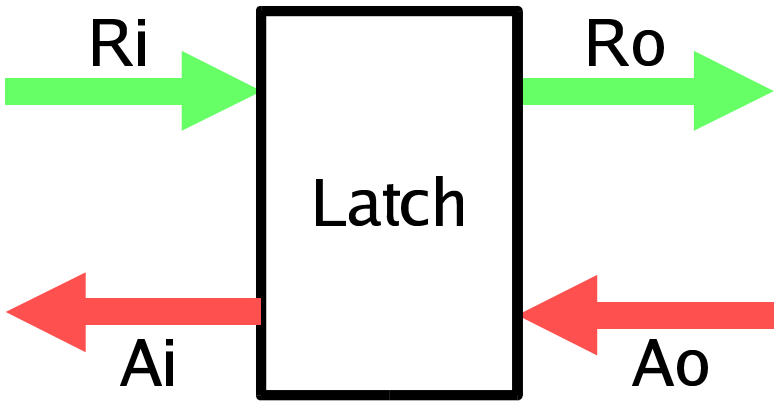
Early Output Logic with Anti-Tokens

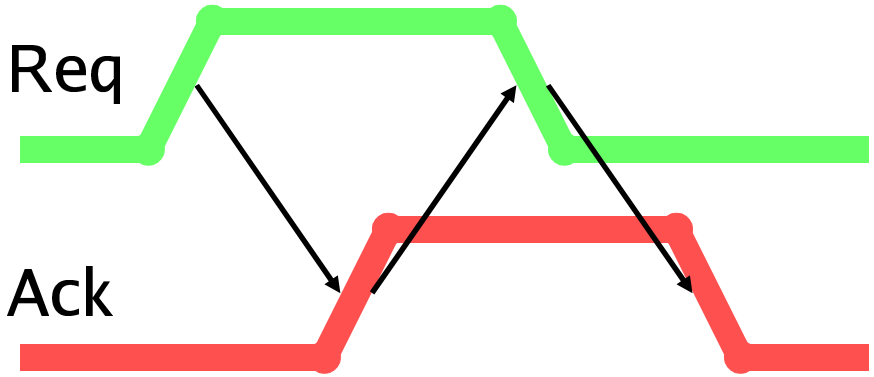
Charlie Brej, Jim Garside
APT Group
Manchester University

Outline

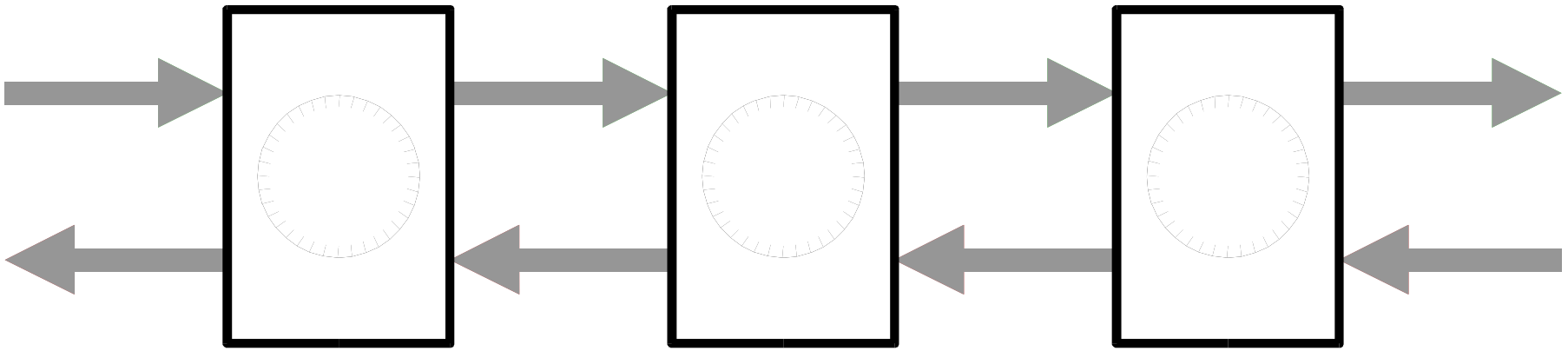
- Asynchronous Logic
  - DIMS (Delay Insensitive Minterm Synthesis)
- Early Output Logic
  - Guarding
- Anti-Tokens
  - Collisions
- Conclusions

