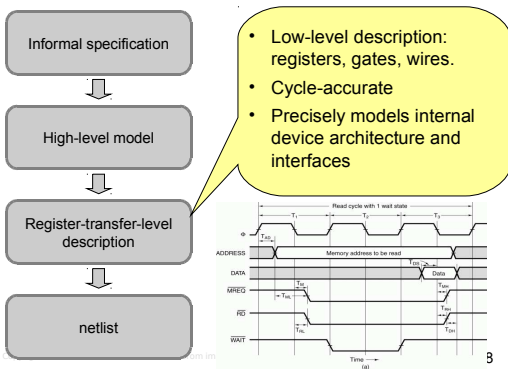
## Hardware design workflow



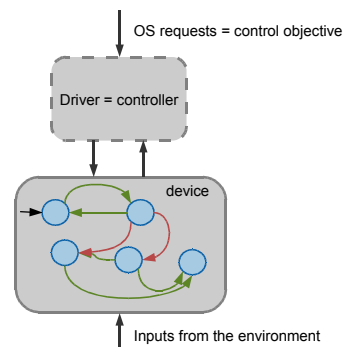
## Hardware design workflow



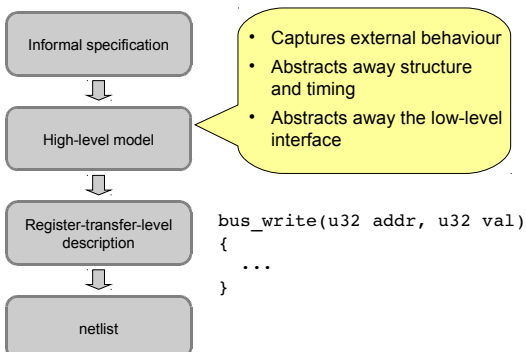
## Hardware design workflow



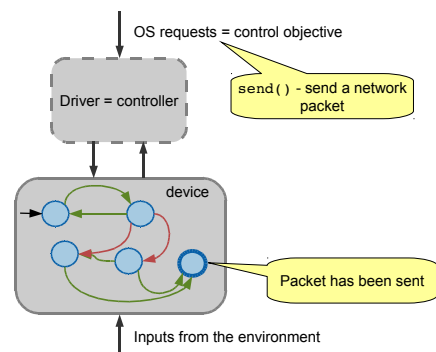
## Driver synthesis as controller synthesis



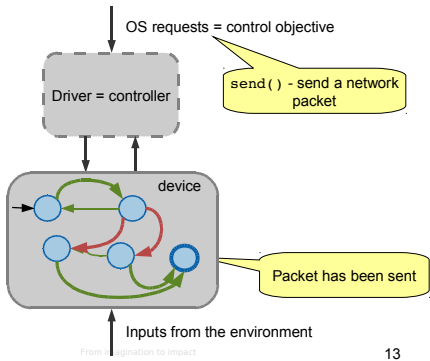
## Hardware design workflow



## Driver synthesis as controller synthesis



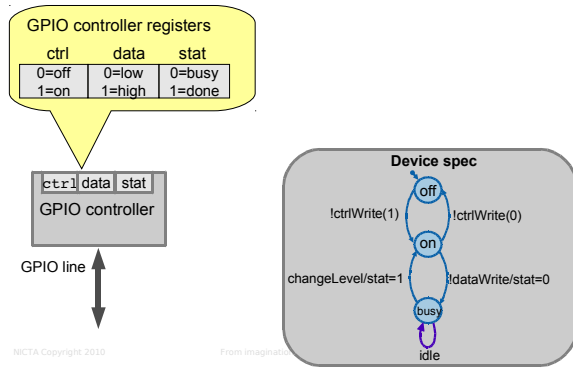
# Driver synthesis as controller synthesis



