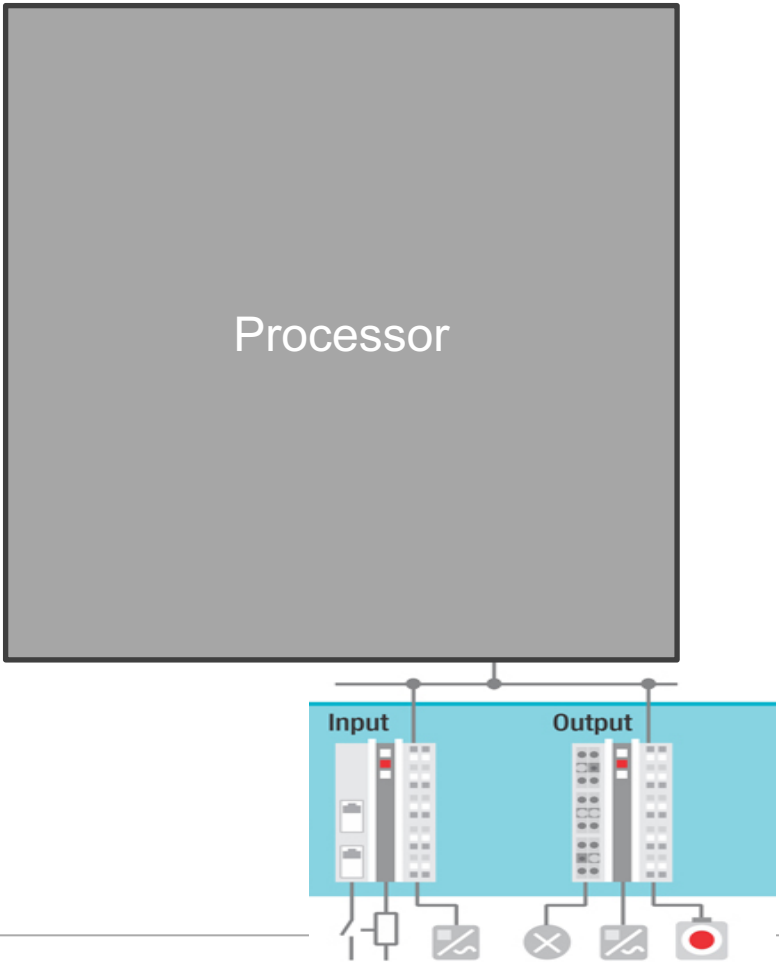


# **Building an Industrial ARM™**

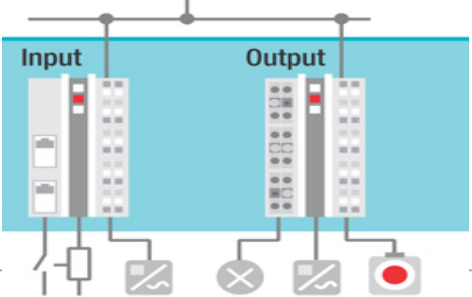
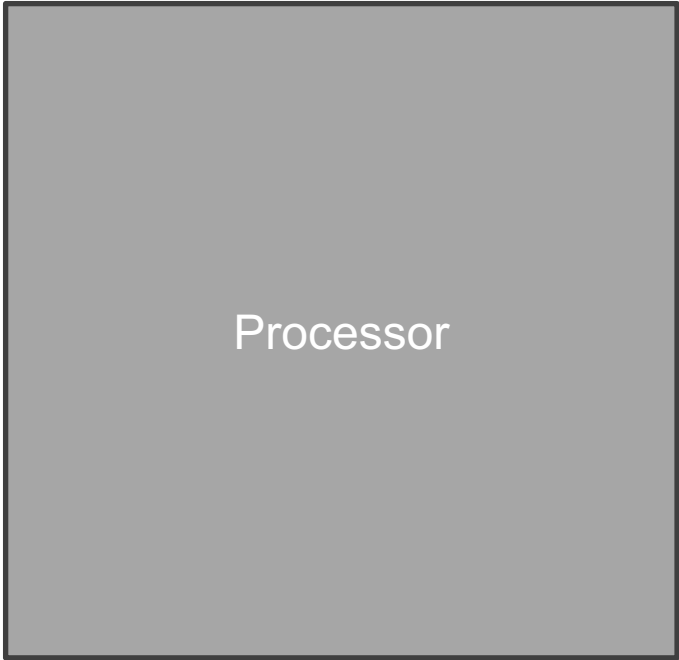
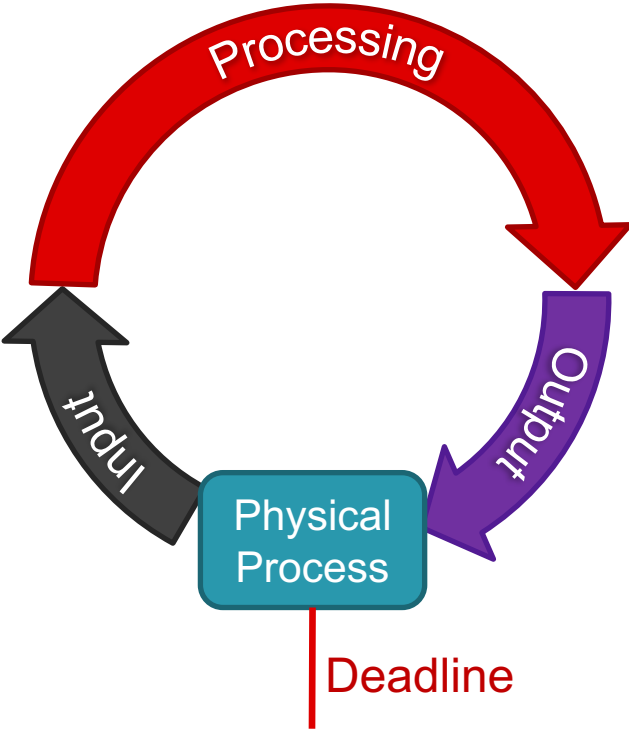
## **AM65xx Architecture Differentiations for Industrial Applications**

**Managing Delay**

# Processor Components

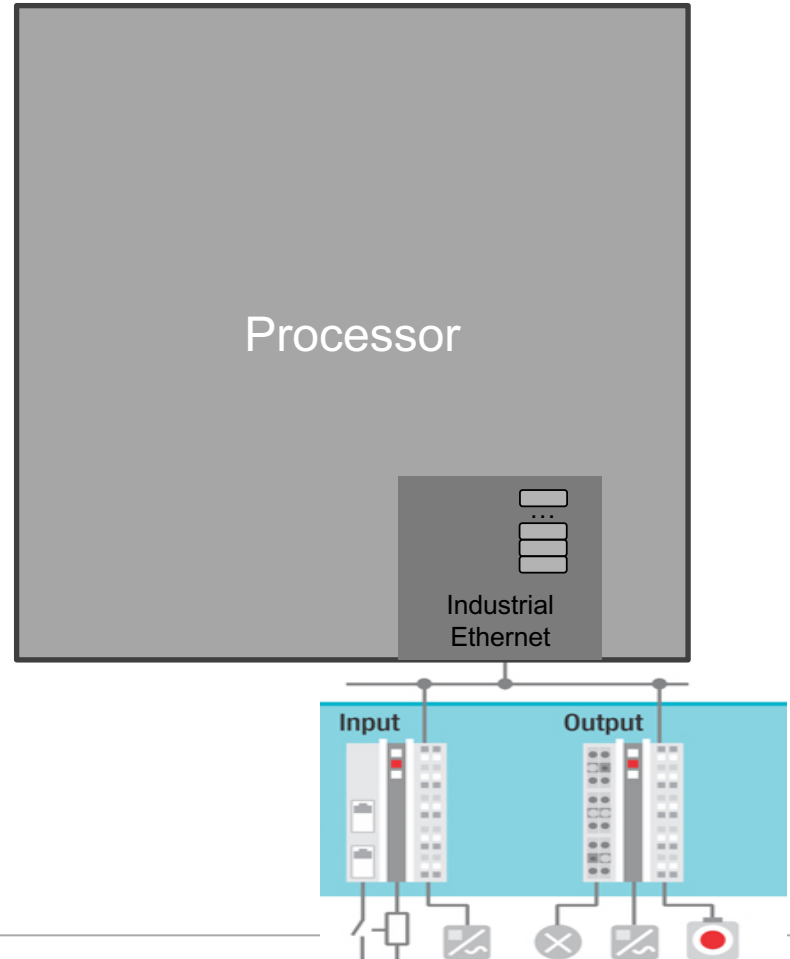


# Processor Components



# Processor Components

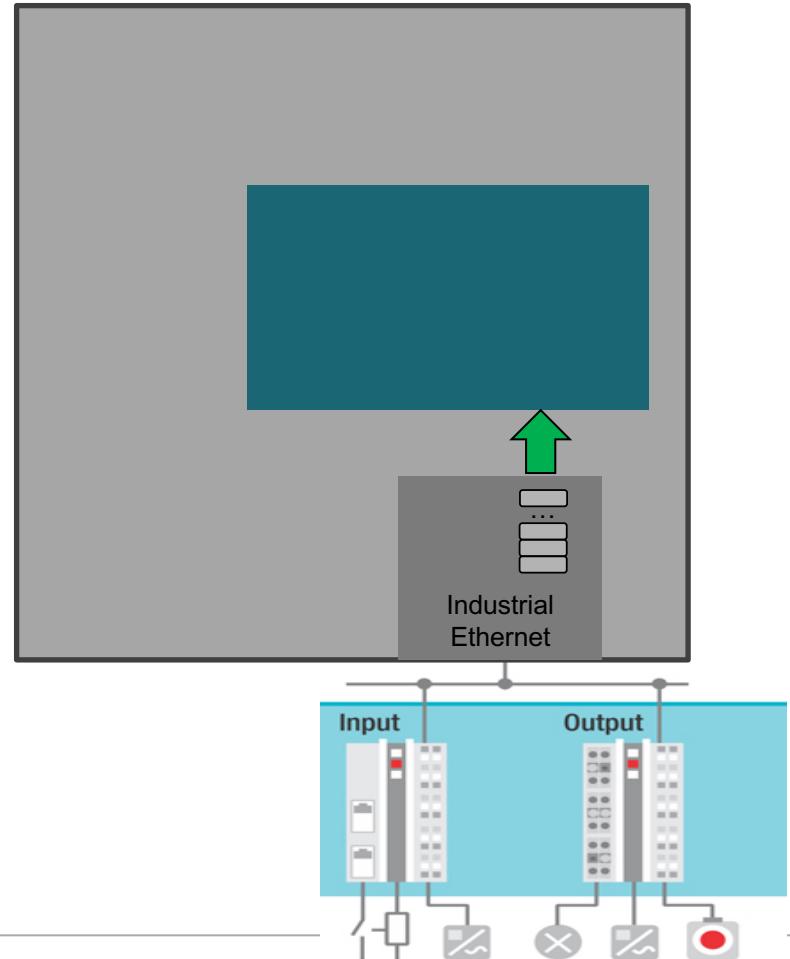
- Industrial Ethernet
  - Interface
  - Queues for moving data





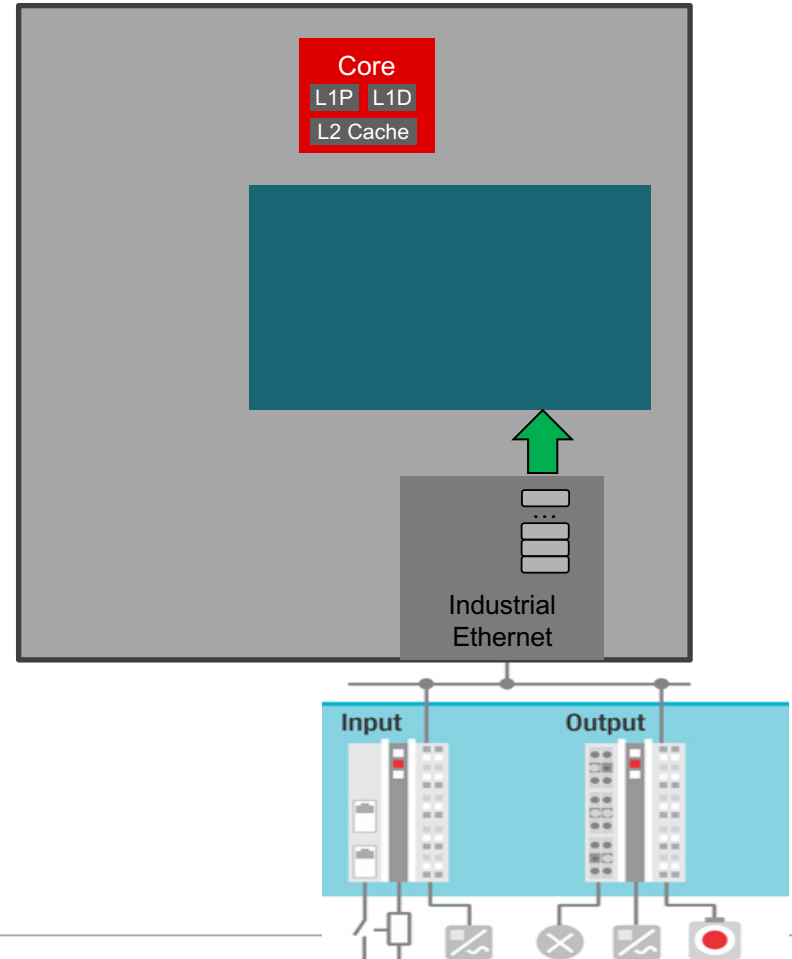
# Processor Components

- Industrial Ethernet
  - Interface
  - Queues for moving data
- Busses and Interconnect
  - Data movement

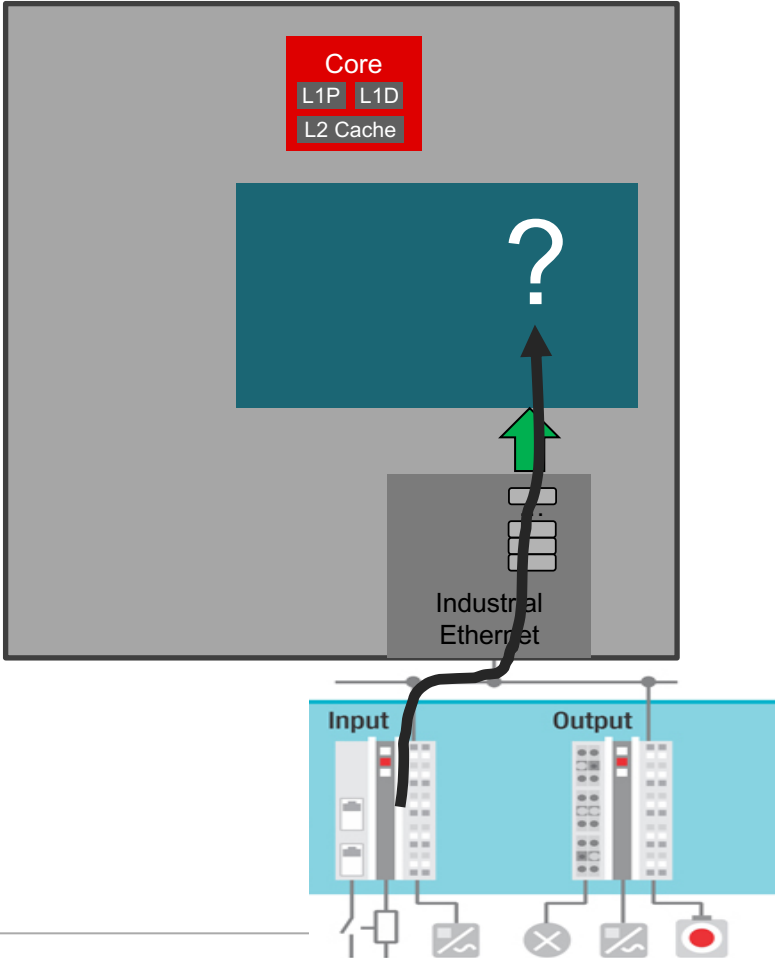
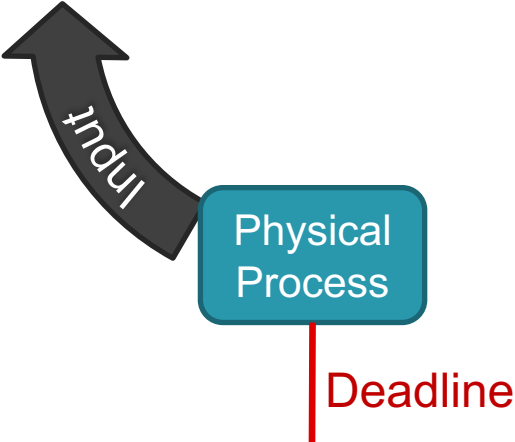


