

---

# **Verification of Synchronous Realizability of Interfaces from Timing Diagram Specifications**

---

Abdelhalim El Aboudi, El Mostapha Aboulhamid, Eduard Cerny

Université de Montréal

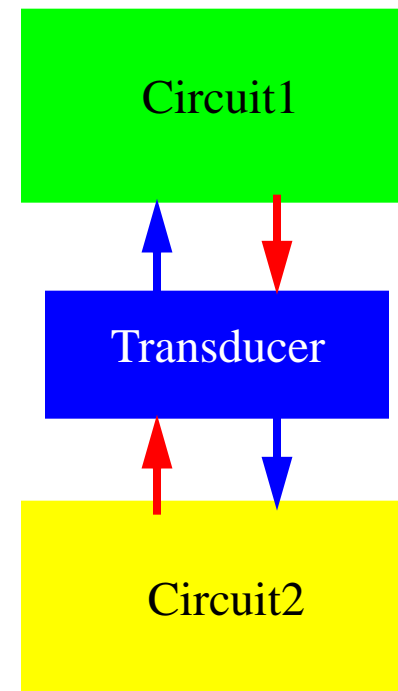
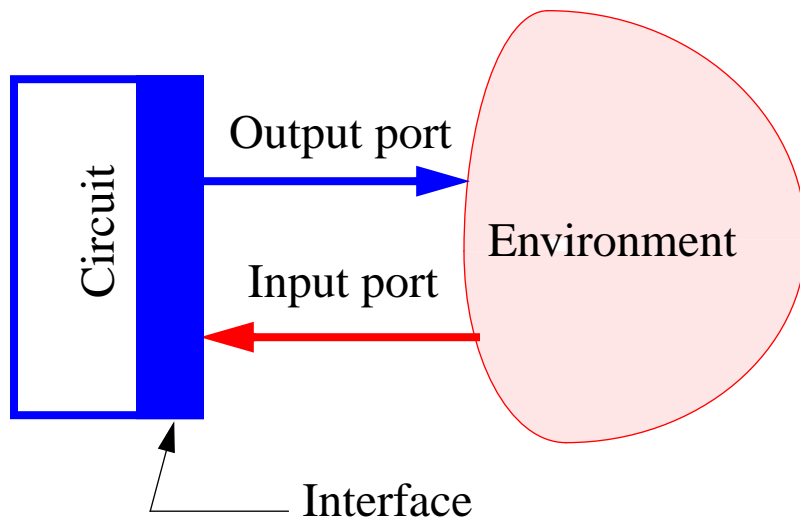
Date : 14/12/98