NICTA Copyright 2010

From Imagination to Impact

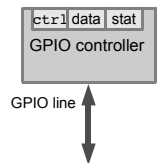
# Example: GPIO controller



NICTA Copyright 2010

From Imagination to Impact

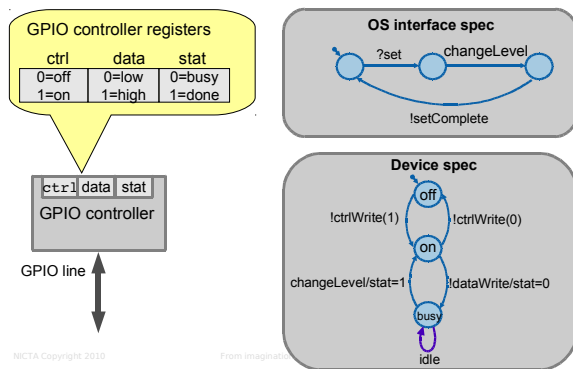
# Example: GPIO controller



NICTA Copyright 2010

From Imagination to Impact

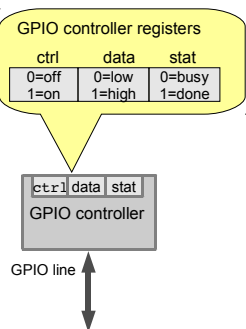
# Example: GPIO controller



NICTA Copyright 2010

From Imagination to Impact

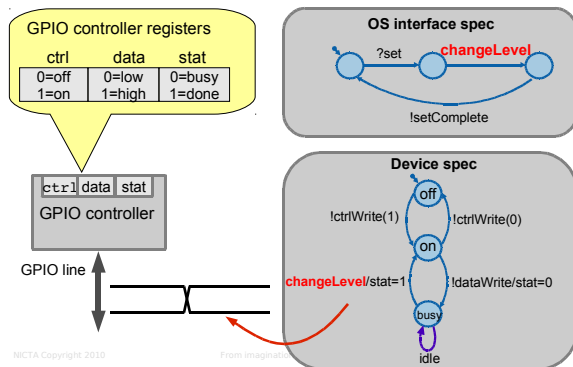
# Example: GPIO controller



NICTA Copyright 2010

From Imagination to Impact

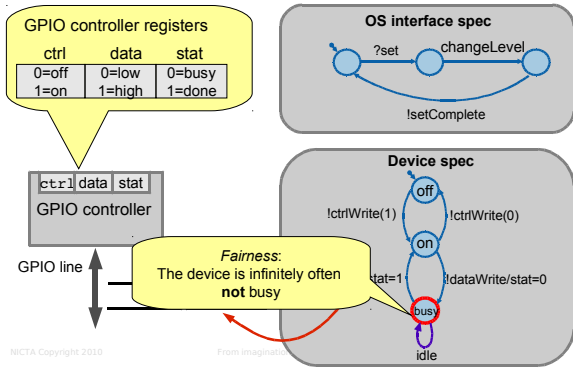
# Example: GPIO controller



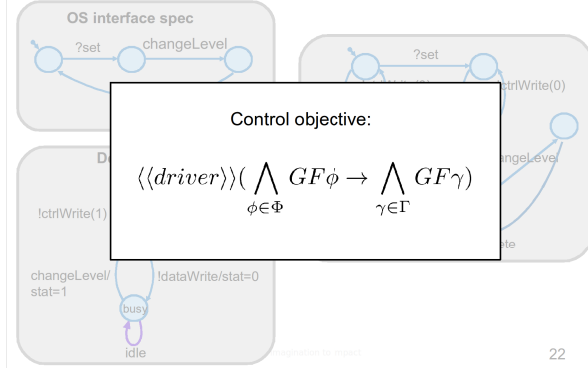
NICTA Copyright 2010

From Imagination to Impact

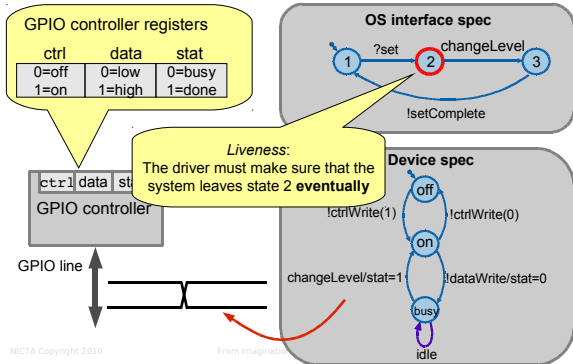
### Example: GPIO controller



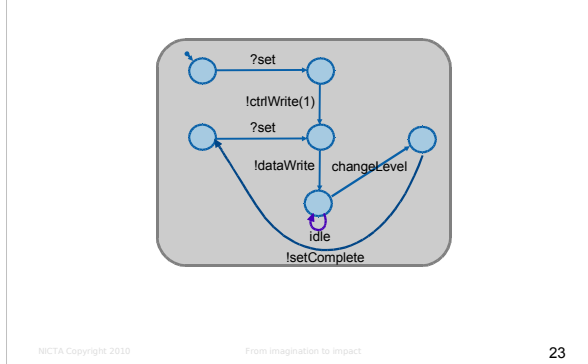
### Example: GPIO controller



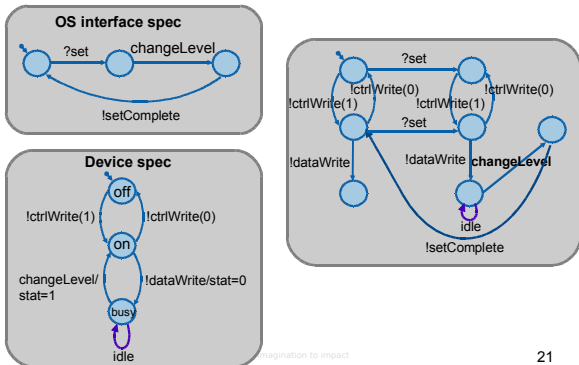
### Example: GPIO controller



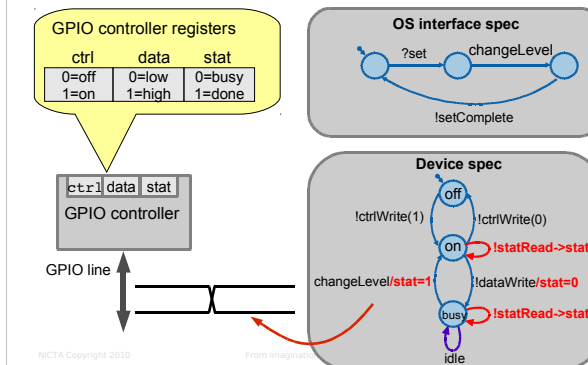
### Example: GPIO controller



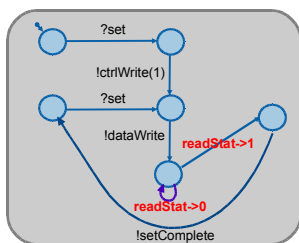
### Example: GPIO controller



### Example: GPIO controller



## Example: GPIO controller



NICTA Copyright 2010

From imagination to impact

25

## Challenges



- Challenge #3: Driver synthesis for devices with the DMA capability
- Challenge #4: Efficient C code generation
- Challenge #5: Synthesising concurrent implementations
- Challenge #6: Optimal synthesis
- Challenge #7: Verifying synthesised drivers

NICTA Copyright 2010

From imagination to impact

28

## Challenges



- Challenge #1: State Explosion