# Processor Components

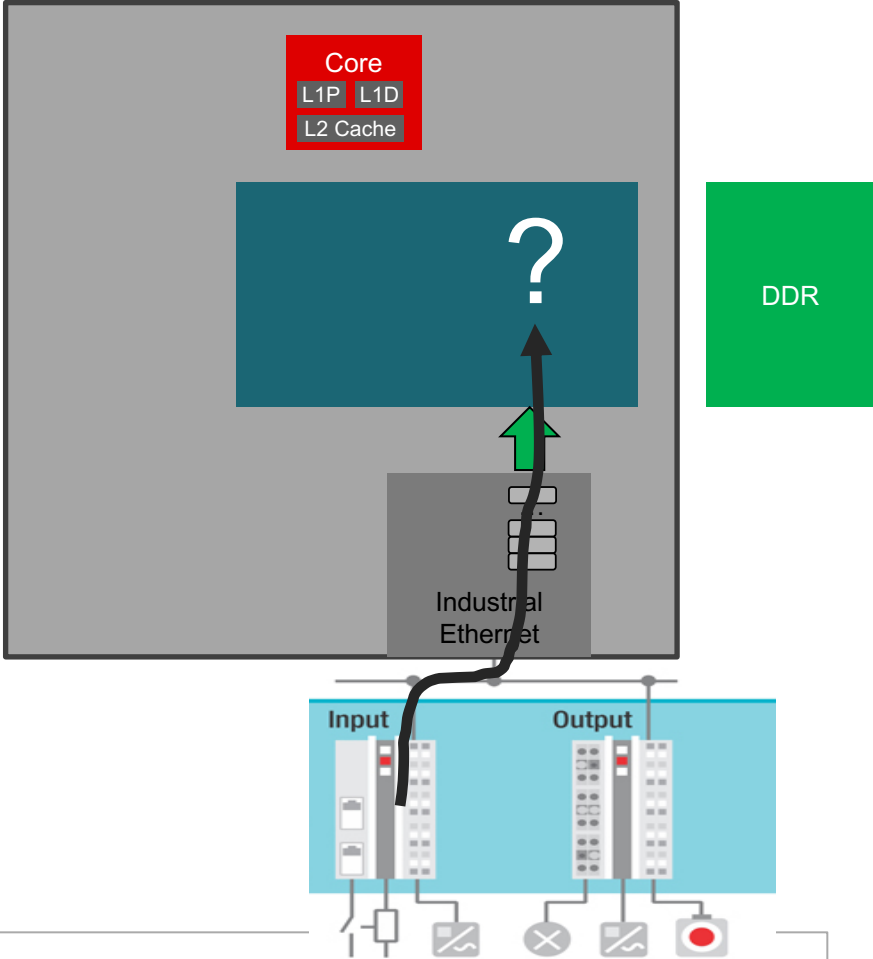
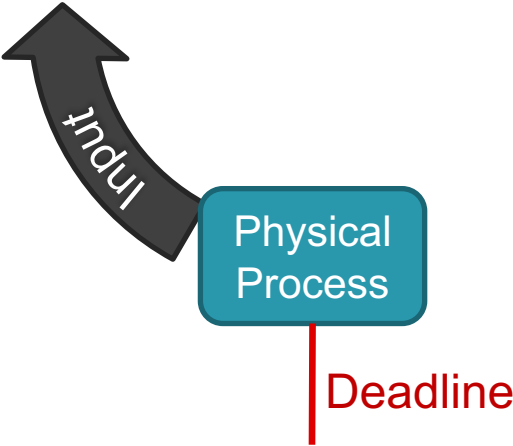
- Industrial Ethernet
  - Interface
  - Queues for moving data
- Busses and Interconnect
  - Data movement
- Core
  - CPU
  - Caches



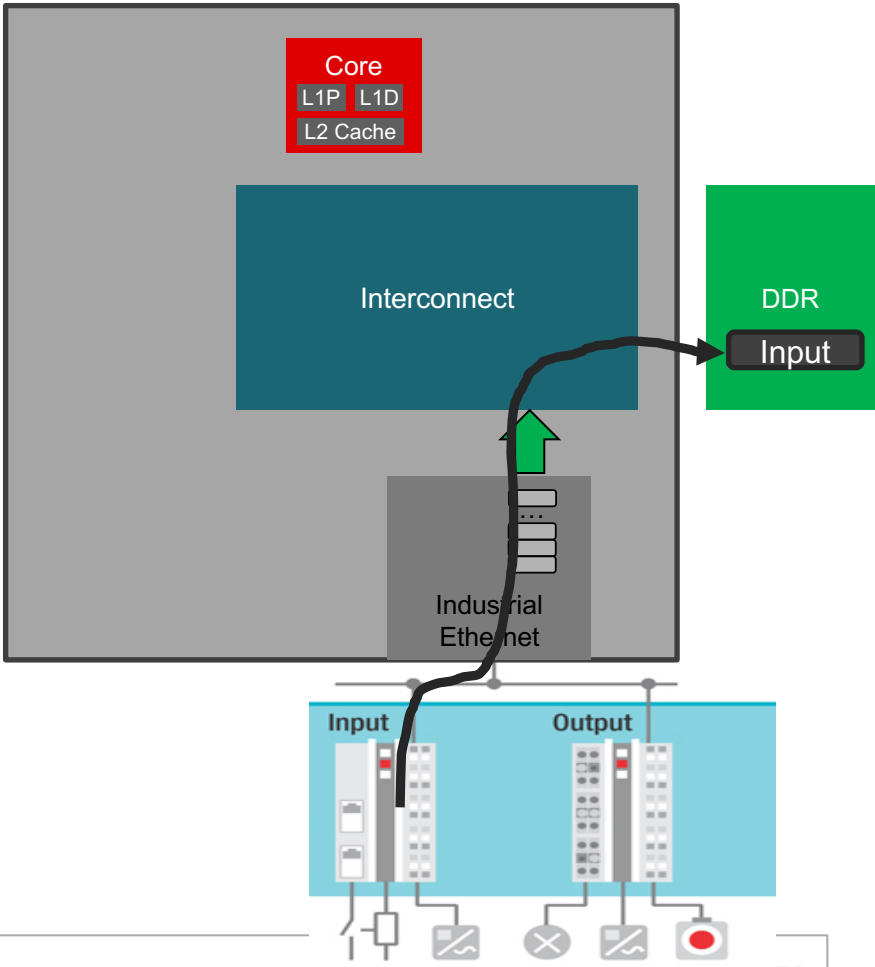
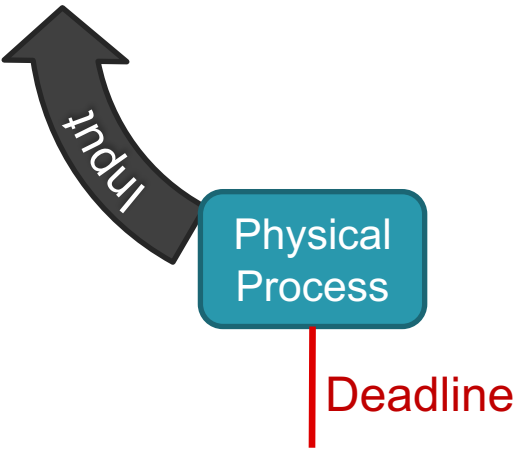
# Where to store input?



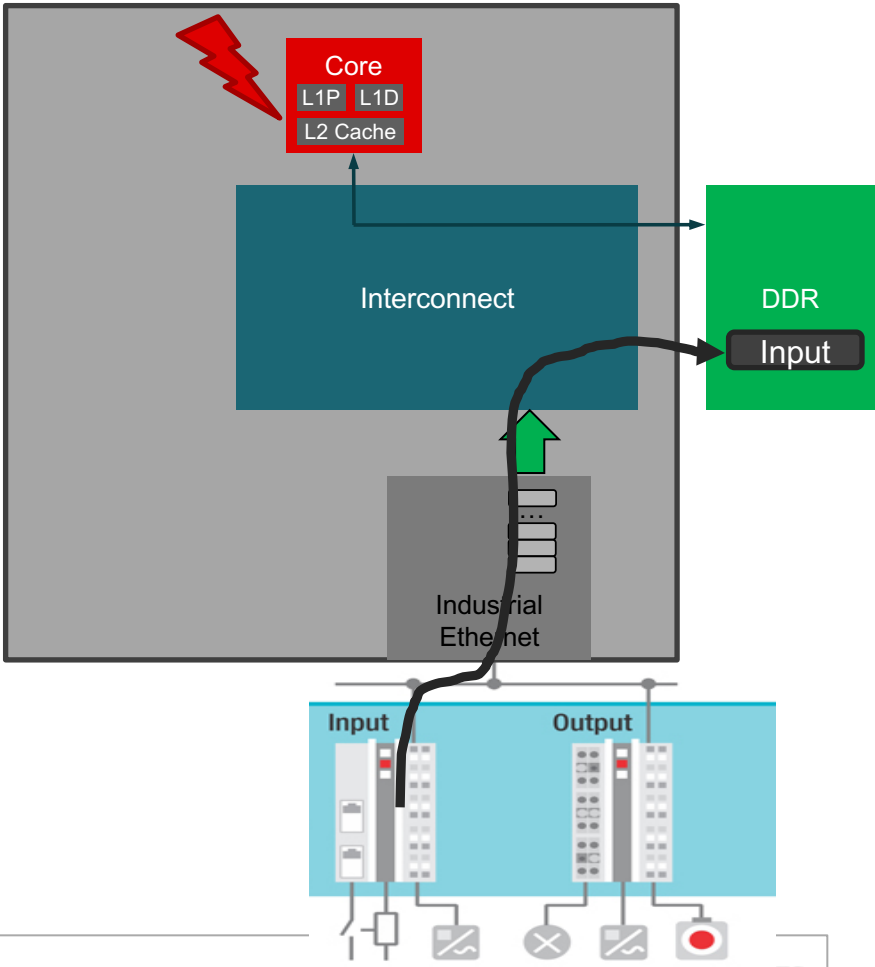
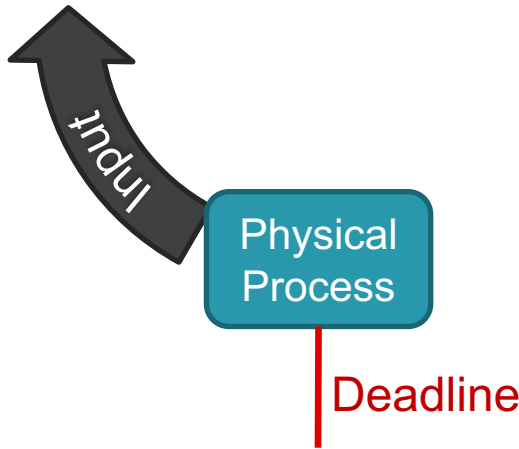
# Store Input in DDR



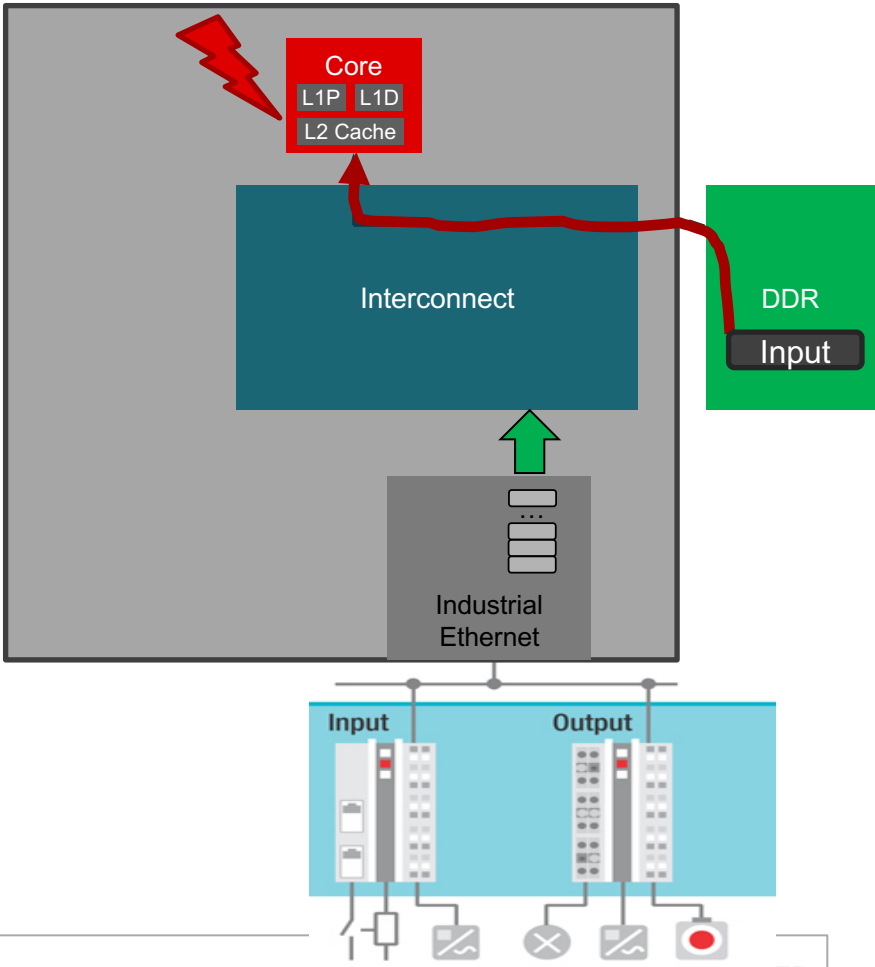
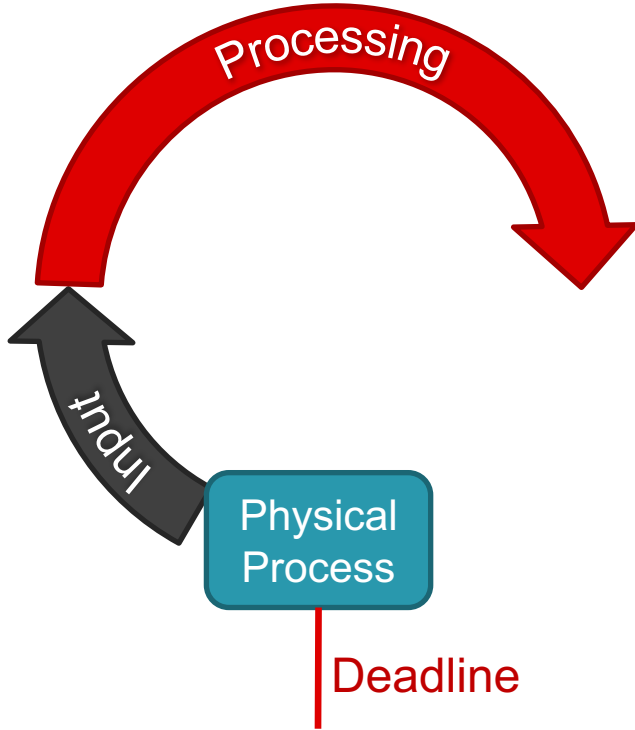
# Store Input in DDR