# Objectives

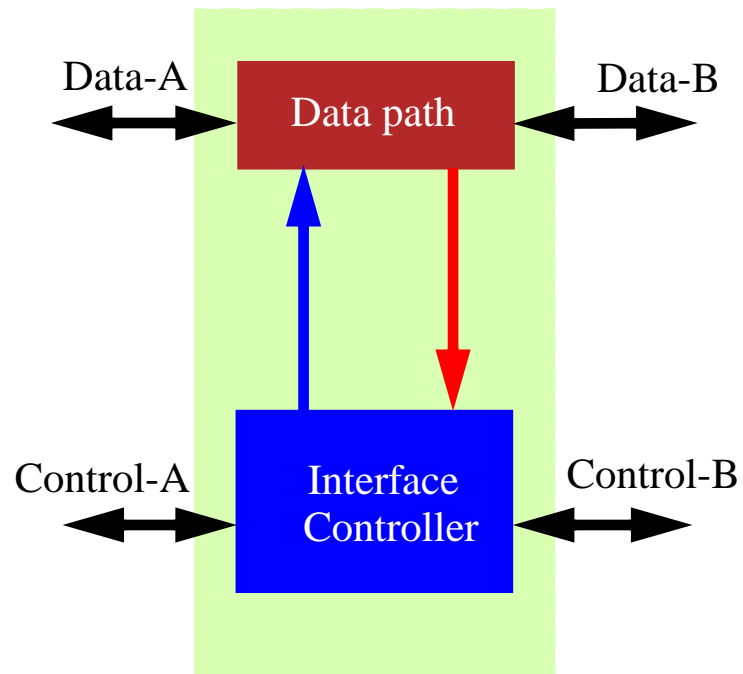
---

- Synthesis of the interface or the transducer specified by a Timing Diagram



# *Interface controllers*

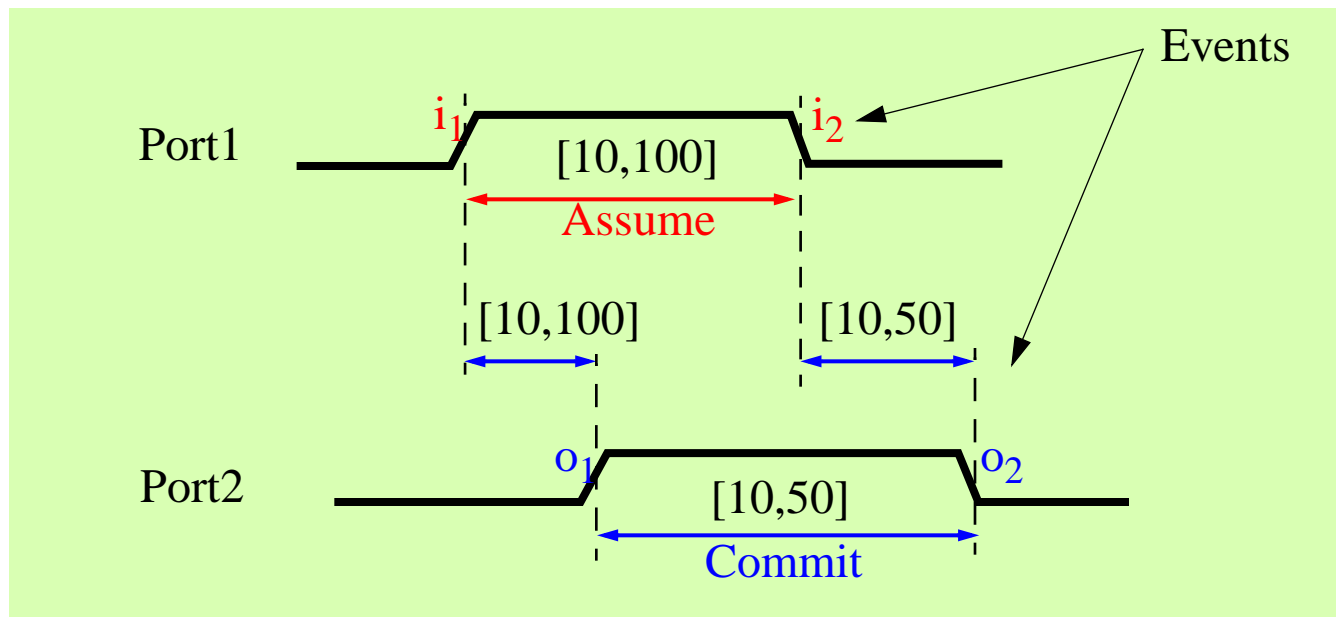
---



## *Synthesis:*

- 👉 Establishing the data path
- 👉 Commanding data transfer
- 👉 Synchronizing data transfer by satisfying timing constraints

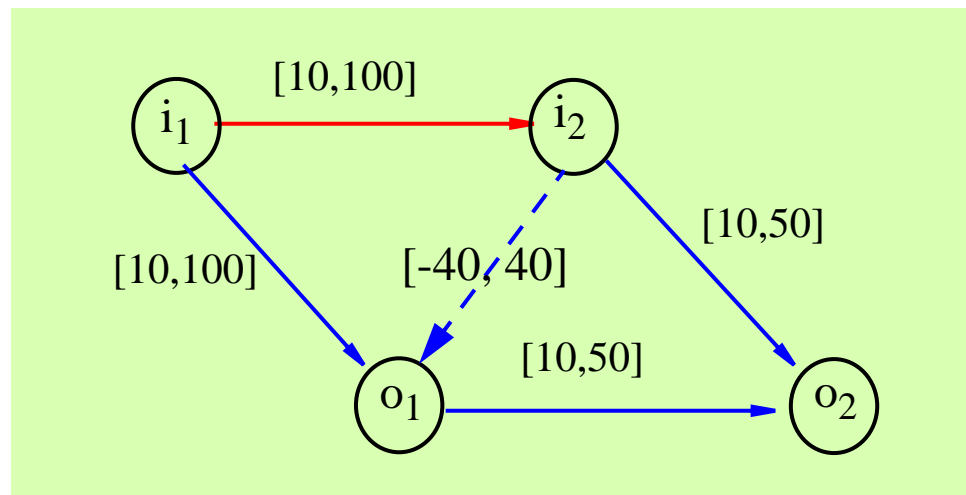
# TD illustration



⇒ Problem: Exact delay unknown between  $i_1$  and  $i_2$

# Event graph

---

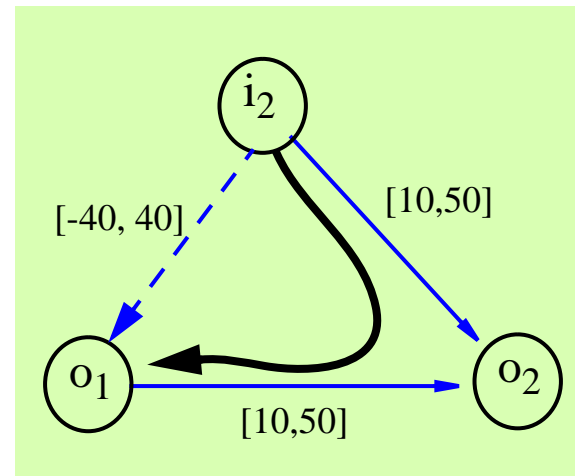
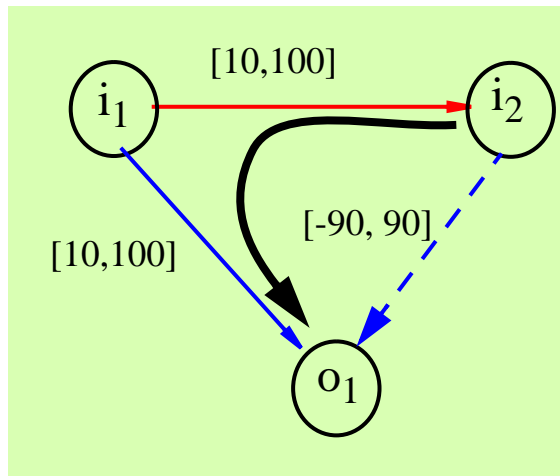