Asynchronous Latch

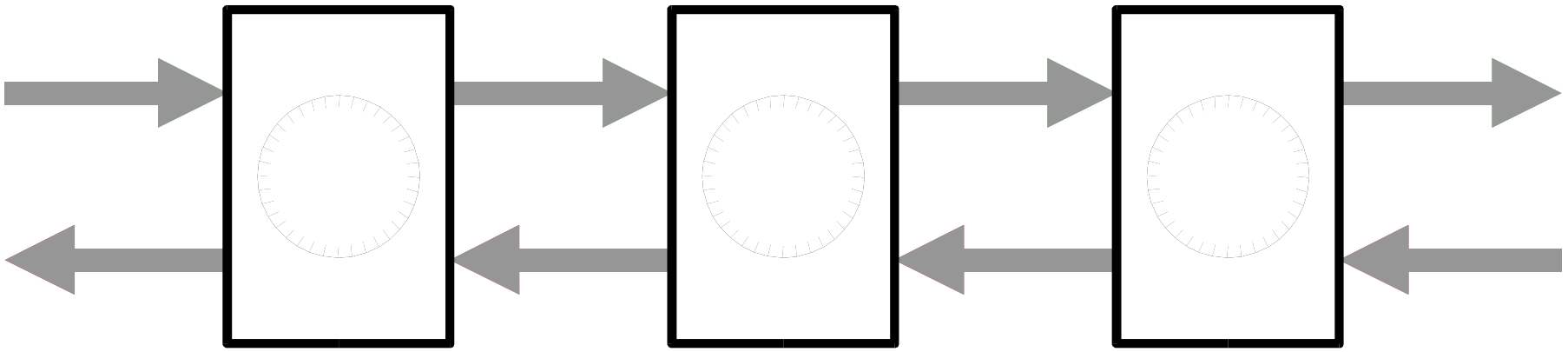




Asynchronous Pipeline

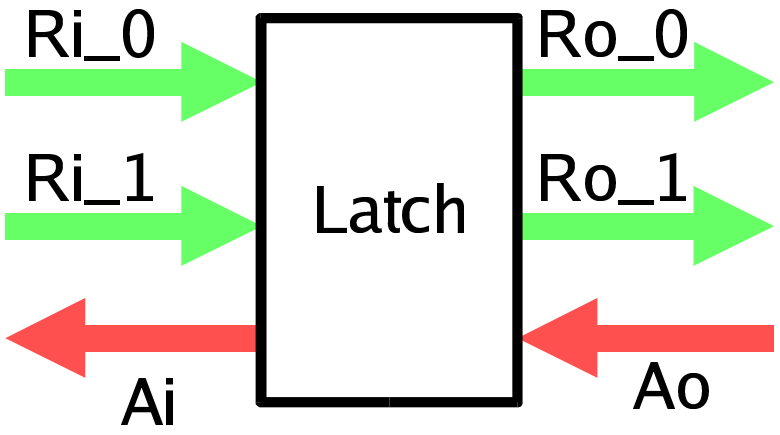


Asynchronous Pipeline Stall