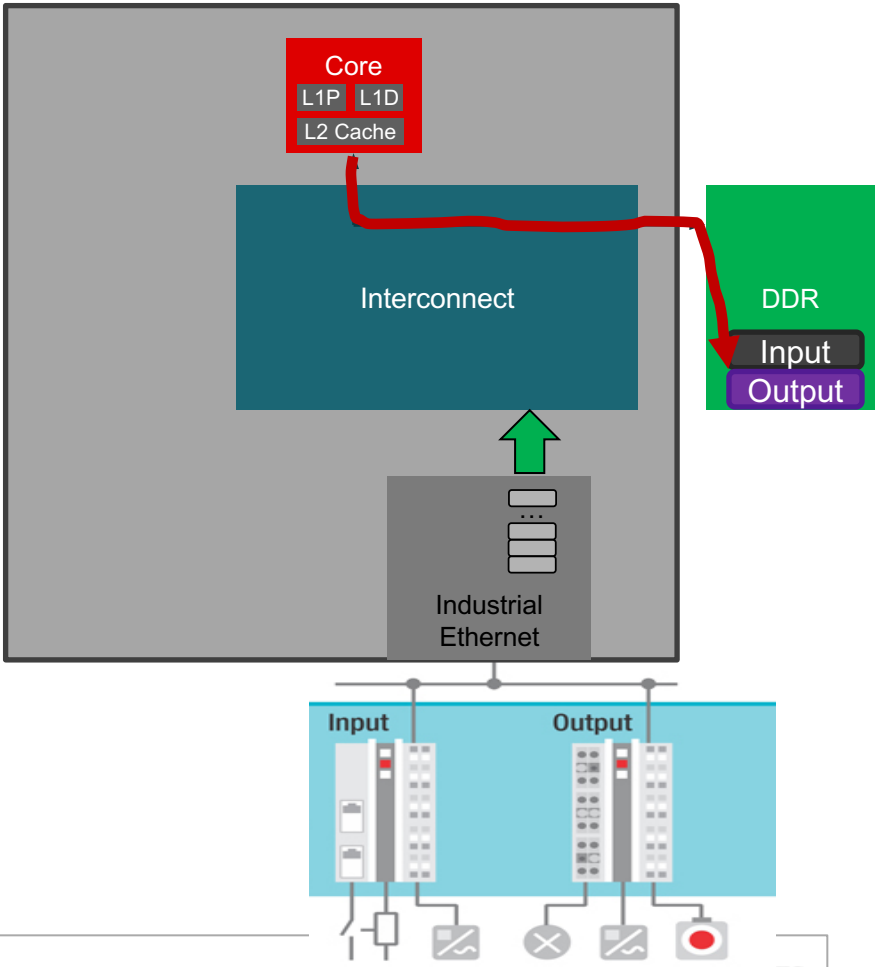
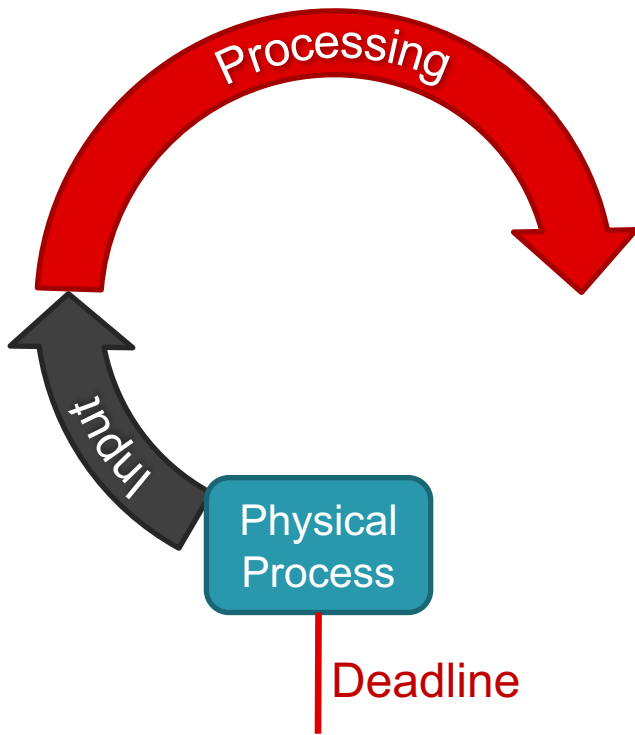
# Interrupt Core



# Read/Process/Write

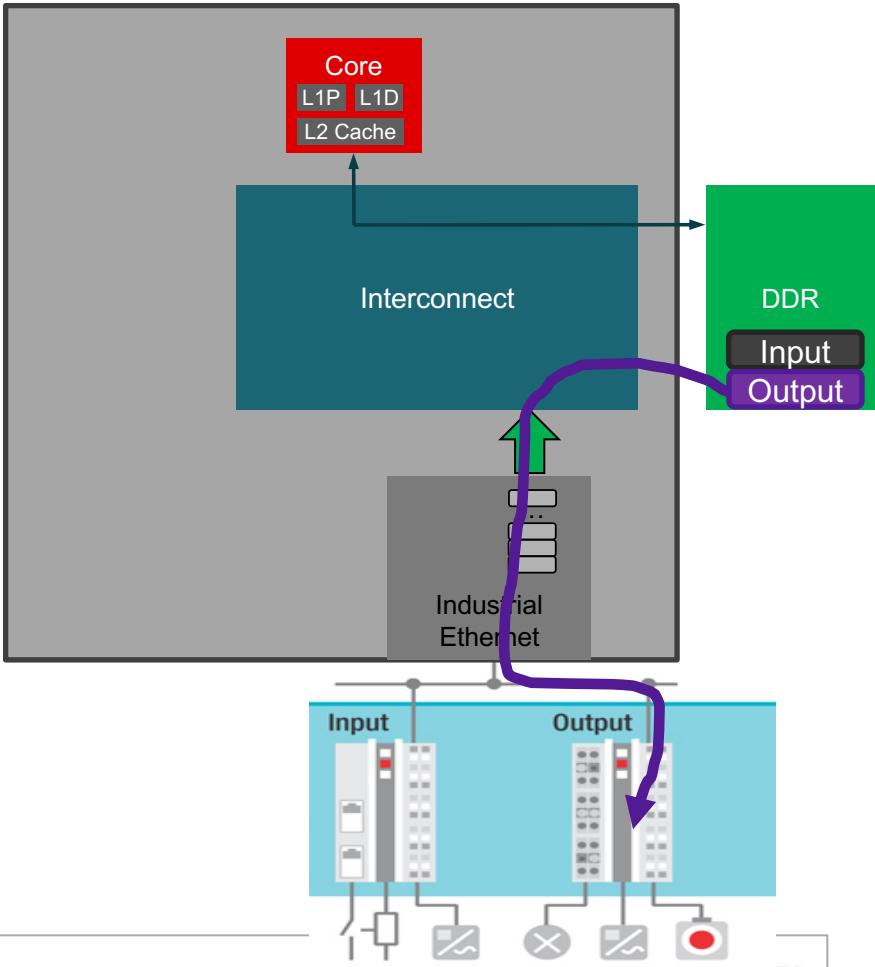
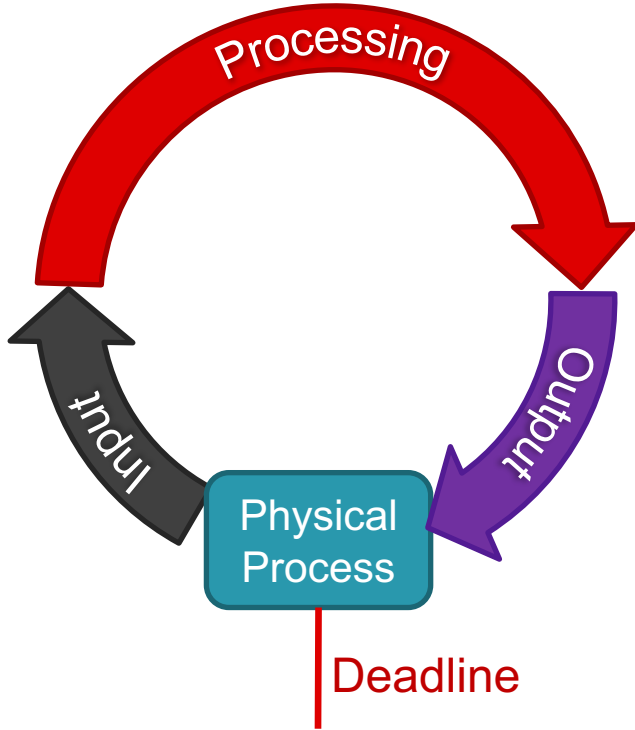


# Read/Process/Write

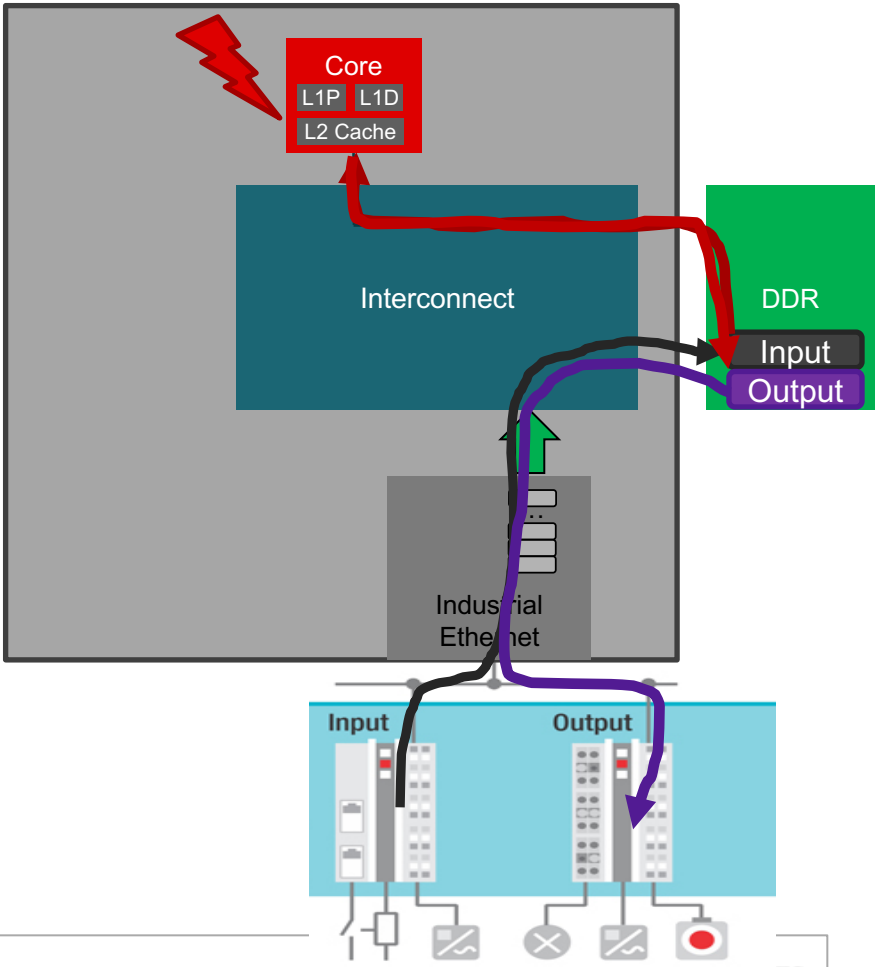
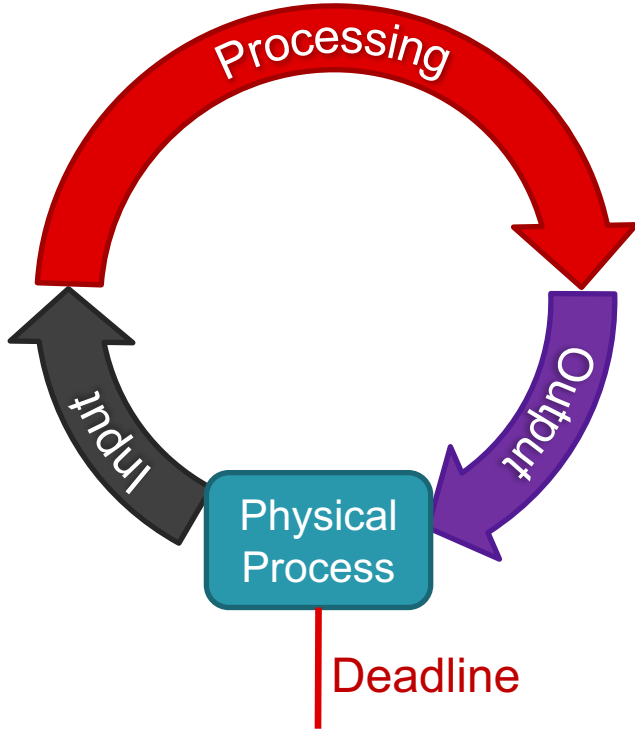




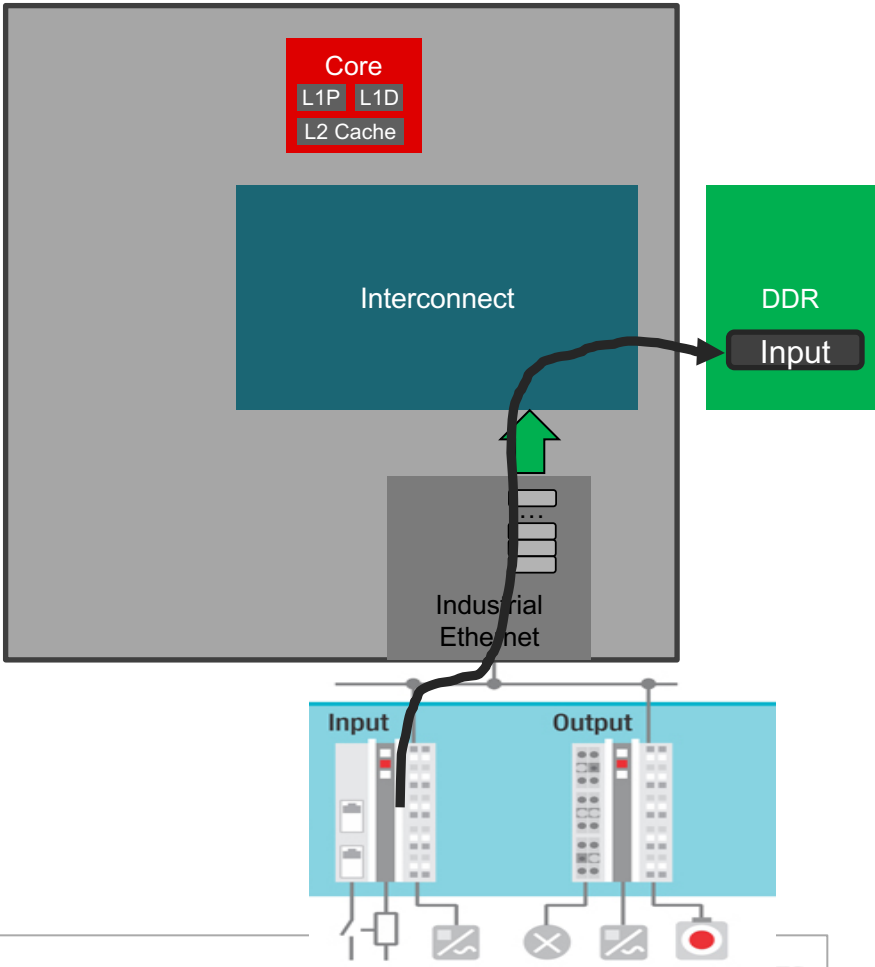
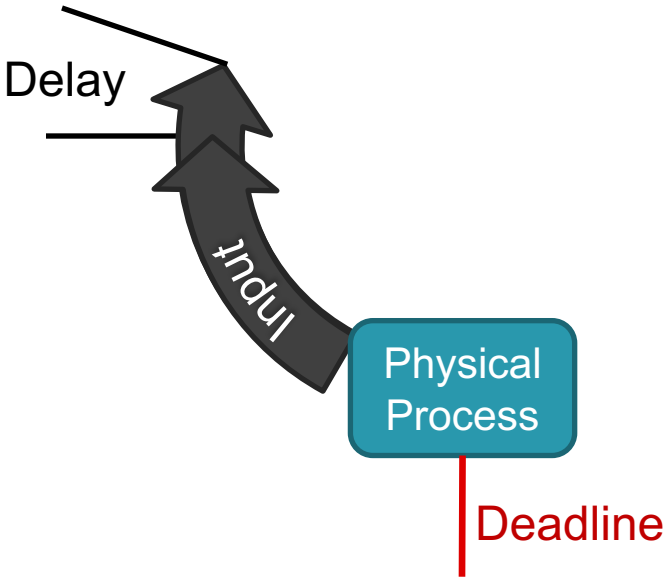
# Output Before Deadline