Implicit commit constraint computed by a maximum separation algorithm

# Timing Analysis of TDs

↳ Maximum separation time:

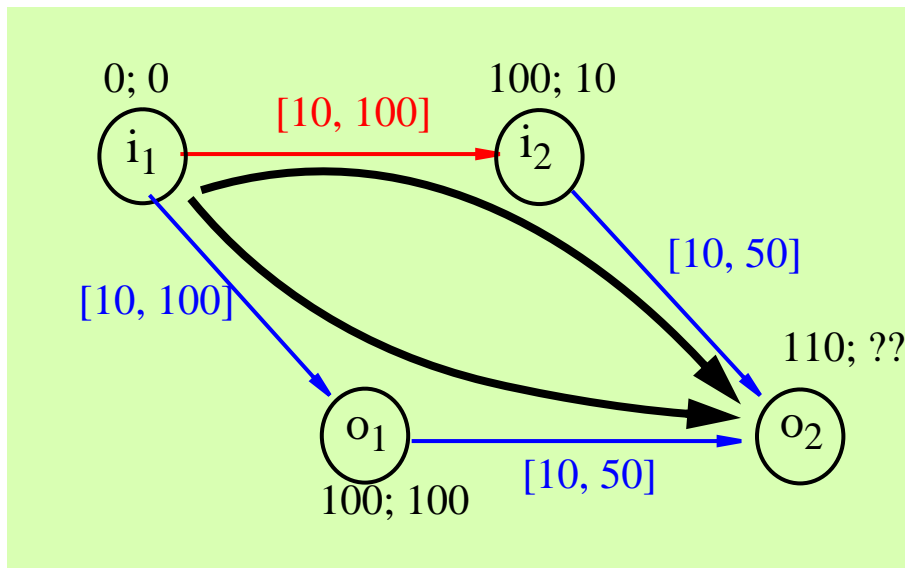
$$s_{ij} = \max(t(e_j) - t(e_i))$$



⇒ nature of events not relevant

# Consistency

→ Find a solution to the system of constraints



Consistency

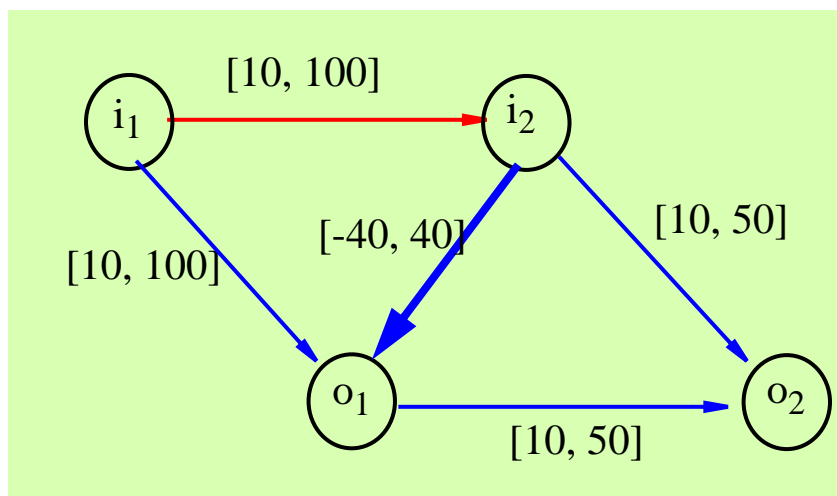
⇒ absolute schedule

⇒ Consistency insufficient

⇒ Introduction of new cons.

# Realizability of TDs

- ⇒ Existence of  $\{t(o_i)\}$  for all possible assignment by the environment to  $\{t(i_k)\}$  respecting the specification.
- ⇒  $\forall (t(i_2)-t(i_1)) \in [10,100], \exists (t(o_1), t(o_2))$  such that all constraints are satisfied.



*Solution:*

$$t(o_1) = \min(t(i_1)+100, t(i_2)+40)$$

$$t(o_2) = \min(t(i_2)+50, t(o_1)+50)$$



# Causal functions

---

$EG = (E, C)$ ,  $C = A \cup K$ .  $|A| = m$ .

$O_s$  output events source for constraints in  $A$ ,  $|O_s| = q$ .