  - Real devices contain dozens of registers, internal FIFOs, support multiple modes of operation and failure modes etc.
  - An efficient synthesis algorithm must rely on abstraction

[1] Henzinger, Jhala, Majumdar, *Counterexample-guided control*, 2003  
 [2] de Alfaro, Roy, *Solving games via three-valued abstraction Refinement*, 2007

NICTA Copyright 2010

From imagination to impact

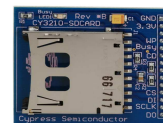
26

## Results



Proof-of-concept implementation:

- Successfully synthesised drivers for real devices based on manually written specifications:
  - Asix AX88772 USB-to-Ethernet adapter
  - Ricoh R5C822 SD host controller
- Performance on par with manually developed drivers



NICTA Cop

ion to impact

29

## Challenges



- Challenge #2: Partial Information
  - The driver cannot directly observe device states and state transitions
  - Partial observations can be obtained by reading register values
  - Device state can change between two register accesses

[1] Chatterjee, Doyen, Henzinger, Raskin, *Algorithms for omega-regular games with imperfect information*, 2007  
 [2] Dimitrova, Finkbeiner, *Abstraction refinement for games with incomplete information*, 2008

NICTA Copyright 2010

From imagination to impact

27

## Conclusions



- Automatic driver synthesis is a radical approach to improving OS reliability
- Driver synthesis poses interesting challenges in the area of controller synthesis algorithms

NICTA Copyright 2010

From imagination to impact

30