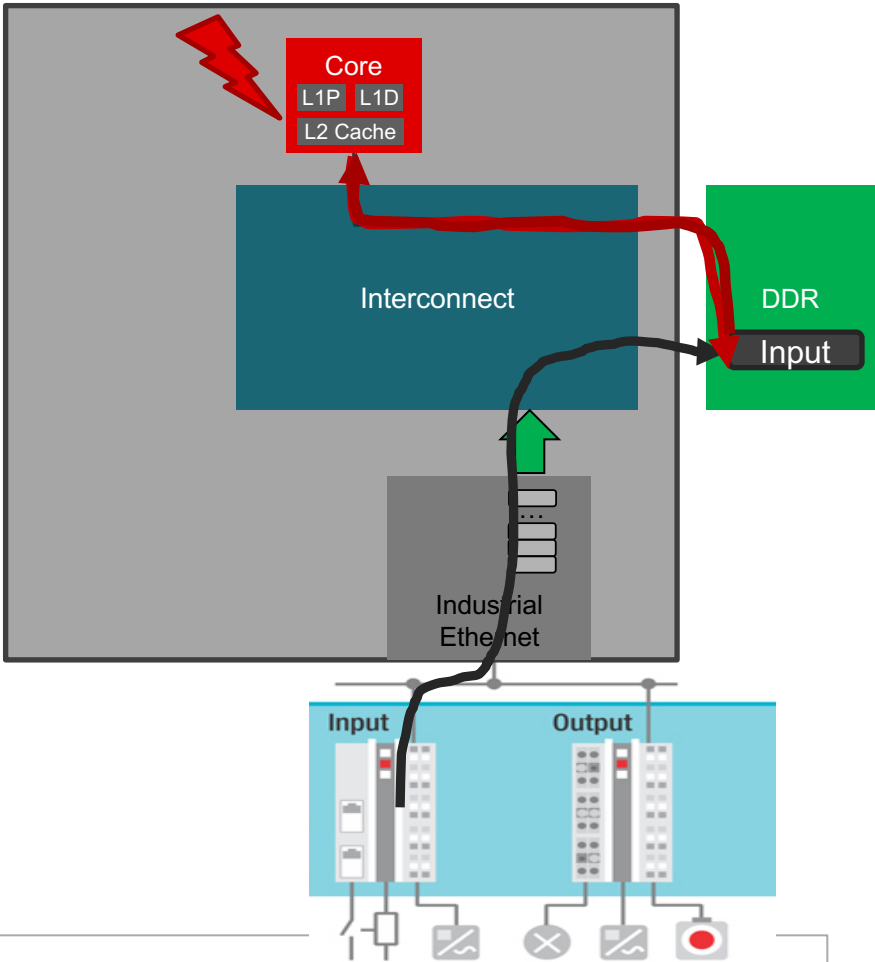
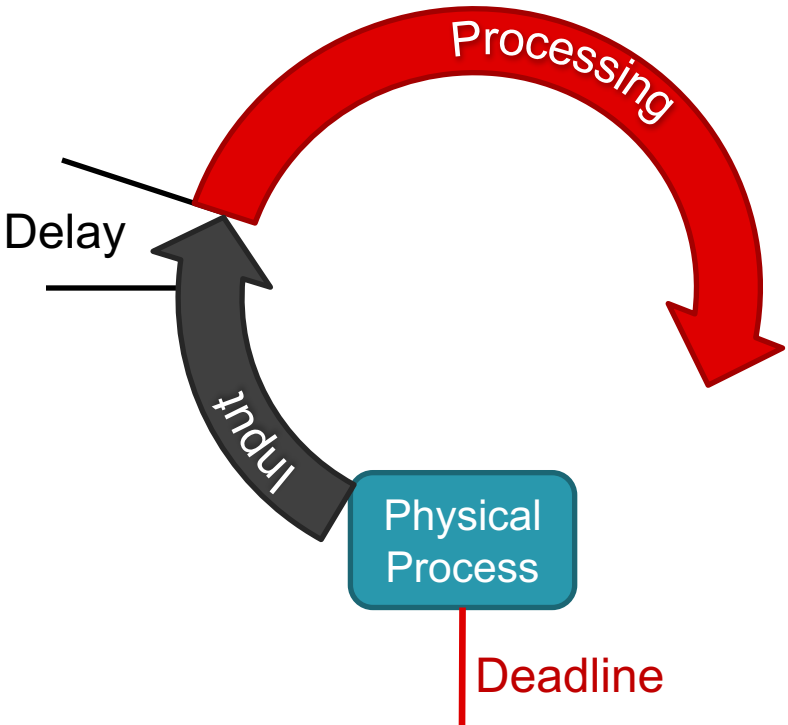
# Putting it All Together



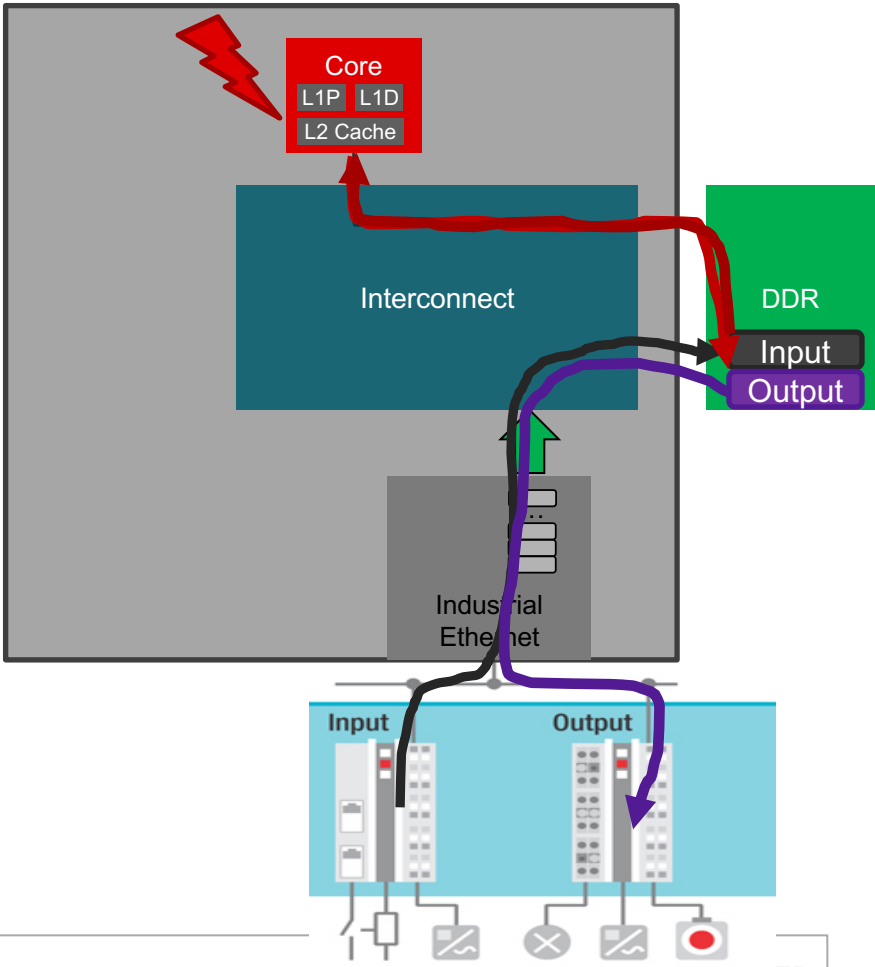
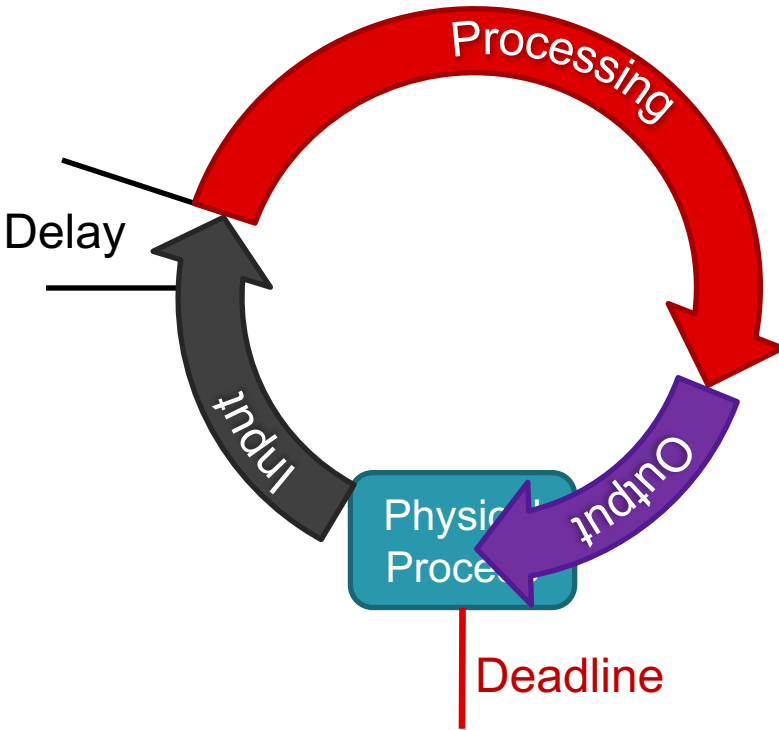
# What if it takes longer?



# Delays Are Bad

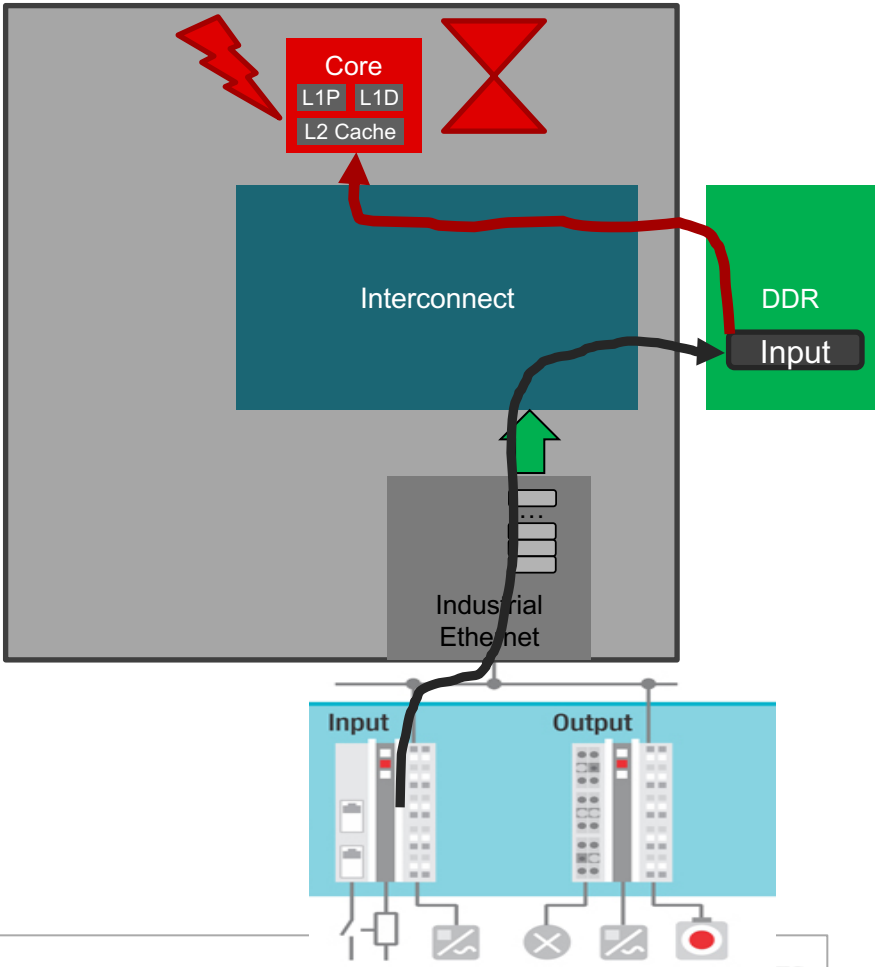
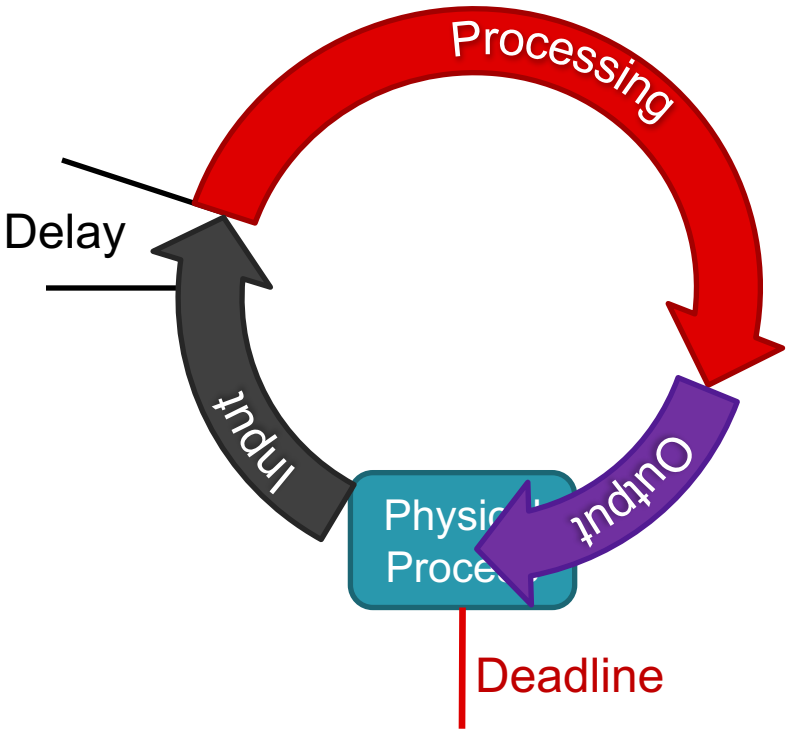