**Definition: (causal function)**

$f: (R^+)^n \rightarrow R^+$  is causal  $\Leftrightarrow$

$\forall \mathbf{x}^0 = (x_1^0, \dots, x_n^0) \in (R^+)^n$ ,  $\forall i \in \{1, \dots, n\}$  we have either:

$$f(\mathbf{x}^0) \geq x_i^0, \text{ or}$$

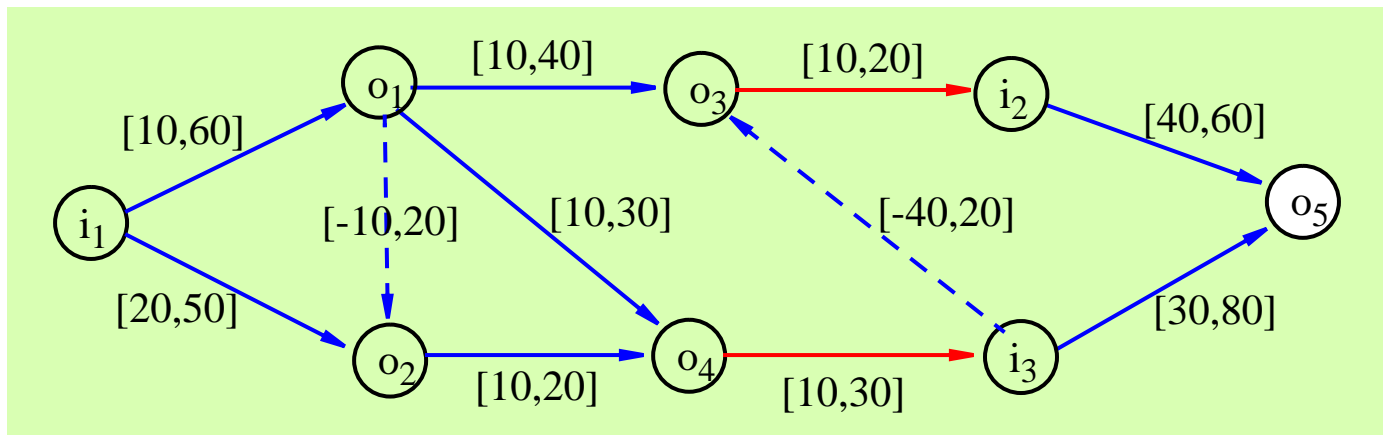
$$\forall x_i^1 \geq x_i^0 : f(x_1^0, \dots, x_i^1, \dots, x_n^0) = f(x_1^0, \dots, x_i^0, \dots, x_n^0).$$

**Definition:**  $h: O_s \rightarrow R^+$  causal if  $\exists q$  causal functions  $f_k: (R^+)^m \rightarrow R^+$  /  $\forall o_k \in O_s$ ,

( $k = 1, \dots, q$ ), we have  $h(o_k) = f_k(t(i_1), \dots, t(i_m))$ .

# Realizability

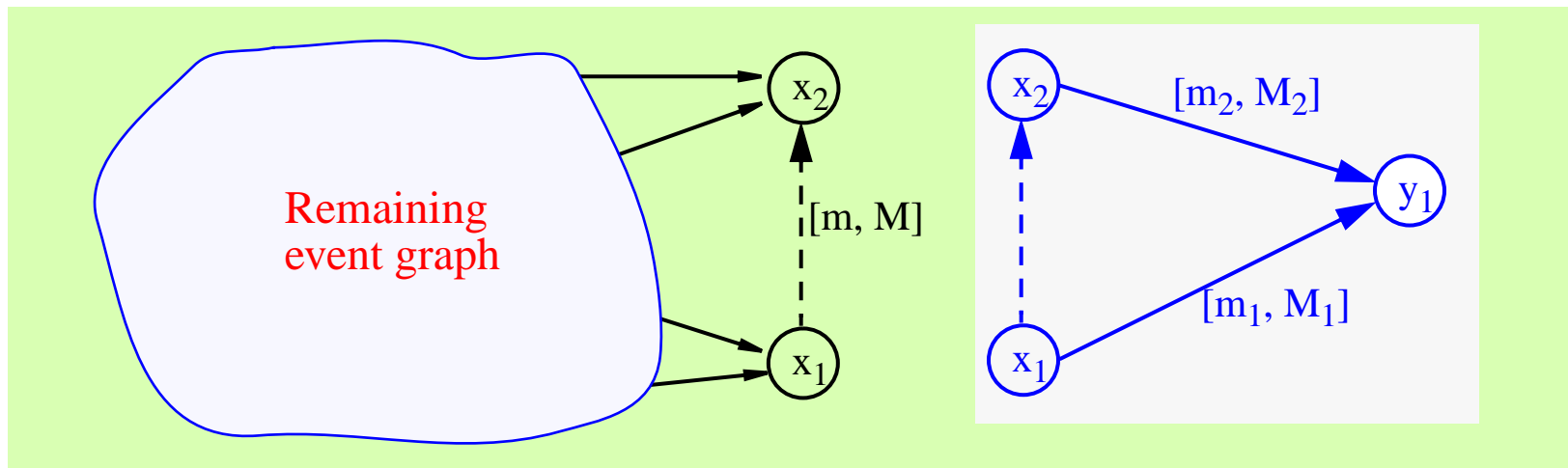
$\Rightarrow \exists h: O_s \rightarrow R^+ / \forall (t(i_1), \dots, t(i_m))$  from the space of inputs,  
 all constraints  $c_{ij}$  can be satisfied.



$$h : t(o_3) = t(i_1) + 100 \text{ and } t(o_4) = t(i_1) + 70$$

# Local consistency

- ⇒ Proceed in a reverse topological order.
- ⇒ Compute  $m$  and  $M$  over the event graph without node  $y_1$
- ⇒ If  $m_1 - M_2 \leq m \leq M \leq M_1 - m_2$  not satisfied then
- ⇒ Add dashed commit constraint if necessary
- ⇒ Remove the last node

