

# Toward a TLM to RTL refinement: a formal approach

Jean-François Le Tallec<sup>1,2</sup>

Julien DeAntoni<sup>1</sup>


<sup>1</sup>AOSTE team, Université de Nice Sophia Antipolis, INRIA Sophia Antipolis Méditerranée,  
06902 Sophia Antipolis, France

<sup>2</sup>Scaleo Chip, 1681 route des Dolines, Sophia Antipolis, 06560 Valbonne, France

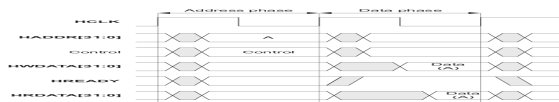
October 27, 2009

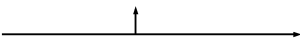


# Problematic

TLM: 

RTL:

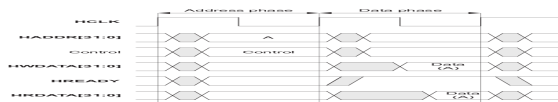


TLM: 

How to progressively refine the communication ?

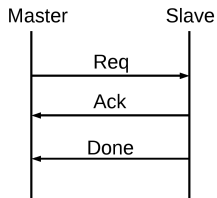
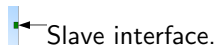
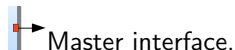
- What kind of information is needed ?
- How to express logical relations ?

RTL:



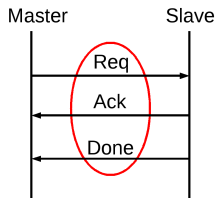
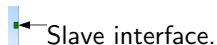
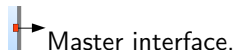
# Our approach

- Communication centric approach
  - Interface definition
  - Synchronisation via formal language
  - Behavior refinement via clock constraints



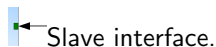
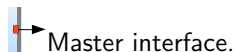
# Our approach

- Communication centric approach
  - Interface definition
  - Synchronisation via formal language
  - Behavior refinement via clock constraints

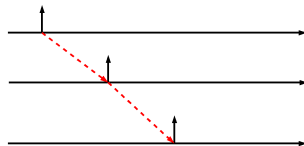
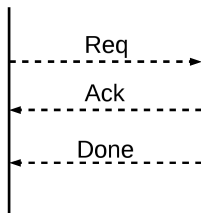


# Our approach

- Communication centric approach
  - Interface definition
  - Synchronisation via formal language
  - Behavior refinement via clock constraints

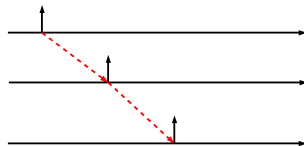
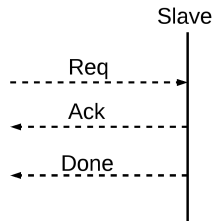
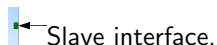
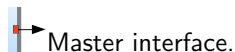


Master



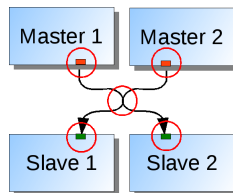
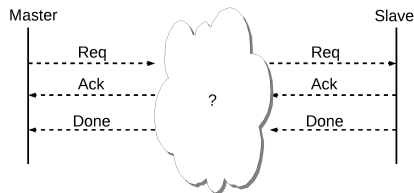
# Our approach

- Communication centric approach
  - Interface definition
  - Synchronisation via formal language
  - Behavior refinement via clock constraints



# Our approach

- Communication centric approach
  - Interface definition
  - Synchronisation via formal language
  - Behavior refinement via clock constraints

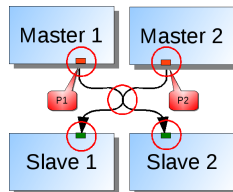
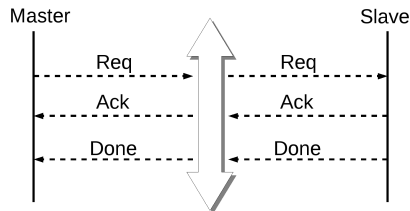


$$\begin{aligned}
 \bigcup_{i \leq M} Req_{Slave_i} & \equiv \bigcup_{j \leq N} Req_{Master_j} \\
 \bigcup_{i \leq M} Ack_{Slave_i} & \equiv \bigcup_{j \leq N} Ack_{Master_j} \\
 \bigcup_{i \leq M} Done_{Slave_i} & \equiv \bigcup_{j \leq N} Done_{Master_j} \\
 \forall i \leq N, j \leq N, i \neq j, Req_{Master_i} & \# Req_{Master_j} \\
 \forall i \leq M, j \leq M, i \neq j, Req_{Slave_i} & \# Req_{Slave_j} \\
 \bigcup_{i \leq M} Req_{Master_i} & \sim \bigcup_{j \leq M} Ack_{Master_j} \\
 \bigcup_{i \leq M} Ack_{Slave_i} & \sim \bigcup_{j \leq M} Done_{Slave_j}
 \end{aligned}$$



# Our approach

- Communication centric approach
  - Interface definition
  - Synchronisation via formal language
  - Behavior refinement via clock constraints



$$\begin{aligned}
 & \forall i \leq N, CFReq_{Masteri} \sqsubseteq \\
 Req_{Masteri} &= \bigcup_{j, P(i) \geq P(j)} RealReq_{Masterj} \\
 & \forall i \leq N, j \leq N, \\
 (Req_{Masteri} - CFReq_{Masteri}) & \sqsubseteq DelayedReq_{Masteri} \\
 & \forall i \leq N \\
 RealReq_{Masteri} & \sqsubseteq CFReq_{Masteri} + DelayedReq_{Masteri} \\
 & \forall i \leq N, RealReq_{Masteri} \sqsubseteq Ack_{Masteri} \\
 & \forall i, j \leq N, i \neq j, DelayedReq_{Masteri} \# RealReq_{Masterj}
 \end{aligned}$$

# For more explanation ...

... See you at poster session.