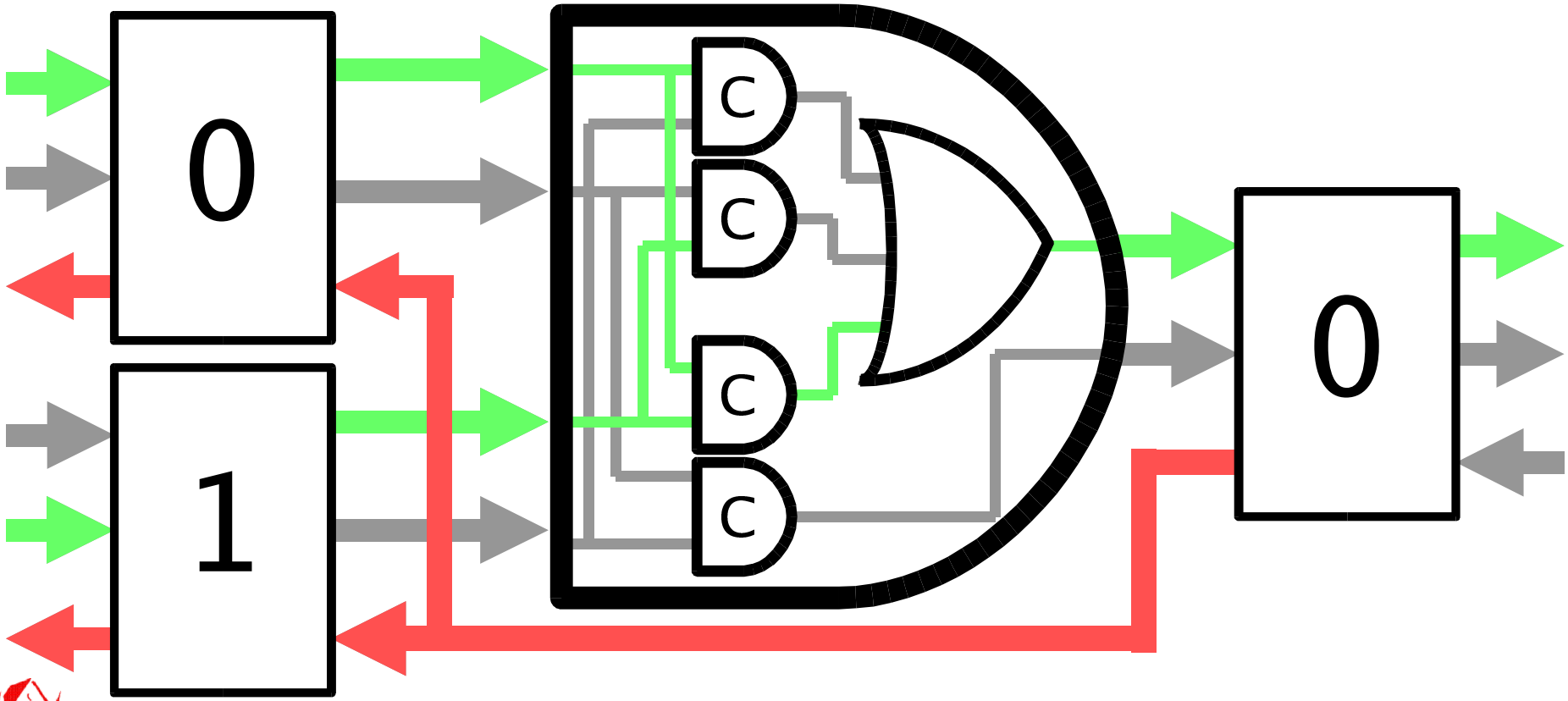


Dual-Rail Latch

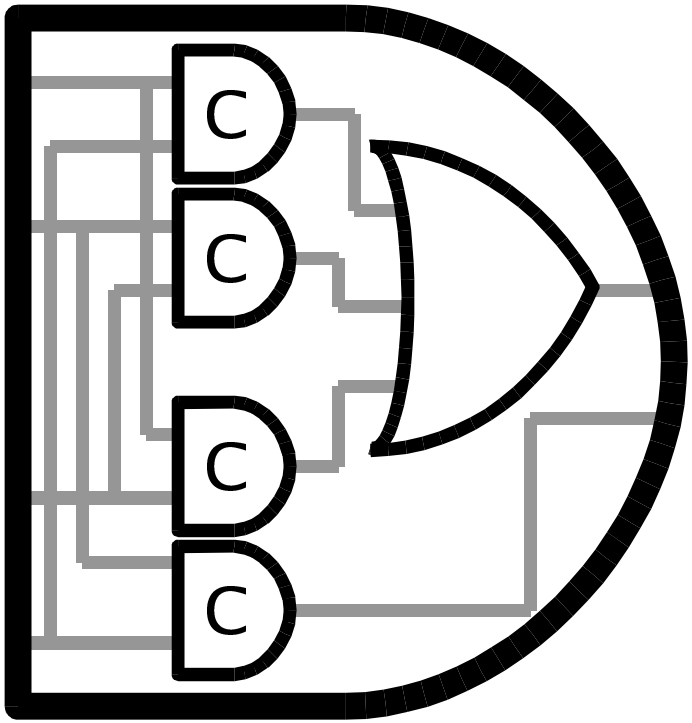
- Dual-Rail
  - 00 = ‘NULL’
  - 01 = 0
  - 10 = 1
  - 11 = Illegal
- Return to ‘NULL’