# Delays Are Bad



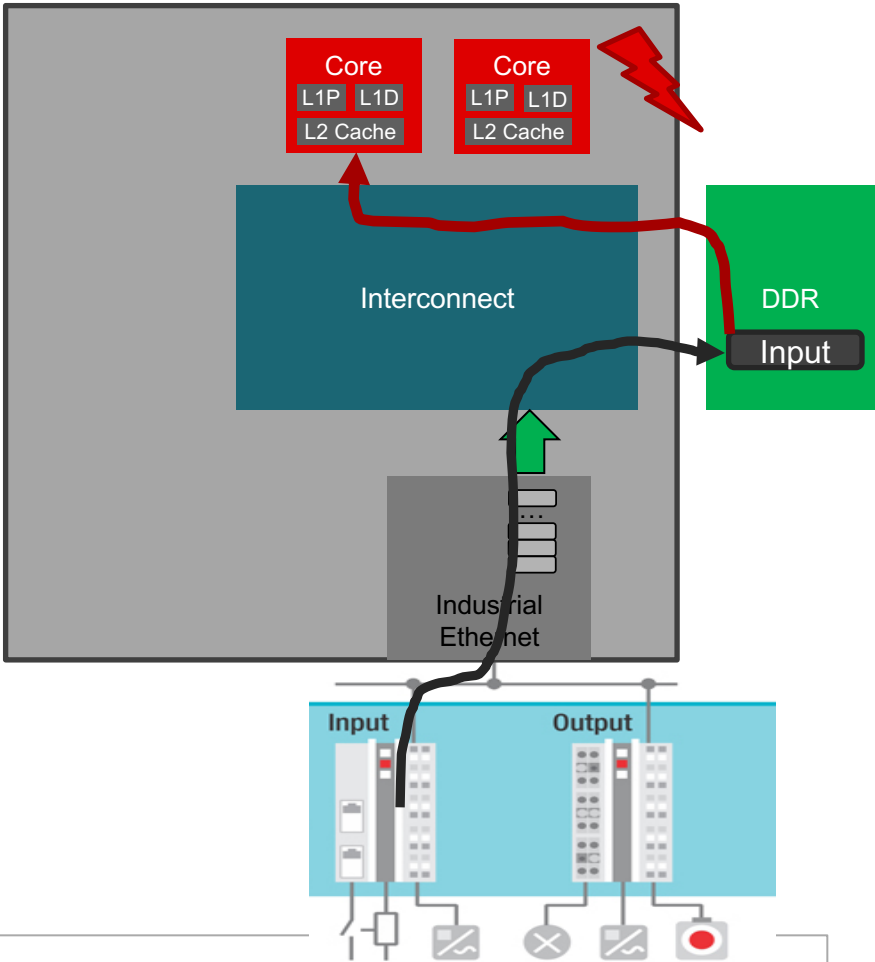
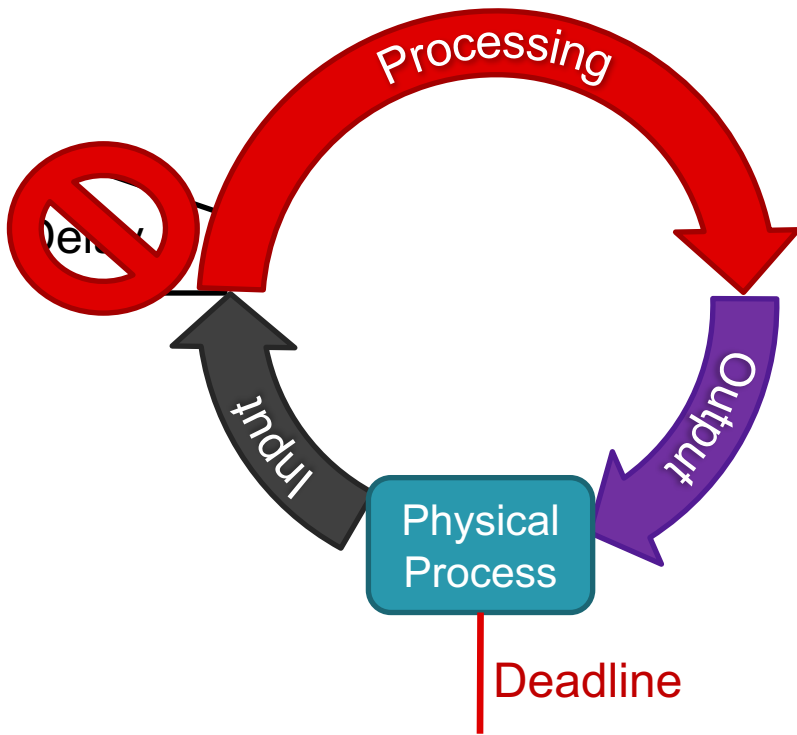
# Sources of this Delay

- Core is busy



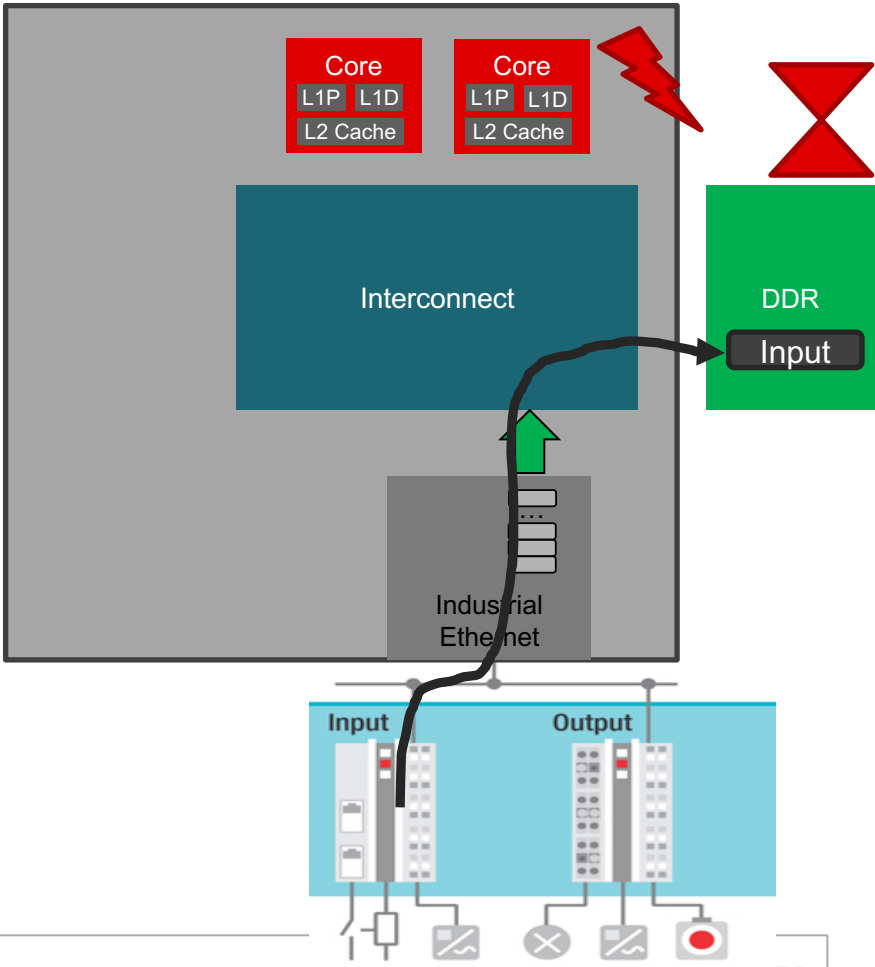
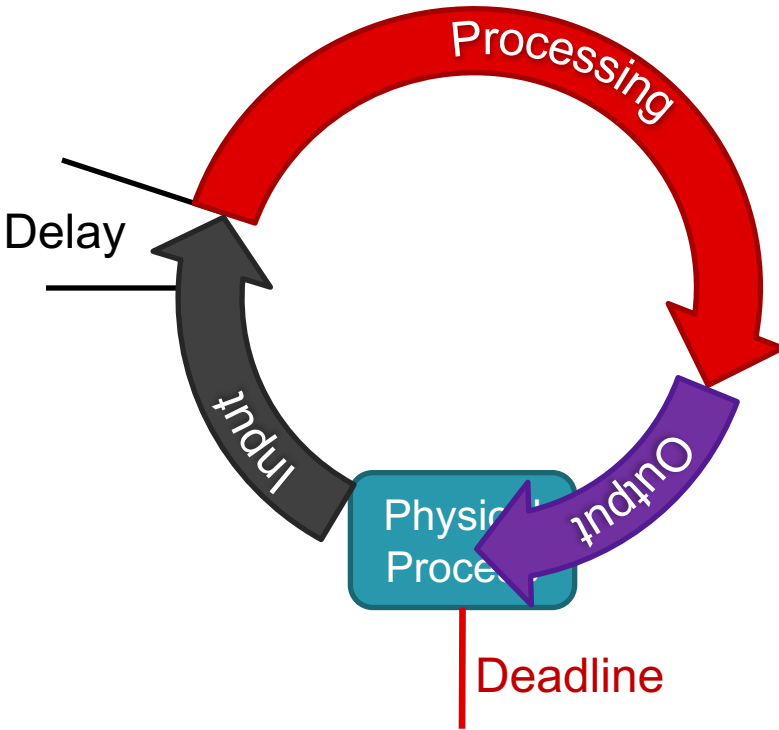
# Add More Cores

- Use new core for interrupts



# Delay from DDR Refresh

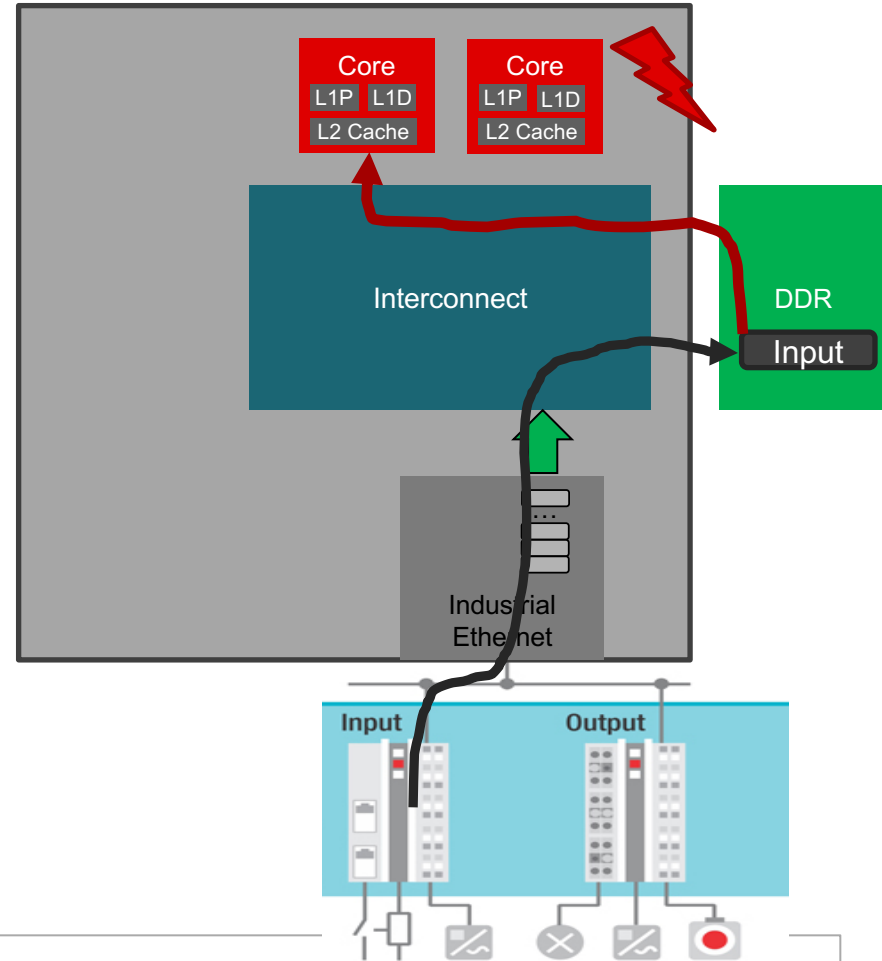
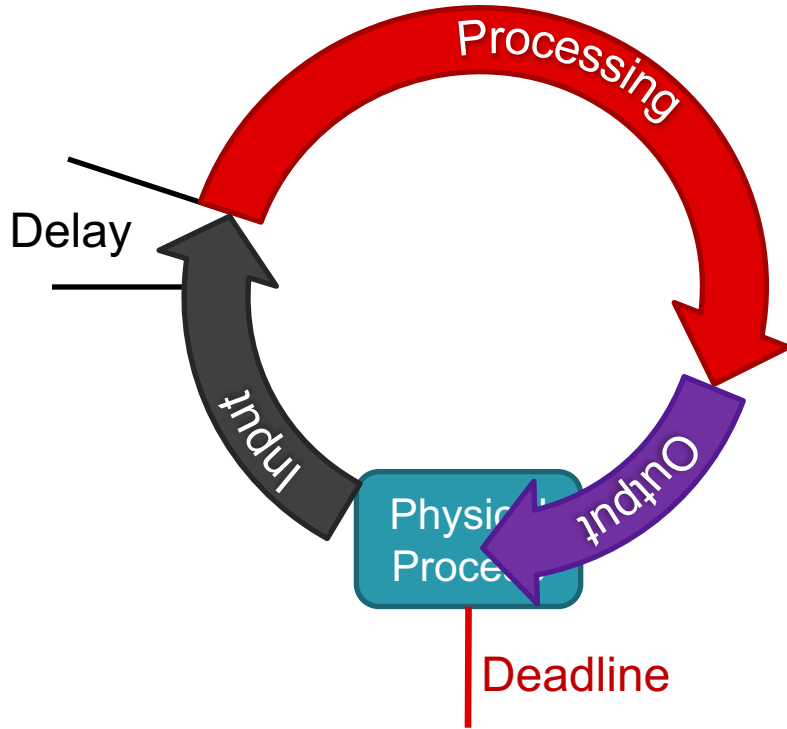
- DDR has to be refreshed at regular rate





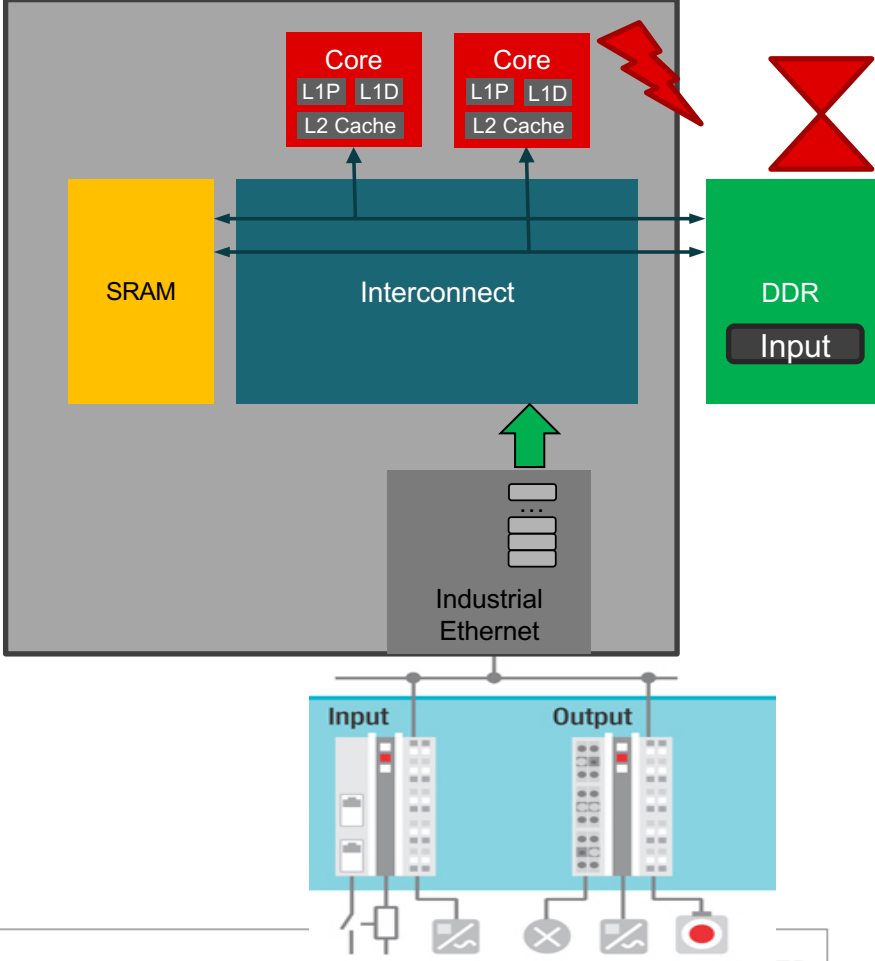
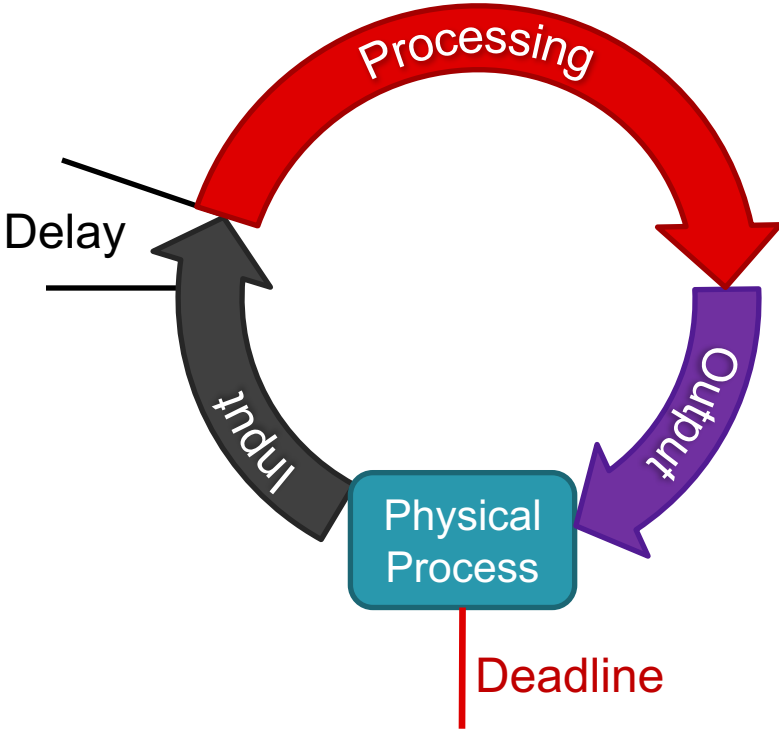
# Delay from DDR Refresh