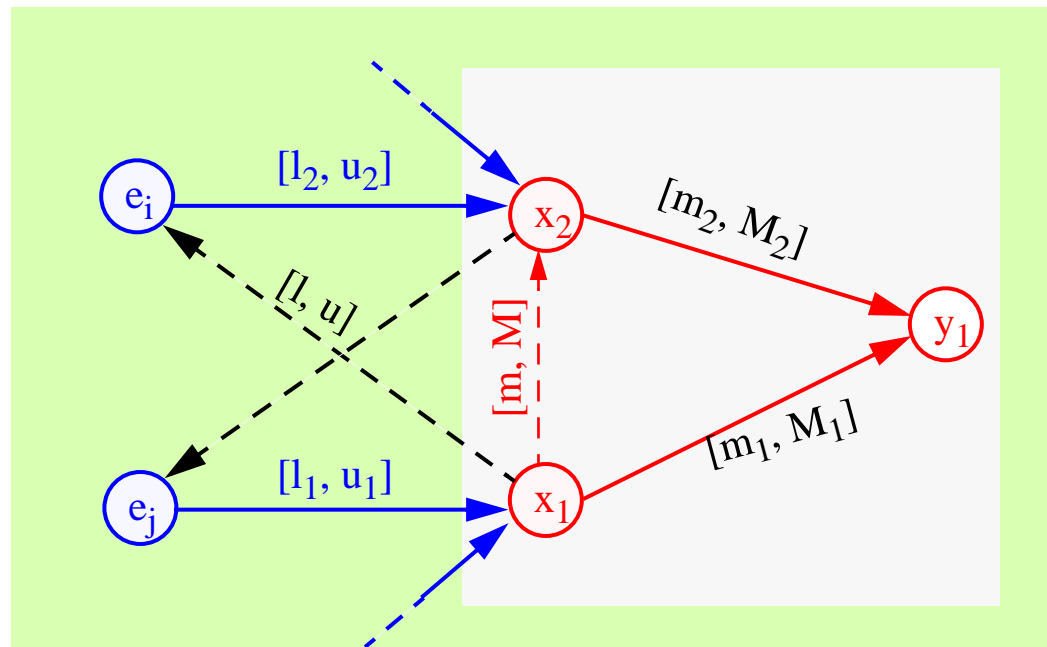


# Local consistency...

⇒ Case of assume constraint

Look backward for commit constraint with:

$$u+u_2 \leq M \text{ and } m \leq l+l_2$$



# Feasible clock periods

---

➤ **Objective:**

Finding a realizable clock period for a synchronous implementation

➤ **General conditions:**

- ✓ **All input events could be captured**
- ✓ **All output events could be generated**
- ✓ **All timing constraints should be satisfied**

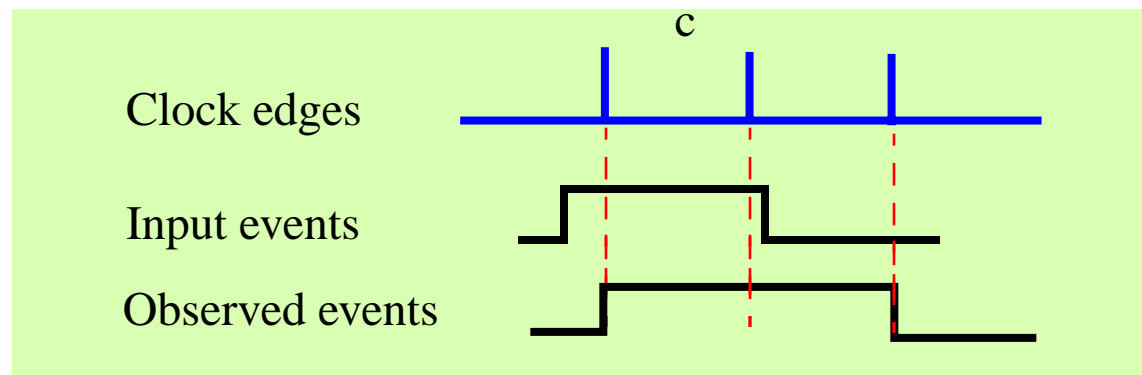
# Sampling rules

## ► Sampling Input ports:

the observed input event occurs at  $t_s(e_i)$  such that:

$$t_s(e_i) = \left\lceil \frac{t(e_i)}{c} \right\rceil \cdot c$$

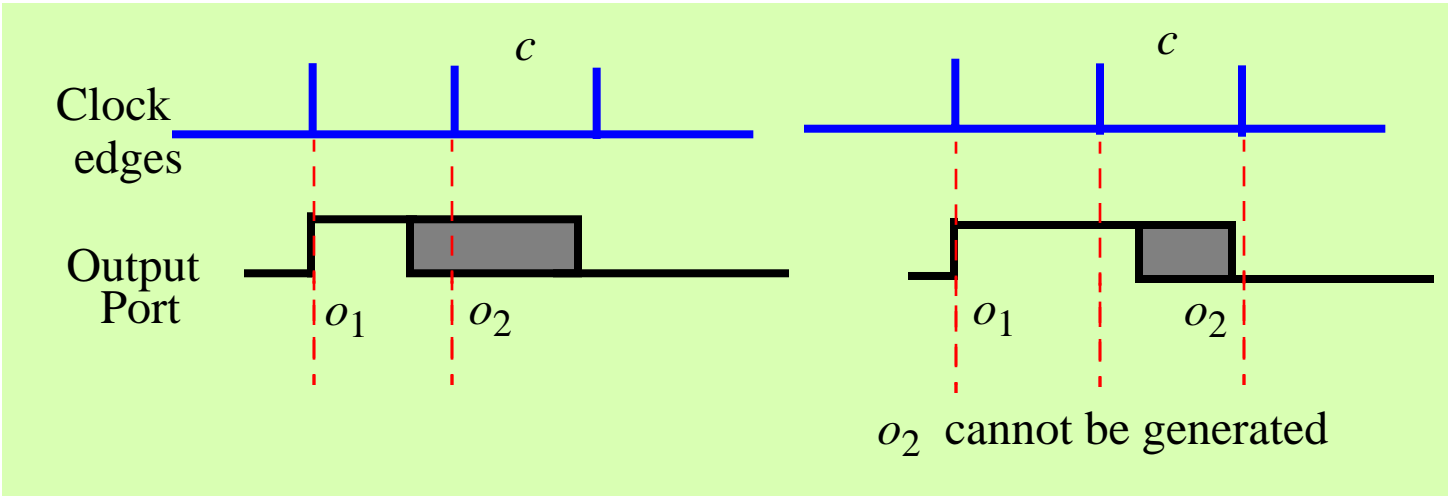
$$0 \leq t_s(e_i) - t(e_i) < c$$



► Four rules for correct sampling:

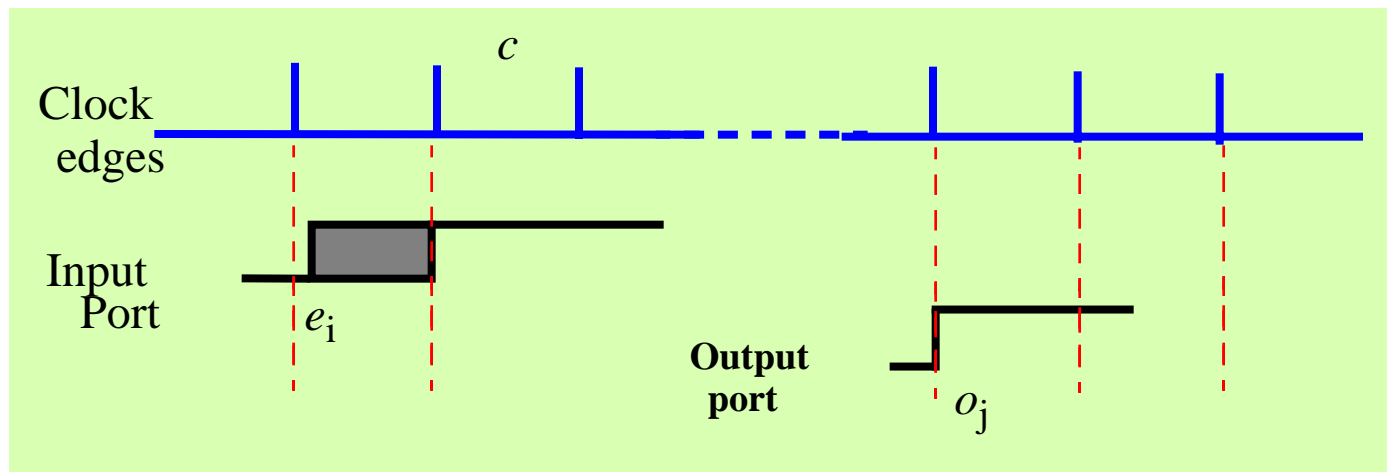
**Rule1:**

$$\left\lceil \frac{l_{ij}}{c} \right\rceil \cdot c \leq t_s(o_j) - t_s(o_i) \leq \left\lfloor \frac{u_{ij}}{c} \right\rfloor \cdot c$$