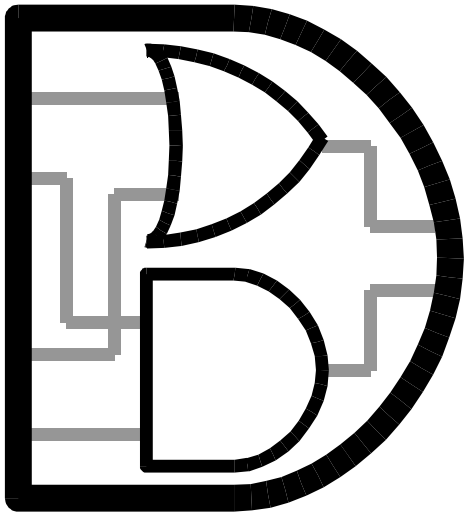


DIMS Logic



DIMS vs Early Output Logic





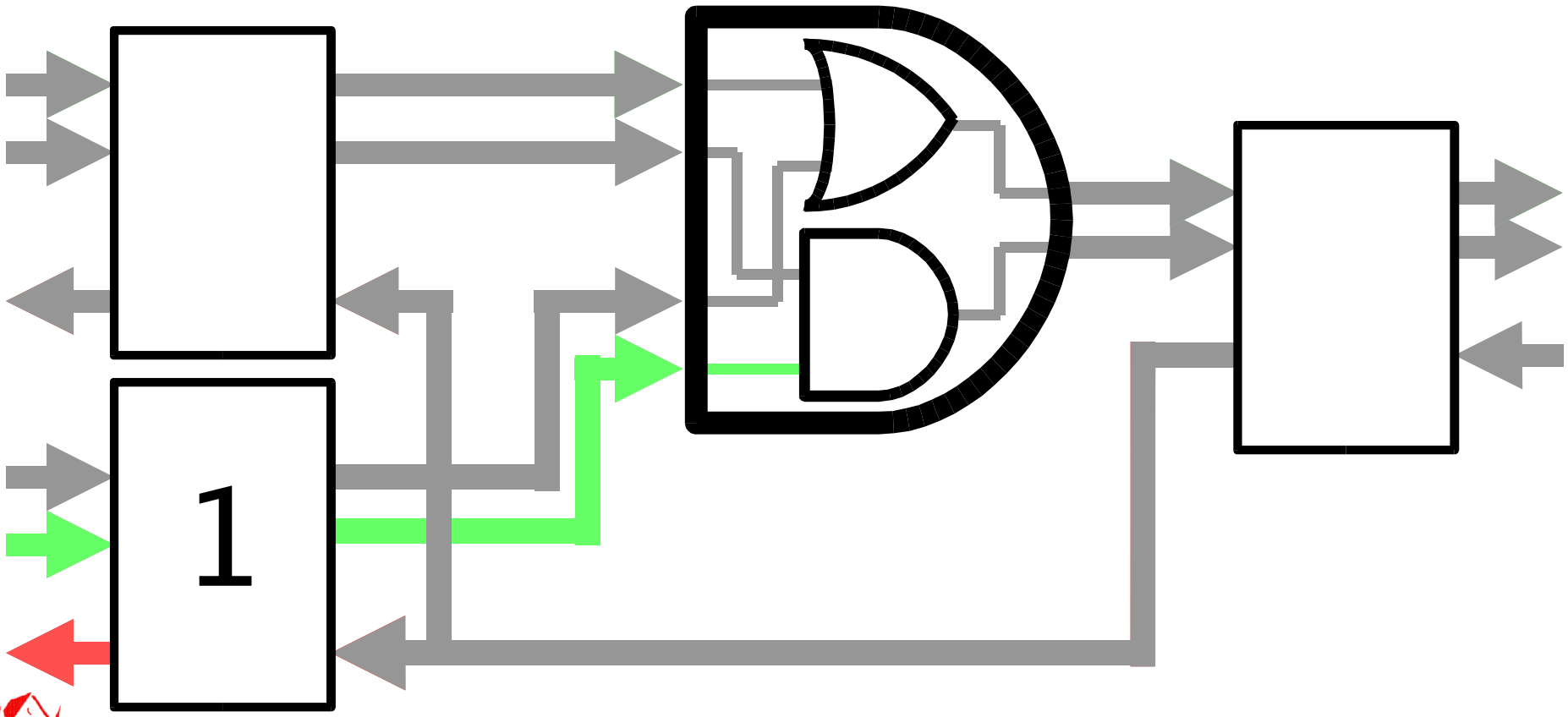
Size:48 transistors

Delay:4 inversions

Size:12 transistors

Delay:2 inversions

Early Output Logic



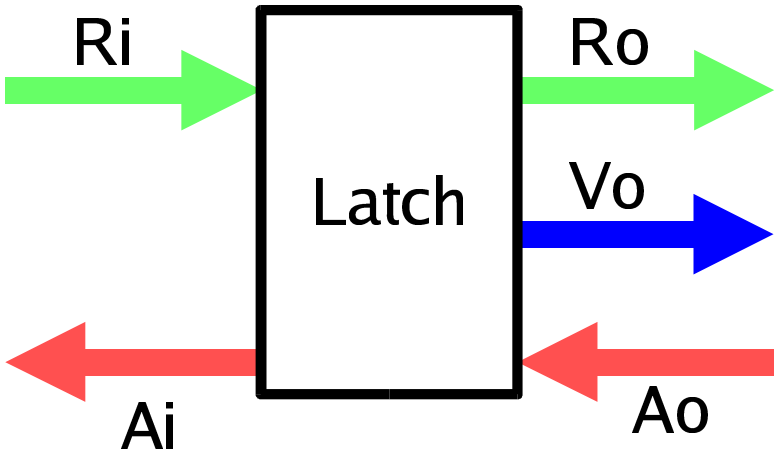
Guarding

Problem:

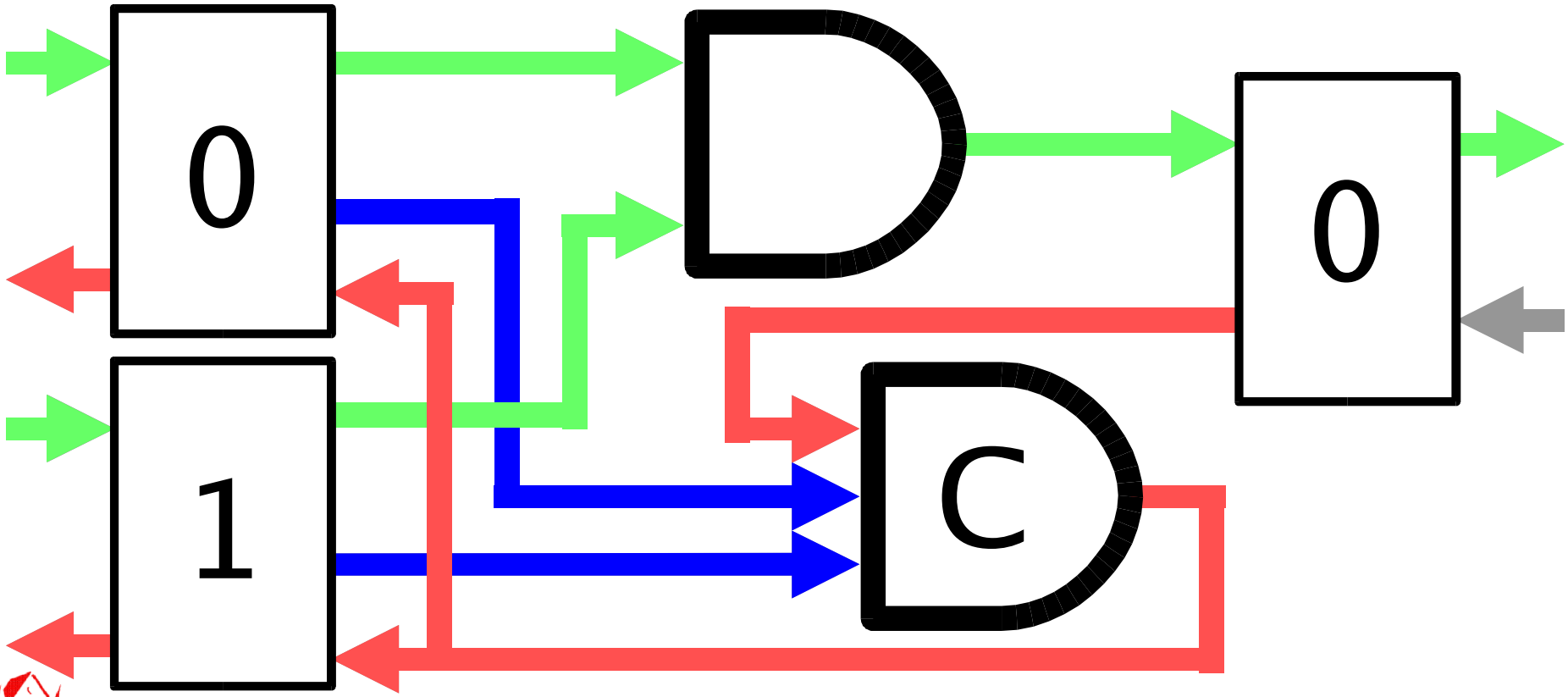
- Inputs
  - Late
  - Unnecessary
- Acknowledge before ready

Solution:

- Validity signal (Vo)