- DDR has to be refreshed at regular rate



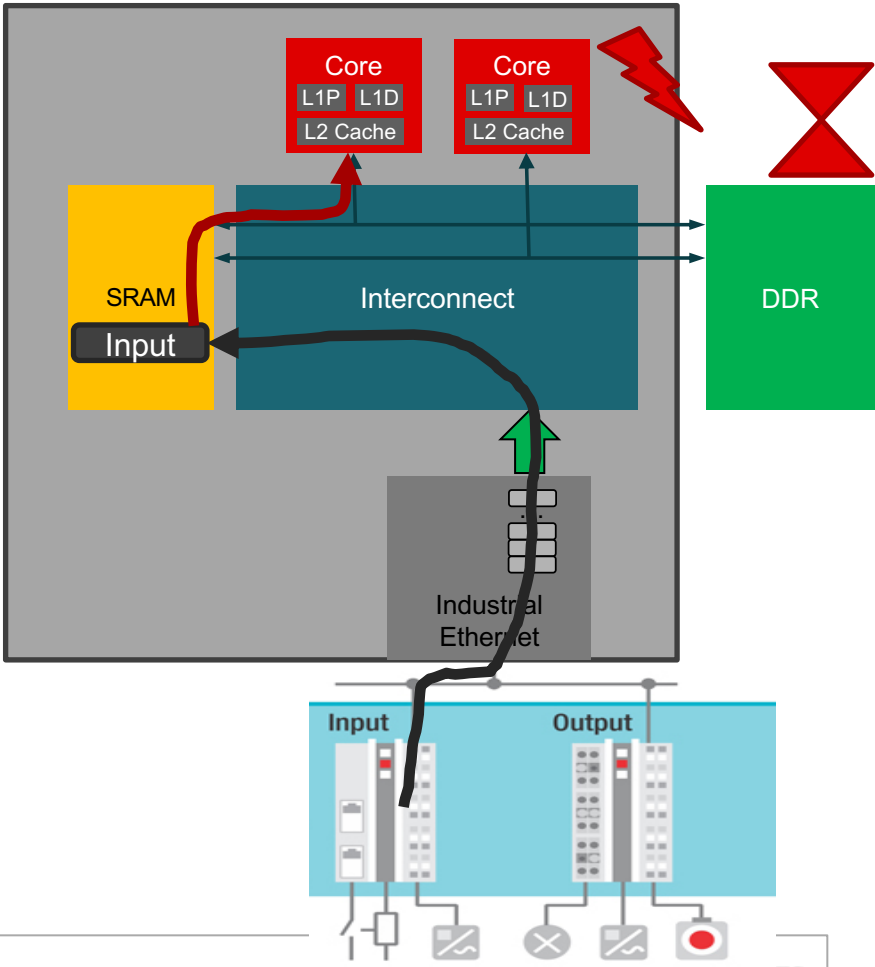
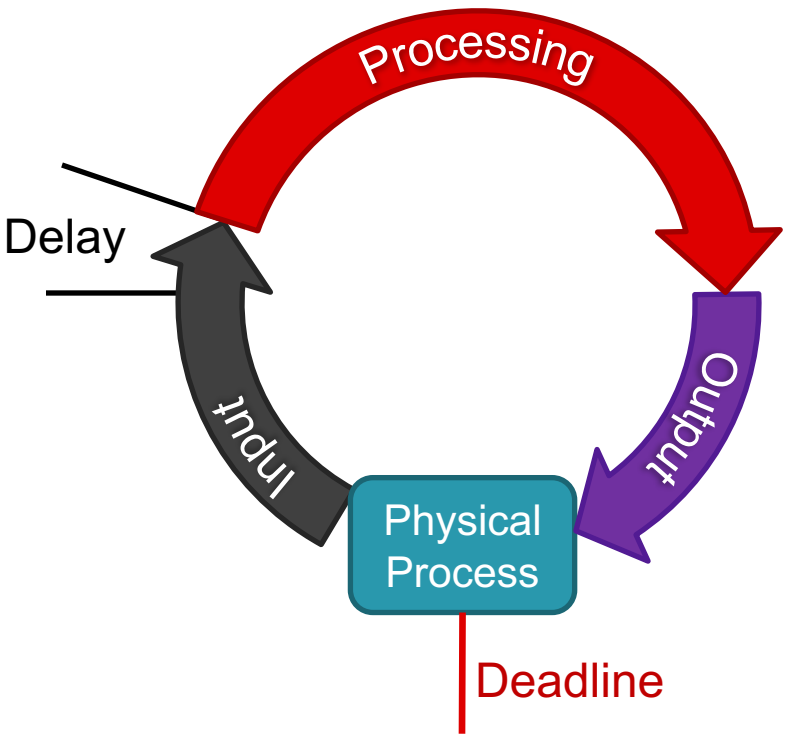
# Add On-Chip Memory

- Store time critical data here

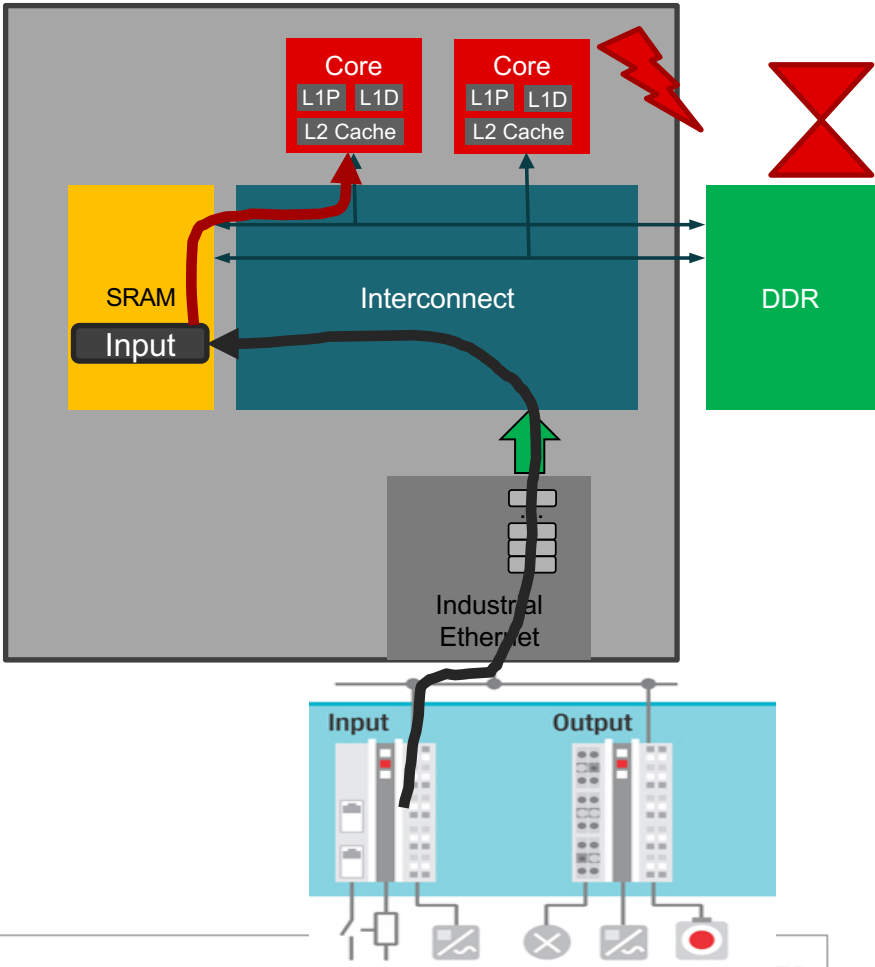
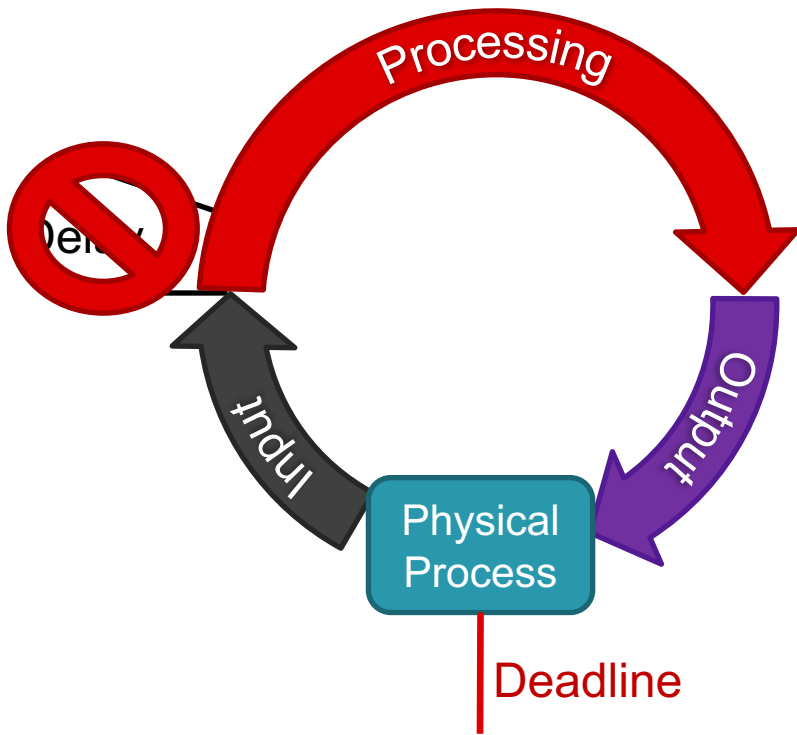


# Add On-Chip Memory

- Store time critical data here

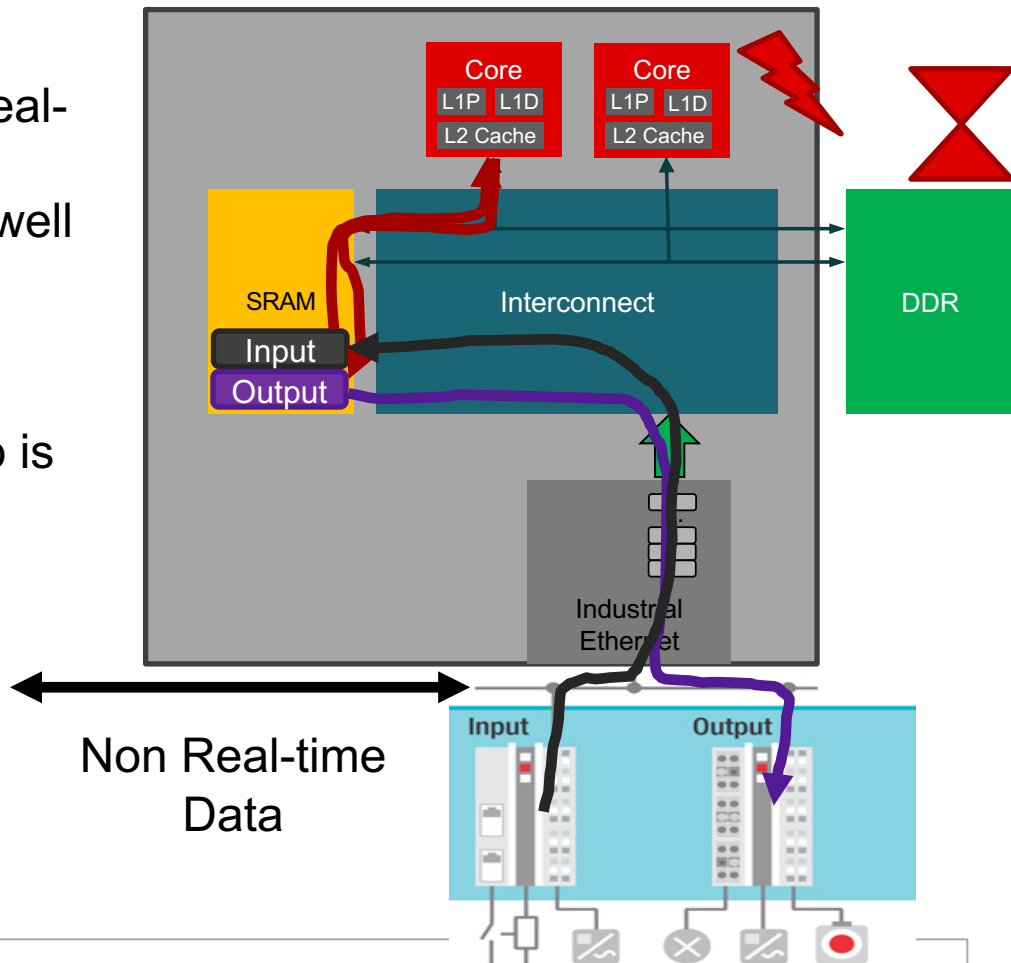


# Add On-Chip Memory



# What about other traffic?

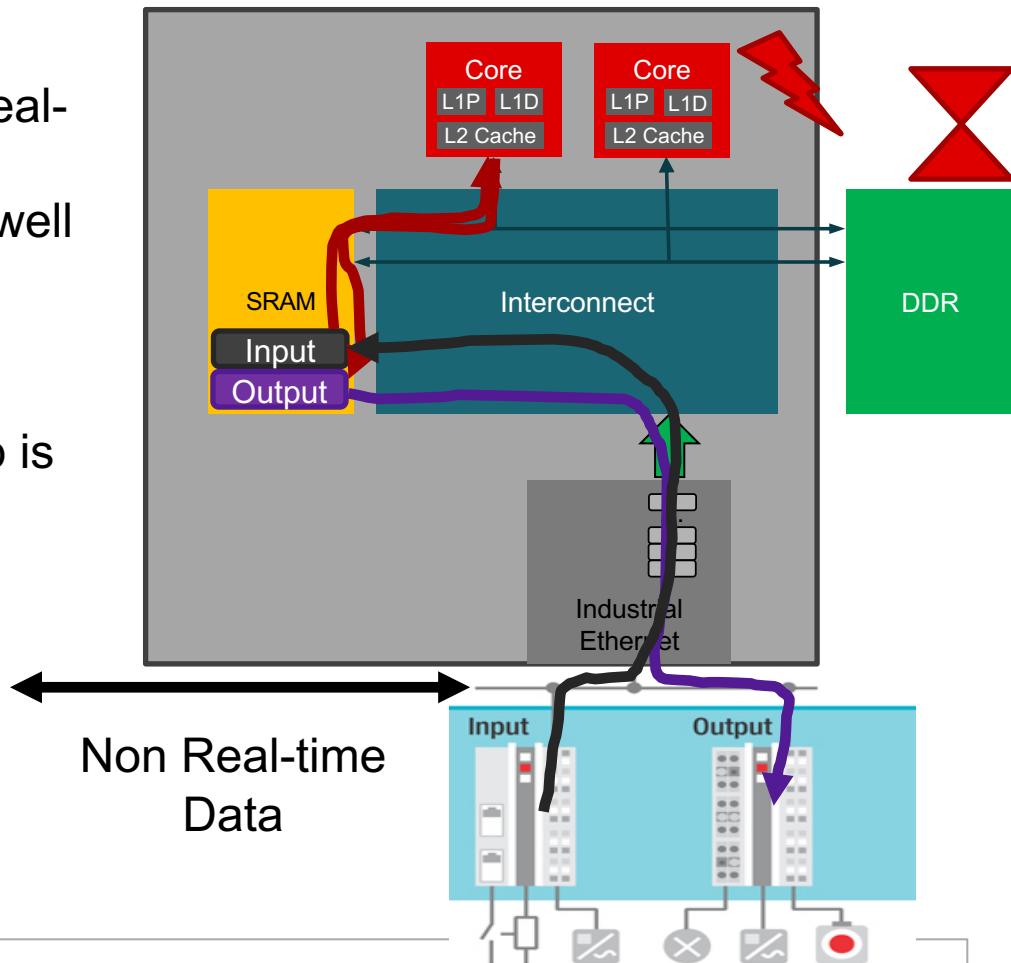
- Modern systems can't just focus on real-time traffic
- Other data needs to be managed as well
- Any data that needs to flow through system but is not time bound
  - Management and Metadata
  - Not used for Closed Loop, or loop is very relaxed



# What about other traffic?

- Modern systems can't just focus on real-time traffic
- Other data needs to be managed as well
- Any data that needs to flow through system but is not time bound
  - Management and Metadata
  - Not used for Closed Loop, or loop is very relaxed

Can this data cause problems with real-time data?

