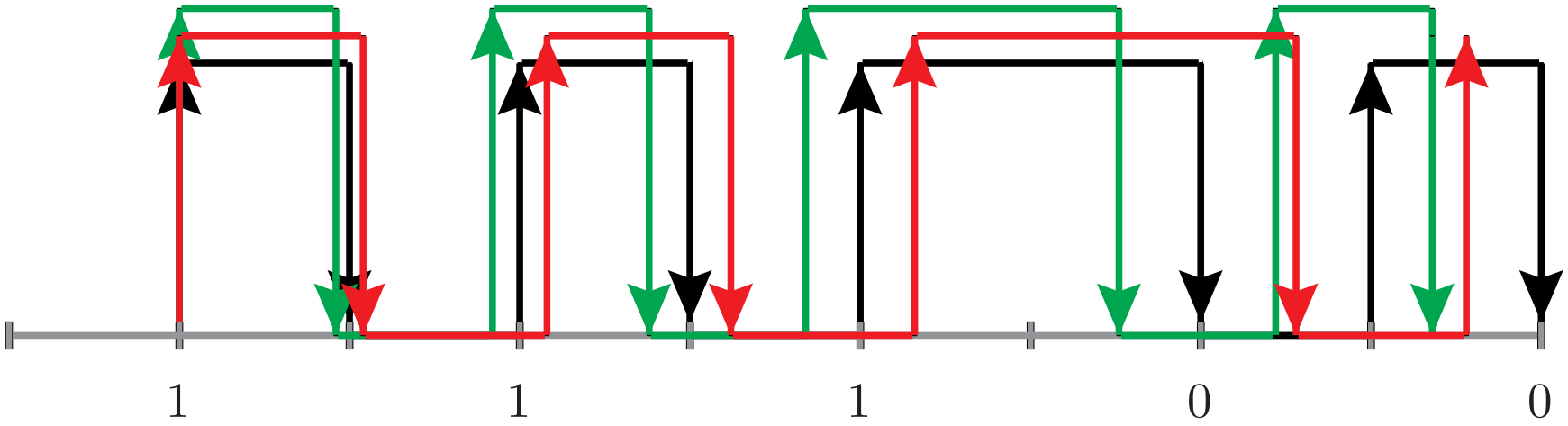
Formal Verification Using Timed Automata

Formal Verification Using Timed Automata



The Enhanced Easy Link (EEL) Protocol

- Philips standard
- Rules for sending control information between components of Philips audio systems
- Timing is essential part of the protocol
- Philips uses one processor chip for several tasks
- Philips allows high tolerances on timing
- What is the maximal timing tolerance so that the protocol is correct?

Informal Description of EEL

- Manchester Encoding of bit strings

- Receiver must recognize first bit of a new message:
  bus voltage low when no message is sent, first bit of a message must be 1
- Receiver cannot recognize downgoing edges due to hardware restrictions:
  - receiver only takes into account upgoing edges
  - bit strings must be $5 + 8n$ bits long for some $n \in \mathbb{N}$.
- Receiver does not know length of bit string:
  receiver stops decoding after certain time has passed since last upgoing edge
- Receiver should not glue together two messages:
  waiting time before last upgoing edge and the first upgoing edge.

- Receiver keeps track of last received bit and the time elapsed since last upgoing edge.

Informal Description of EEL

- EEL is a bus protocol: all components are connected to one cable
- Uses broadcast communication