**Rule2:**

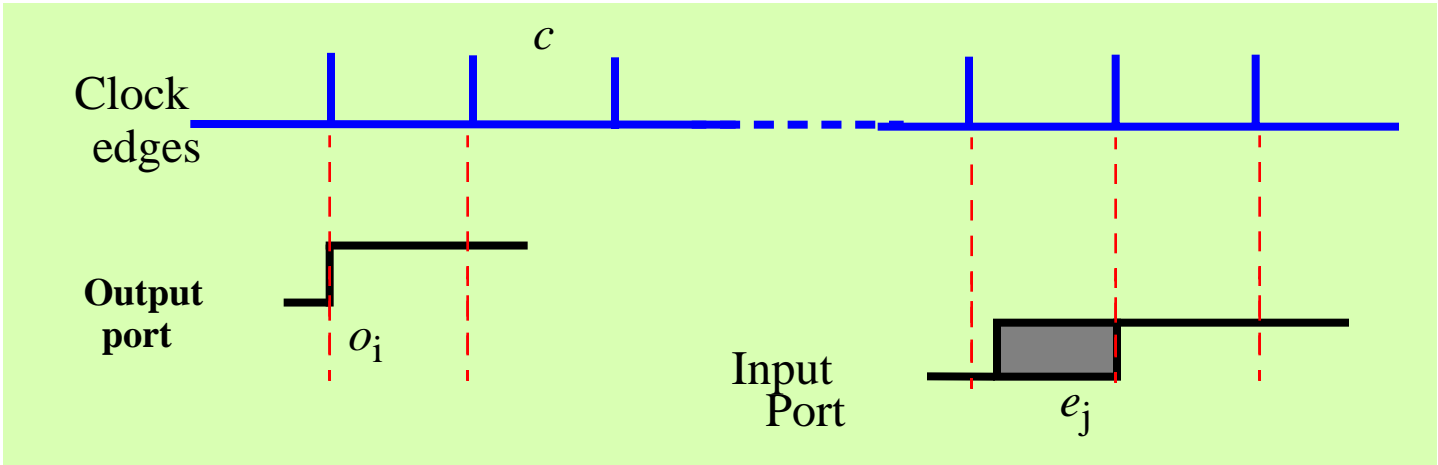
$$\left\lceil \frac{l_{ij}}{c} \right\rceil \cdot c \leq t_s(o_j) - t_s(e_i) \leq \left\lfloor \frac{u_{ij} - c}{c} \right\rfloor \cdot c$$





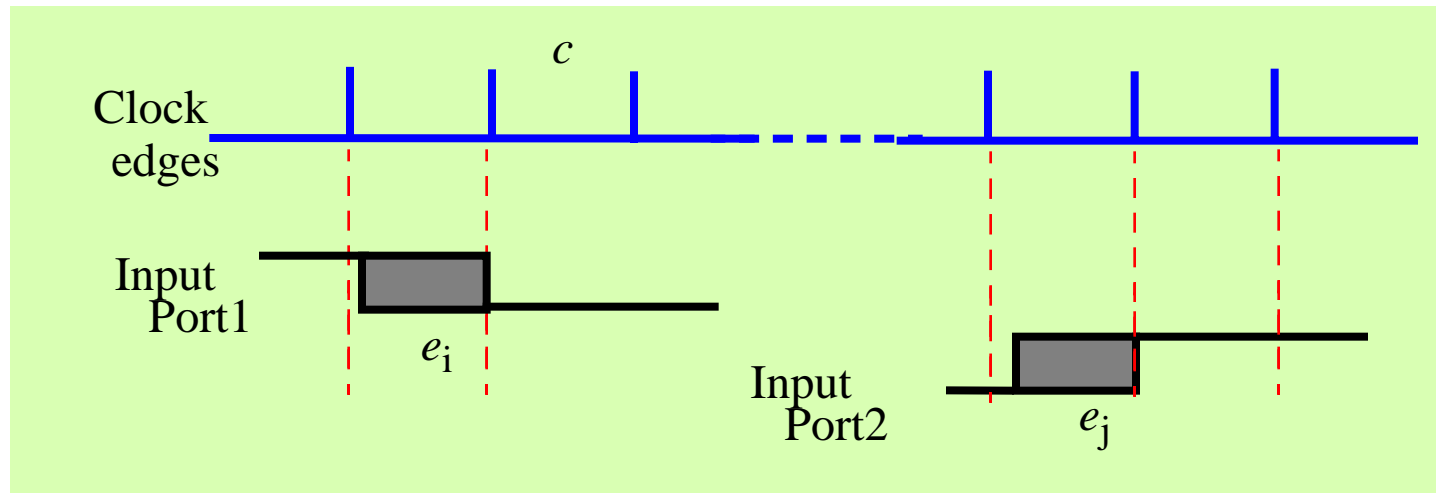
**Rule3:**

$$\left\lceil \frac{l_{ij}}{c} \right\rceil \cdot c \leq t_s(e_j) - t_s(o_i) \leq \left\lfloor \frac{u_{ij} + c}{c} \right\rfloor \cdot c$$



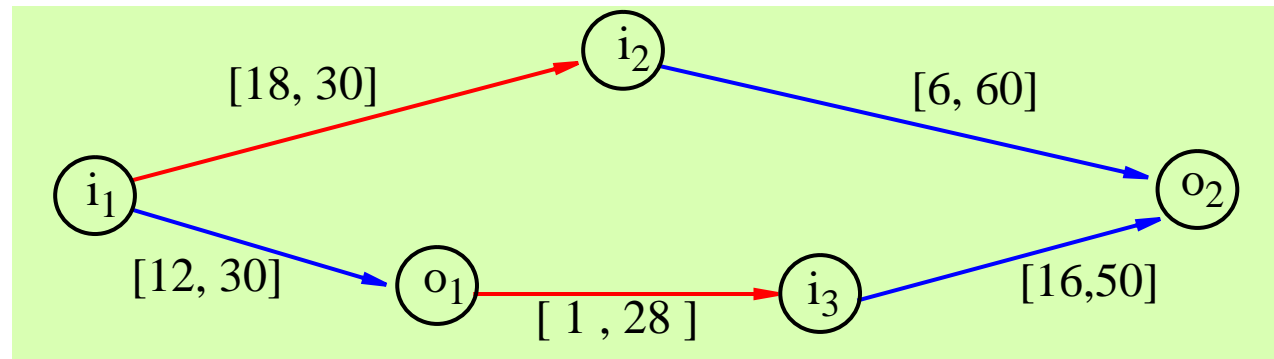
**Rule4:**

$$\left\lceil \frac{l_{ij} - c}{c} \right\rceil \cdot c \leq t_s(e_j) - t_s(e_i) \leq \left\lfloor \frac{u_{ij} + c}{c} \right\rfloor \cdot c$$