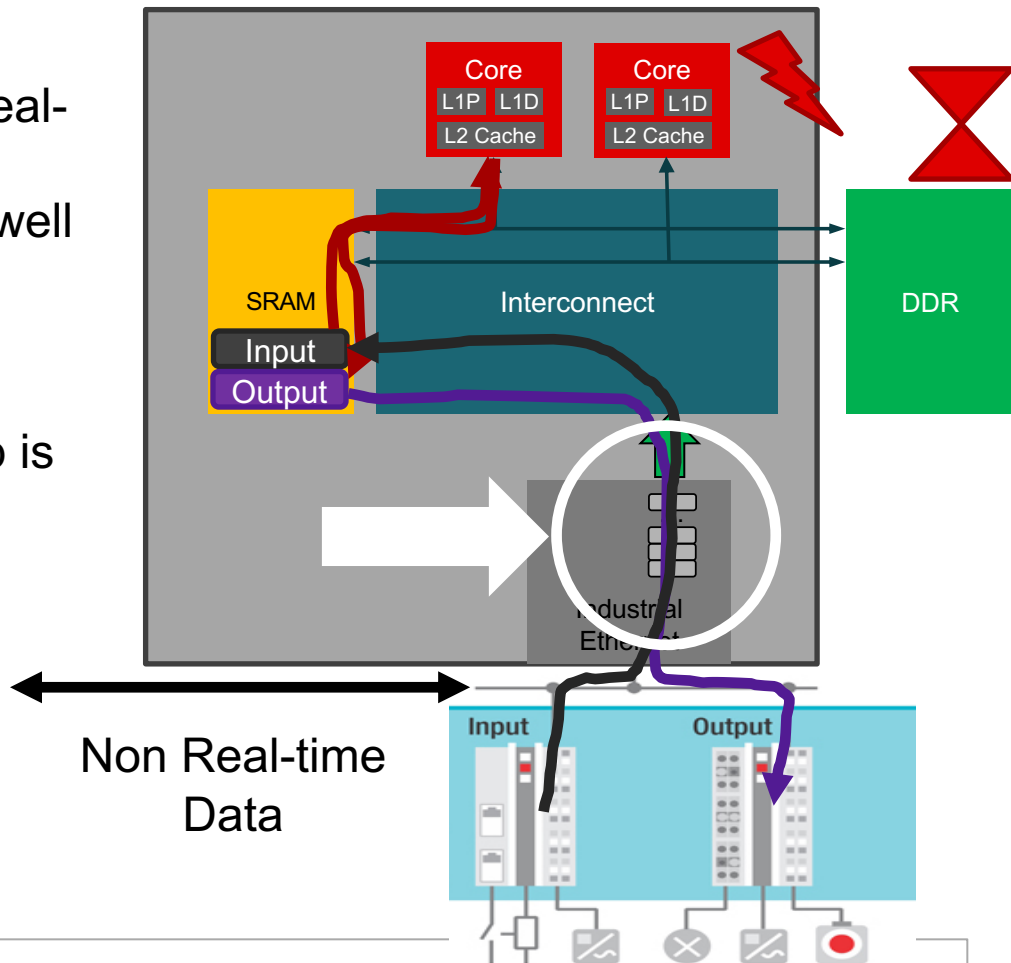


# What about other traffic?

- Modern systems can't just focus on real-time traffic
- Other data needs to be managed as well
- Any data that needs to flow through system but is not time bound
  - Management and Metadata
  - Not used for Closed Loop, or loop is very relaxed

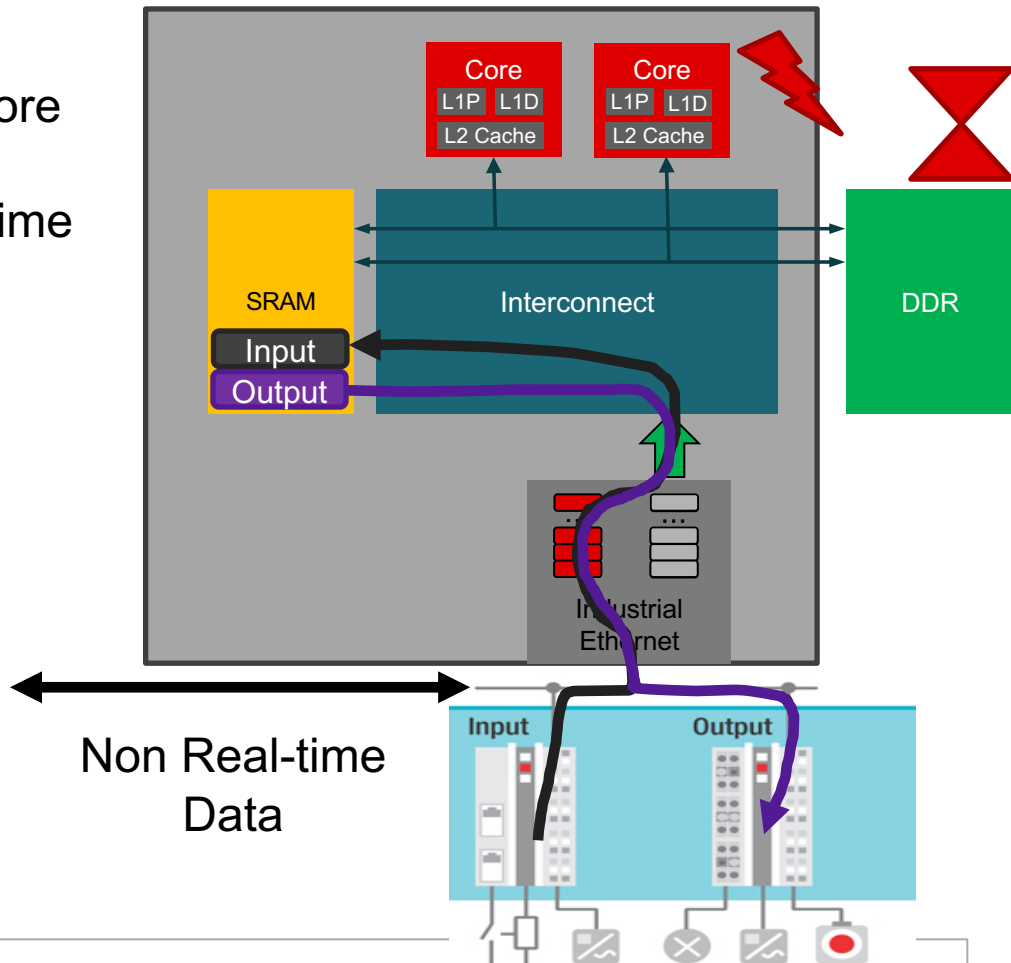
Can this data cause problems with real-time data?

A: Yes! Data clogging queues, for example



# Add More Queues

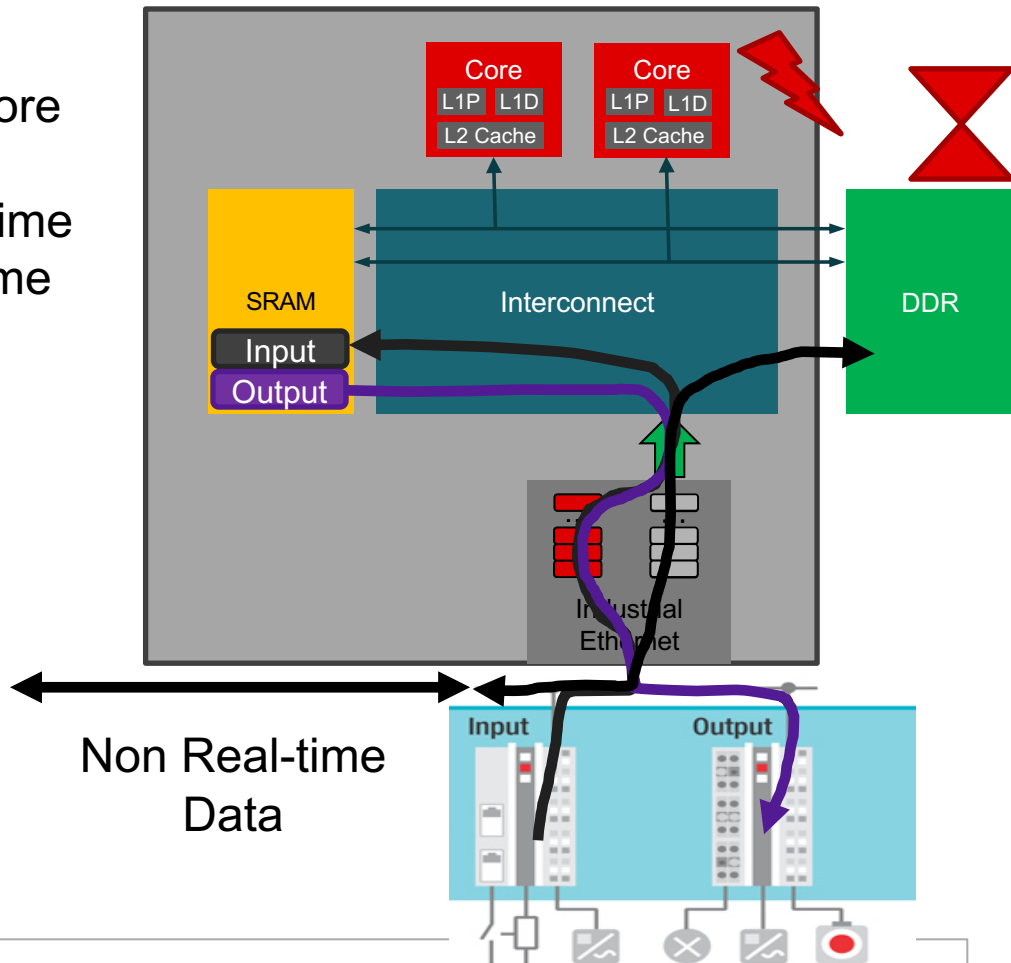
- Much like with CPUs and Memory, more queues are needed
- One queue can be reserved for real-time





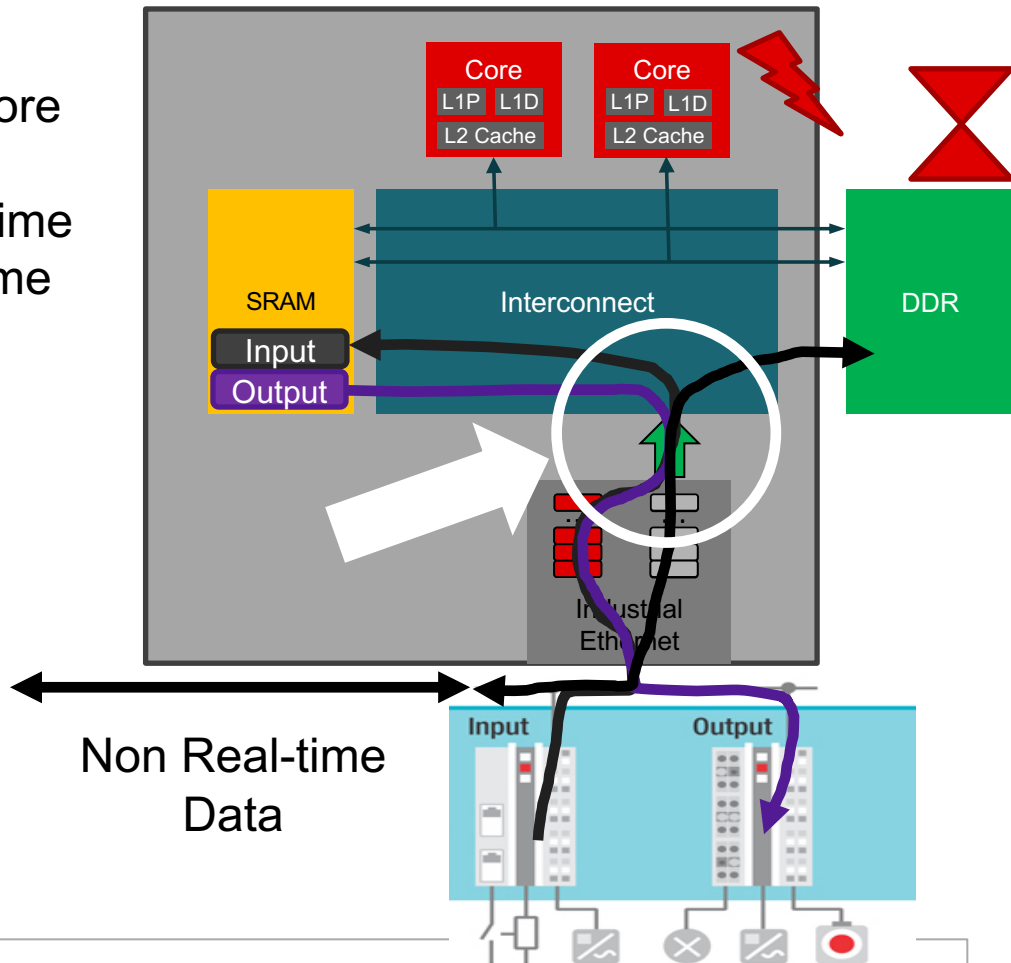
# Add More Queues

- Much like with CPUs and Memory, more queues are needed
- One queue can be reserved for real-time
- The other can be used for non real-time
- Non real-time can be stored in DDR



# Add More Busses

- Much like with CPUs and Memory, more queues are needed
- One queue can be reserved for real-time
- The other can be used for non real-time
- Non real-time can be stored in DDR

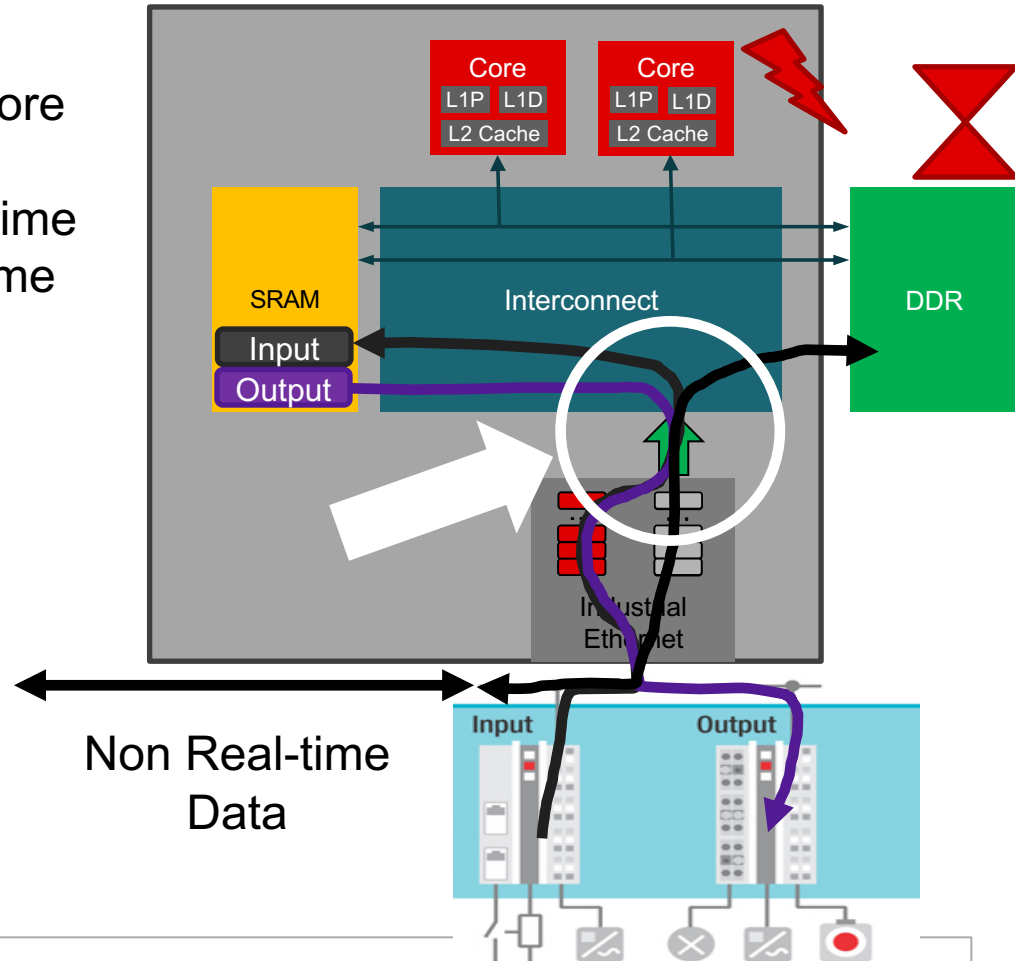


What's going on here?