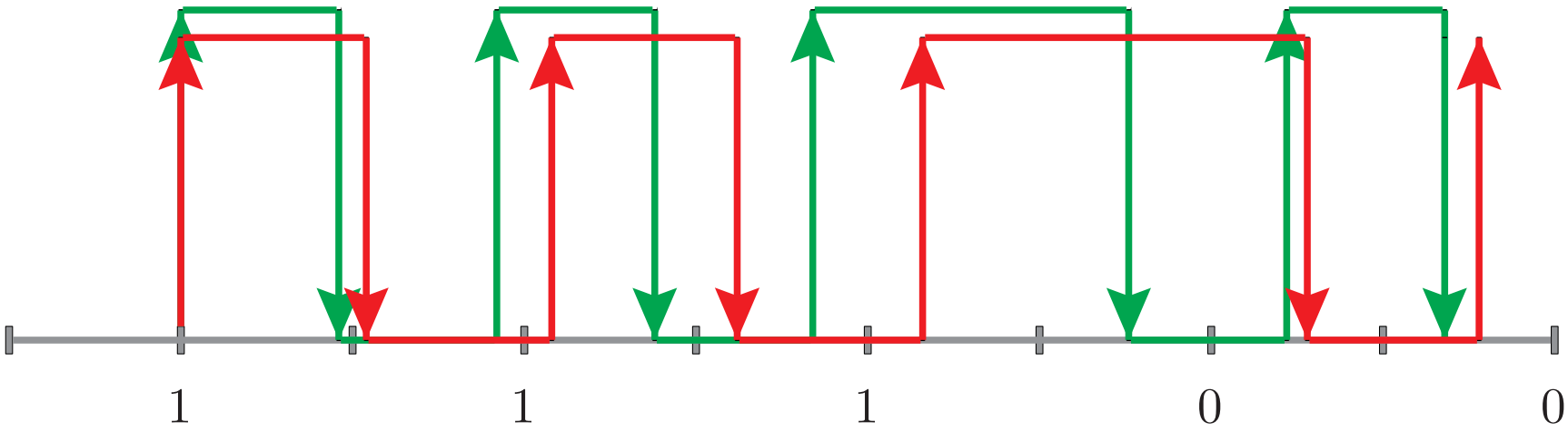
- Collision: two senders start to send at the same time
  - each sender must check whether it is the only one that is sending
  - if bus is on 1, while it sends 0, then it stops sending
  - bus level check:
    (A) just before every upgoing edge,
    (B) at 25% and 75% of a bit slot when sending a 0

Informal Description of EEL

- One bit slot equals 888 microseconds
- EEL has high timing tolerance:
  - Timing of edges: $\pm 5\%$
  - Bus level check (A): $\pm$ 20 microseconds
  - Bus level check (B):$\pm$ 22 microseconds

Correctness of EEL

- In a former version, the bus level check (A) was required only before the first edge of a message
- Bad scenario: fast sender, slow sender, perfect receiver:



- This version of the protocol was implemented and sold!

Correctness of EEL

- Philips is interested in the maximal time tolerance
- In 1994, Griffioen (G94) proved the correctness of EEL for all time tolerances less than $\pm 5.35\%$ using *formal methods*

(G94) Griffioen: Analysis of an Audio Control Protocol with Bus Collision. 1994

Formal Methods

- Writing formal specifications instead of natural language specifications
- May clarify unclear sections or ambiguities, because formalisms have a formally defined semantics
- Formalisms are easy to read
- One may prove properties about the specifications, e.g., the correctness

Correctness of EEL

- Philips is interested in the maximal time tolerance
- In 1994, Griffioen (G94) proved the correctness of EEL for all time tolerances less than $\pm 5.35\%$ using *formal methods*
- Used *timed automata* as a formalism
- Proved the correctness of EEL *without tool support*

(G94) Griffioen: Analysis of an Audio Control Protocol with Bus Collision. 1994

Correctness of EEL

- Philips is interested in the maximal time tolerance
- In 1994, Griffioen (G94) proved the correctness of EEL for all time tolerances less than $\pm 5.35\%$ using *formal methods*
- Used *timed automata* as a formalism
- Proved the correctness of EEL *without tool support*

- In 1996, Bengtsson et al (B96) *automatically* analyzed EEL using the tool *UPPAAL*

(G94) Griffioen: Analysis of an Audio Control Protocol with Bus Collision. 1994
(B96) Bengtsson et al: Verification of an Audio Protocol with Bus Collision Using UPPAAL. 1996.

Formal Methods

- Writing formal specifications instead of natural language specifications
- May clarify unclear sections or ambiguities, because formalisms have a formally defined semantics
- Formalisms are easy to read
- One may prove properties about the specifications, e.g., the correctness *automatically*
- For this, the formalisms needs to have according decidable decision problems