Early Output Guarding



Anti-Tokens

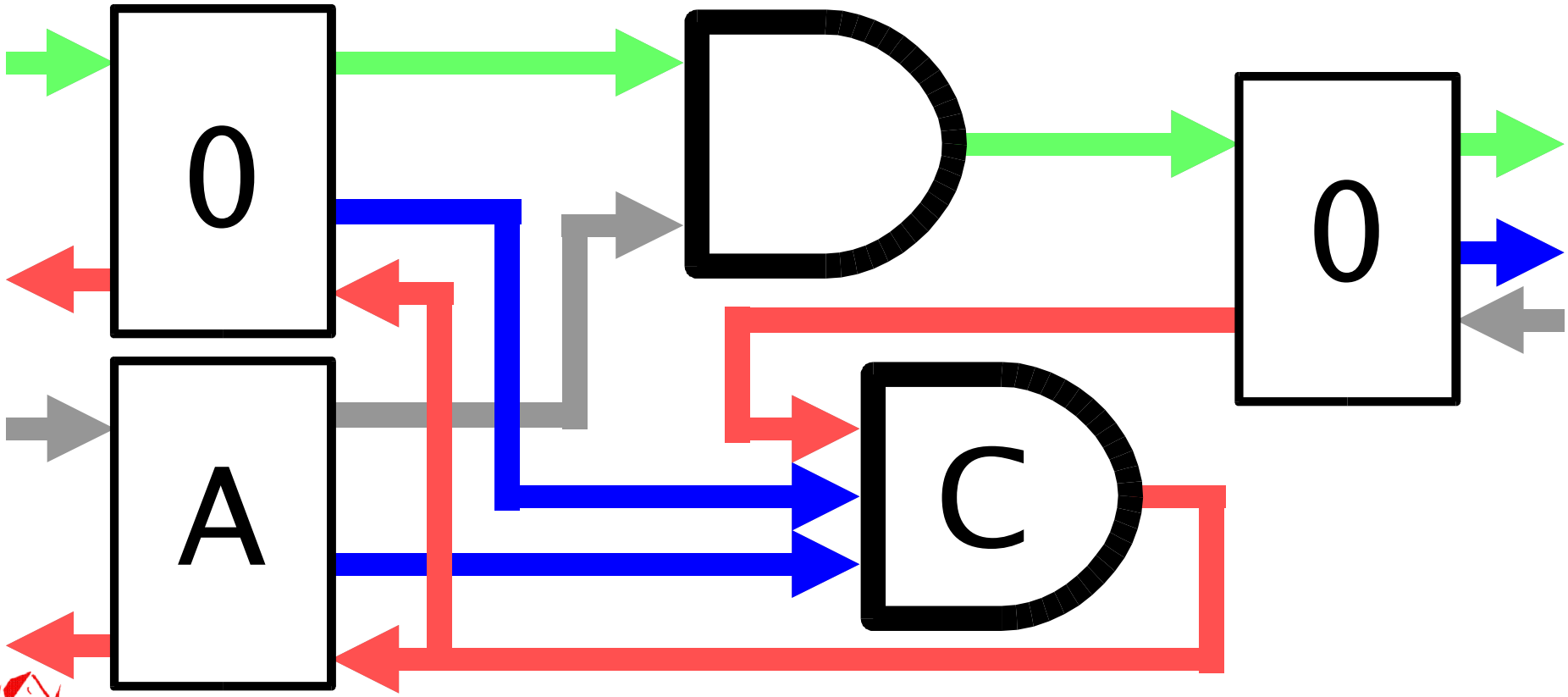
Don't:

- Stall entire stage until late input arrives

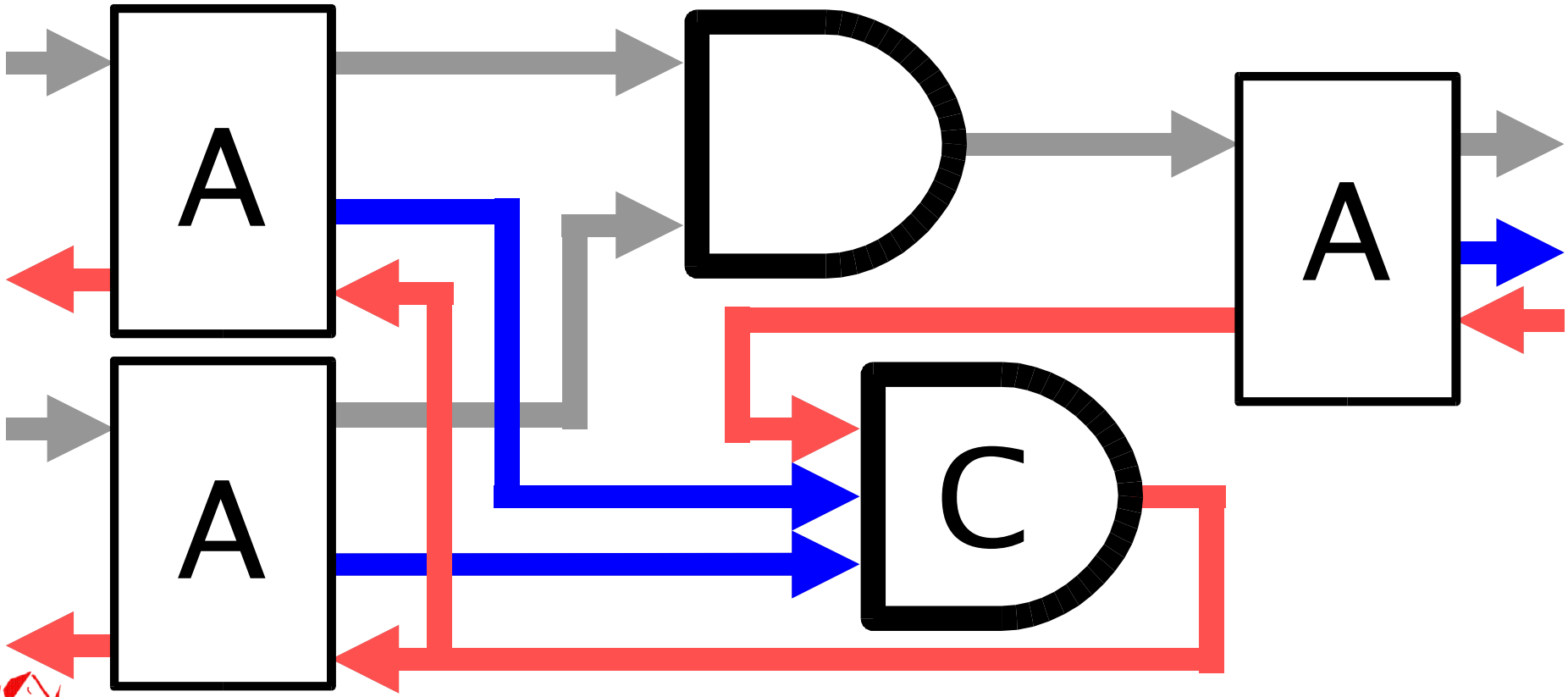
Do:

- Stall the latch instead
  - Early 'Validity'
  - Acknowledge before Data

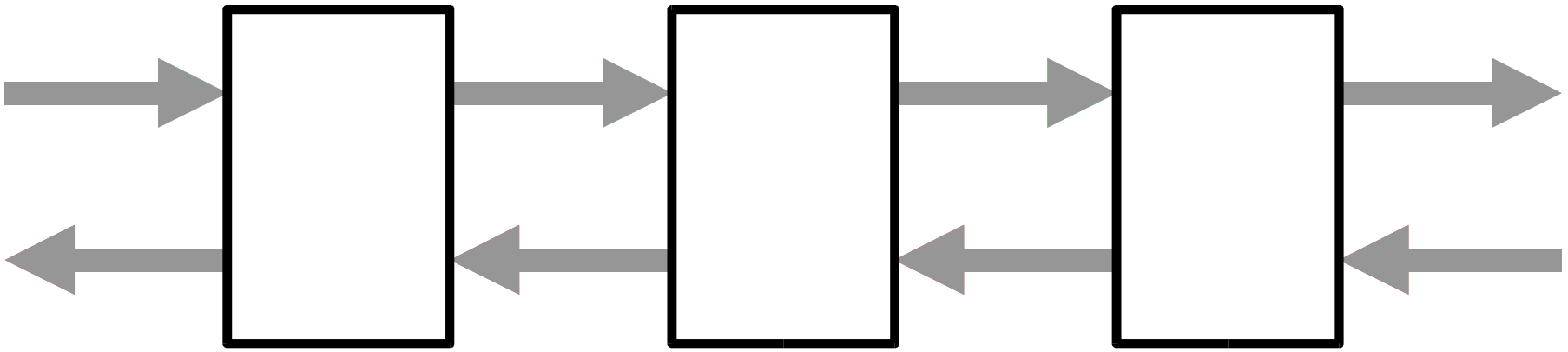
Anti-Token Generation



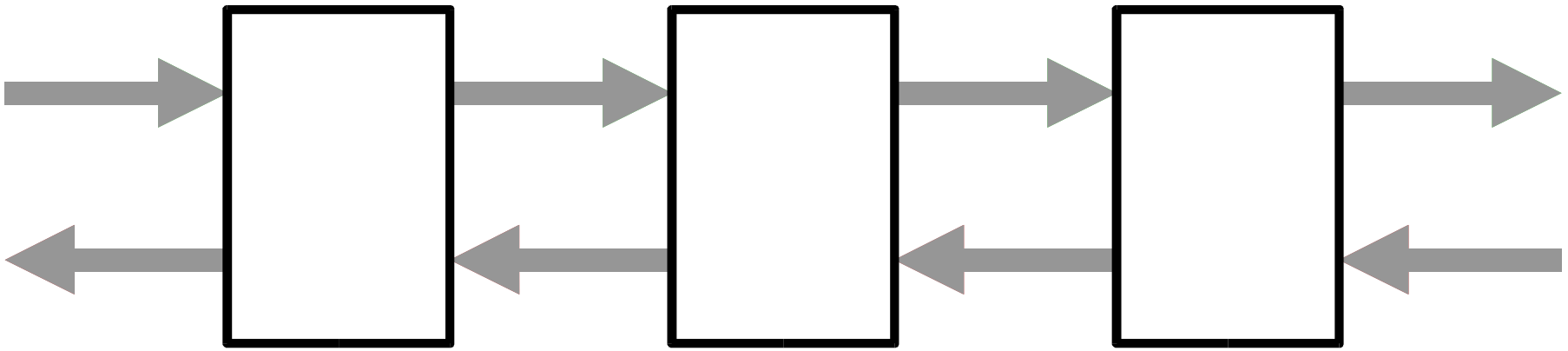
Anti-Token Propagation



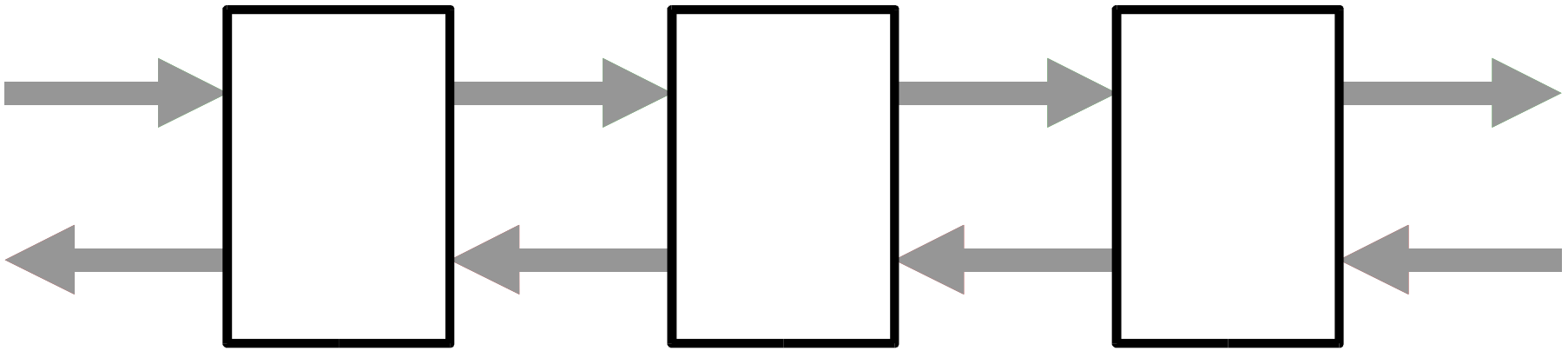
Token Pass



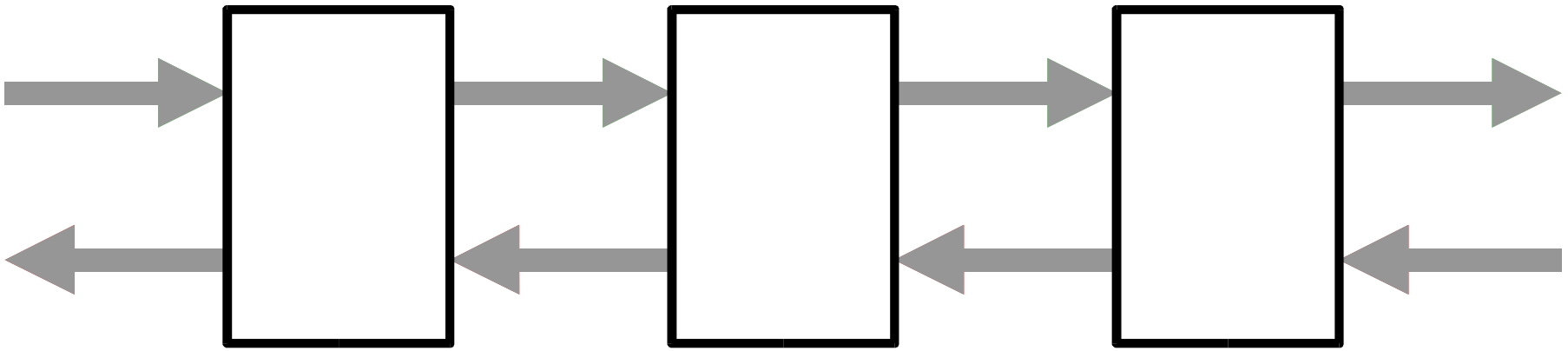
Anti-Token Pass



Token Anti-Token collision

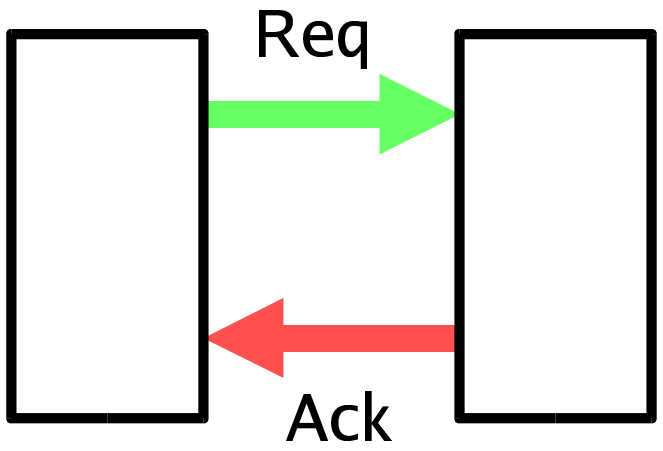


Token Anti-Token collision 2



Dual-Purpose Signals

- Arbiter free
- Req:
  - Token Request
  - Anti-Token Acknowledge
- Ack:
  - Anti-Token Request
  - Token Acknowledge



Conclusions

- New, fine-grain, asynchronous pipeline
- Faster than DIMS (2x)
- Smaller than DIMS (4x)
- Lower power than DIMS
- Some speed advantages over synchronous designs
- Counterflow - no arbitration
- Requires some timing assumptions

Timing Hazard example