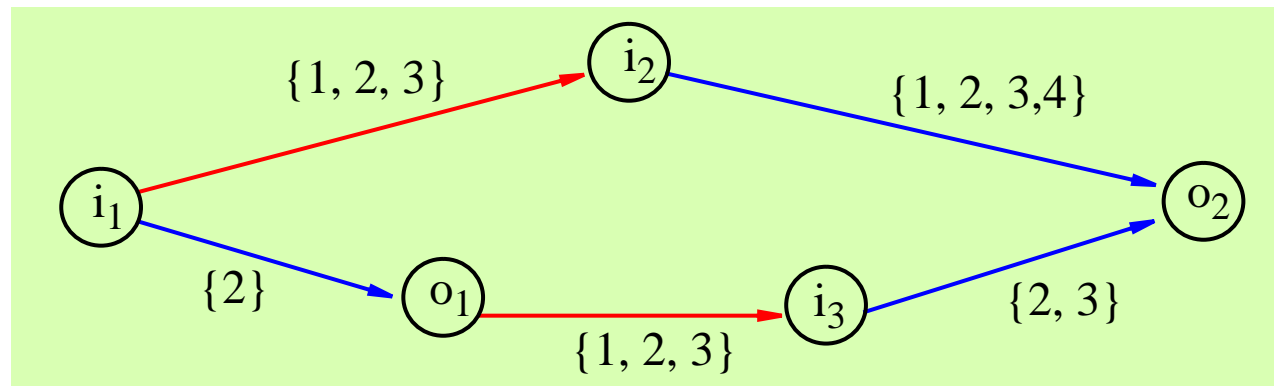


# Discrete time event graph

Given  
Event graph



Event Graph  
For  
 $c = 12$



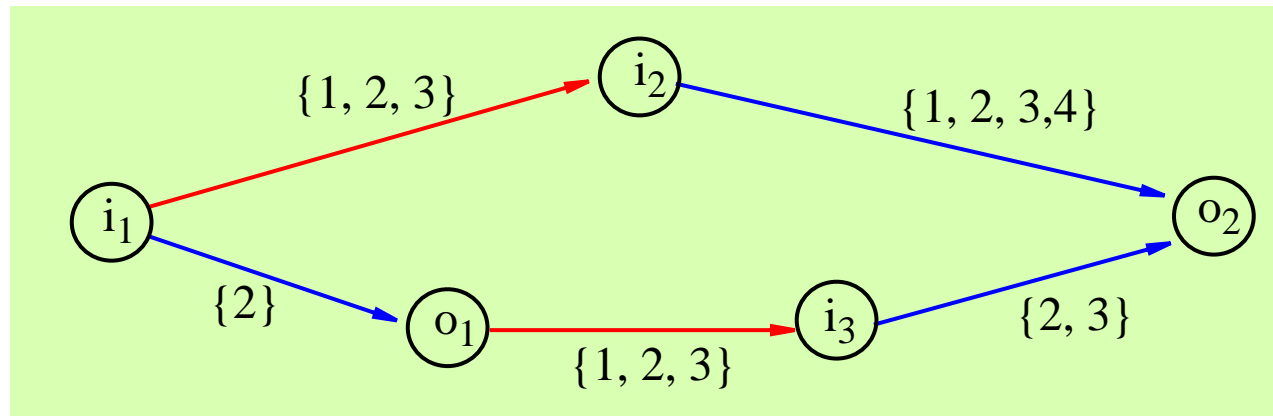
# Relative scheduling

We have:

$$ALAP(t(o_k)) = \min_{e_i \in \text{parents}(o_k)} (t(e_i) + u_{ik}) \quad \text{and}$$

$$ASAP(t(o_k)) = \min(ALAP(t(o_k)), \max_{e_i \in \text{parents}(o_k)} (t(e_i) + d_{ik}))$$

where  $d_{ik} = \max(0, l_{ij})$



# Relative scheduling...

---

*ALAP:*

$$t(o_1) = t_s(i_1) + 2$$

$$t(o_2) = \min(t_s(i_2) + 4, t_s(i_3) + 3)$$

*ASAP:*

$$t(o_1) = t_s(i_1) + 2$$

$$t(o_2) = \max(t_s(i_2) + 1, t_s(i_3) + 2)$$

# Conclusion

---

## Contributions

- ⇒ Finding feasible clock cycles for synchronous implementations
- ⇒ Verification of the synchronous realizability problem for TDs
- ⇒ Deriving a method for synthesizing interface controllers
  - Making an event graph locally consistent
  - Determining a realizable relative schedule for events
- ⇒ Generalization of relative scheduling

## Future work

- ⇒ Hierarchical TDs