Non Real-time  
Data

# Add More Busses

- Much like with CPUs and Memory, more queues are needed
- One queue can be reserved for real-time
- The other can be used for non real-time
- Non real-time can be stored in DDR
- More busses needed to move data

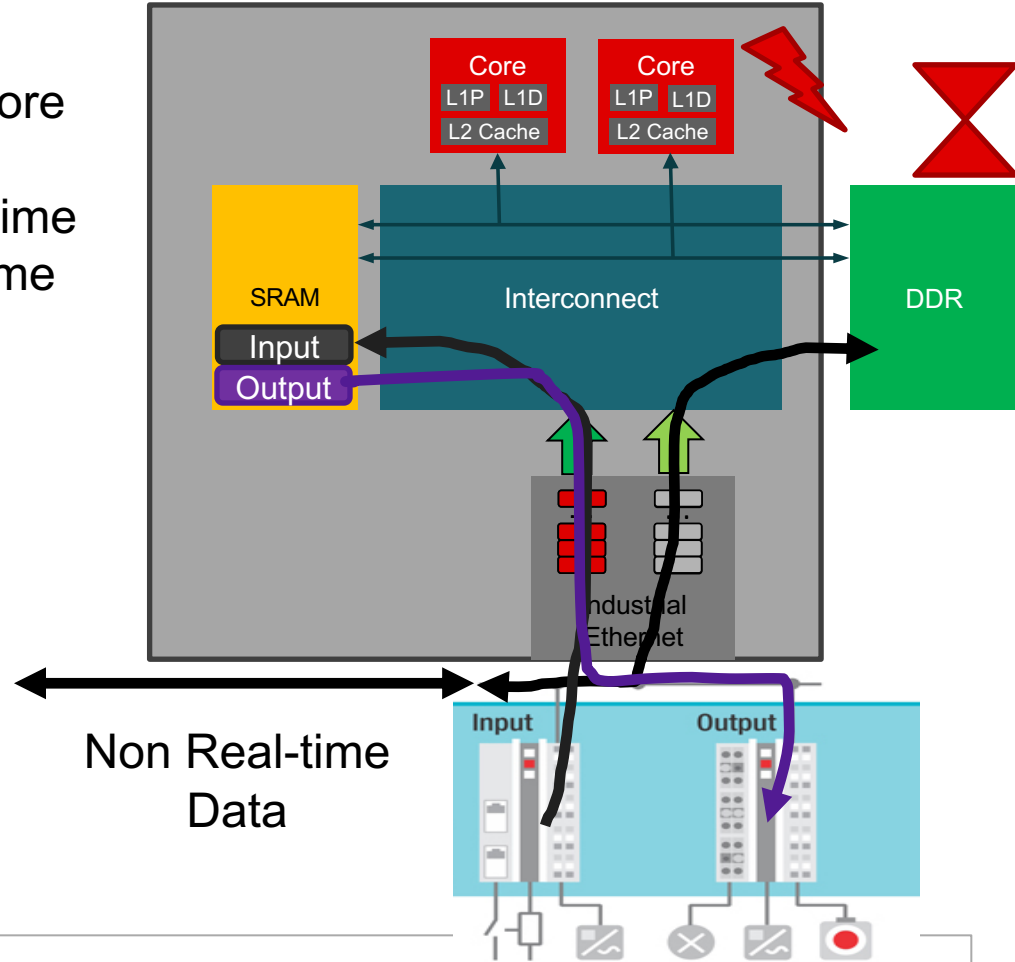


What's going on here?

Non Real-time  
Data

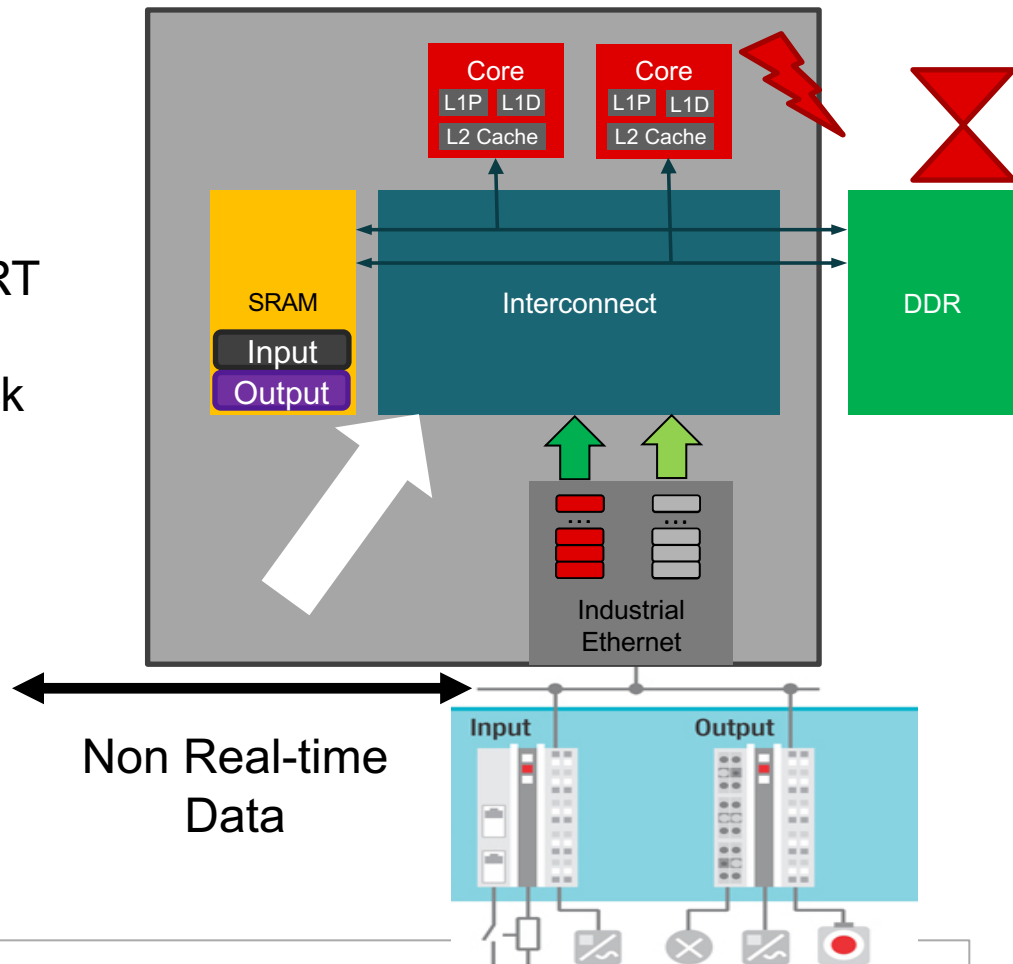
# Add More Busses

- Much like with CPUs and Memory, more queues are needed
- One queue can be reserved for real-time
- The other can be used for non real-time
- Non real-time can be stored in DDR
- More busses needed to move data

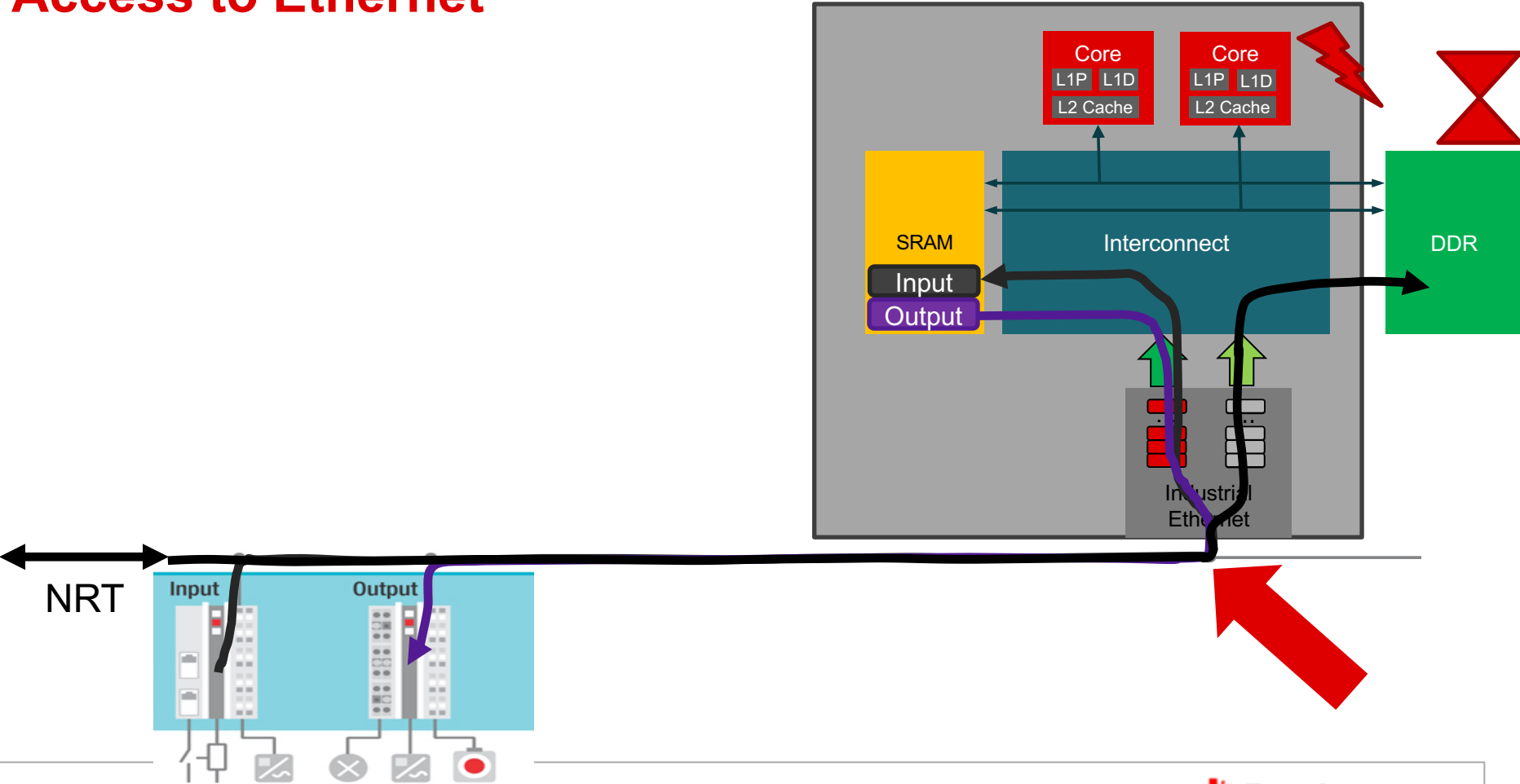


# Interconnect Flexibility

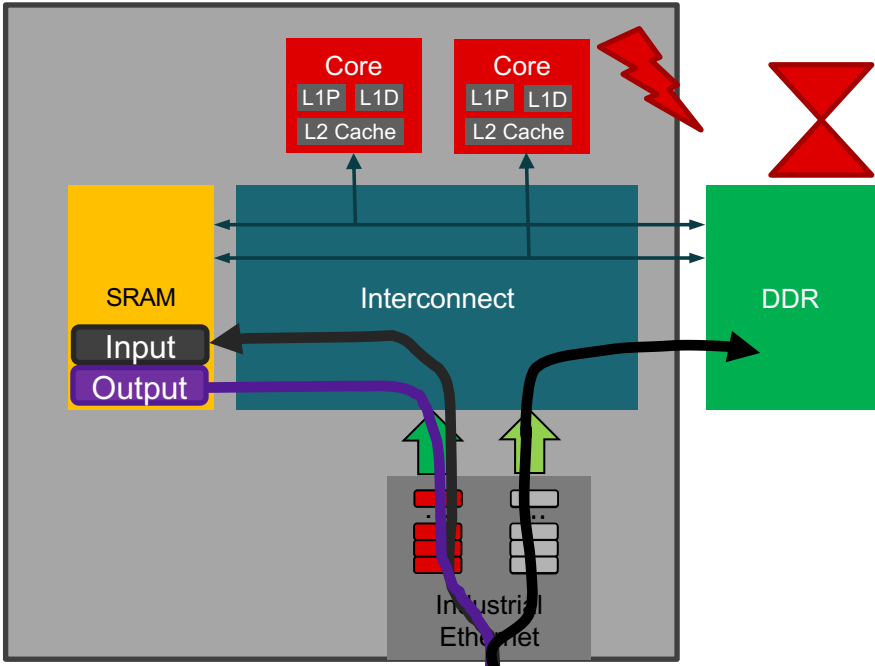
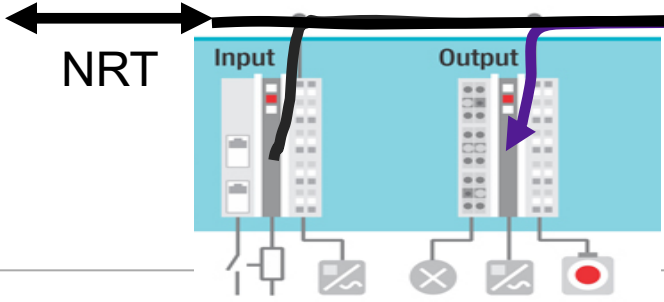
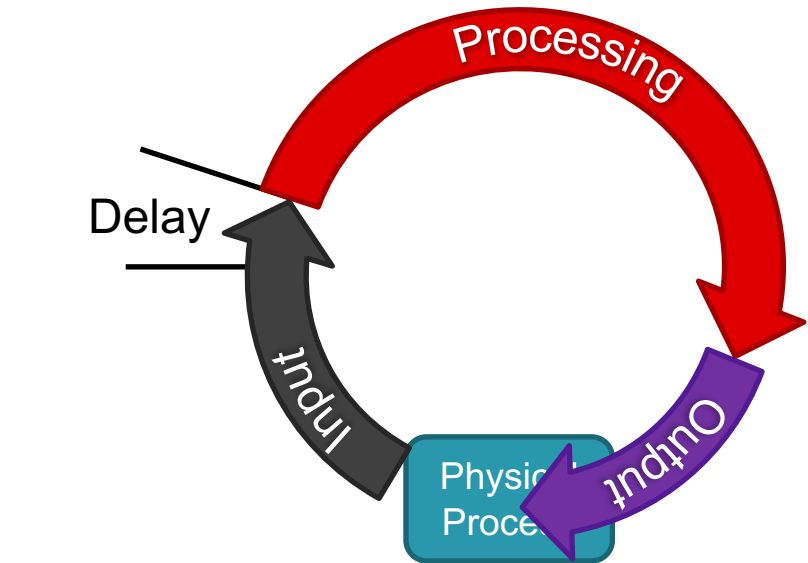
- Connect many points
- Understand non real-time (NRT) and real-time (RT) needs
  - Move RT data concurrent with NRT data
  - Can't allow large transfers to block RT data



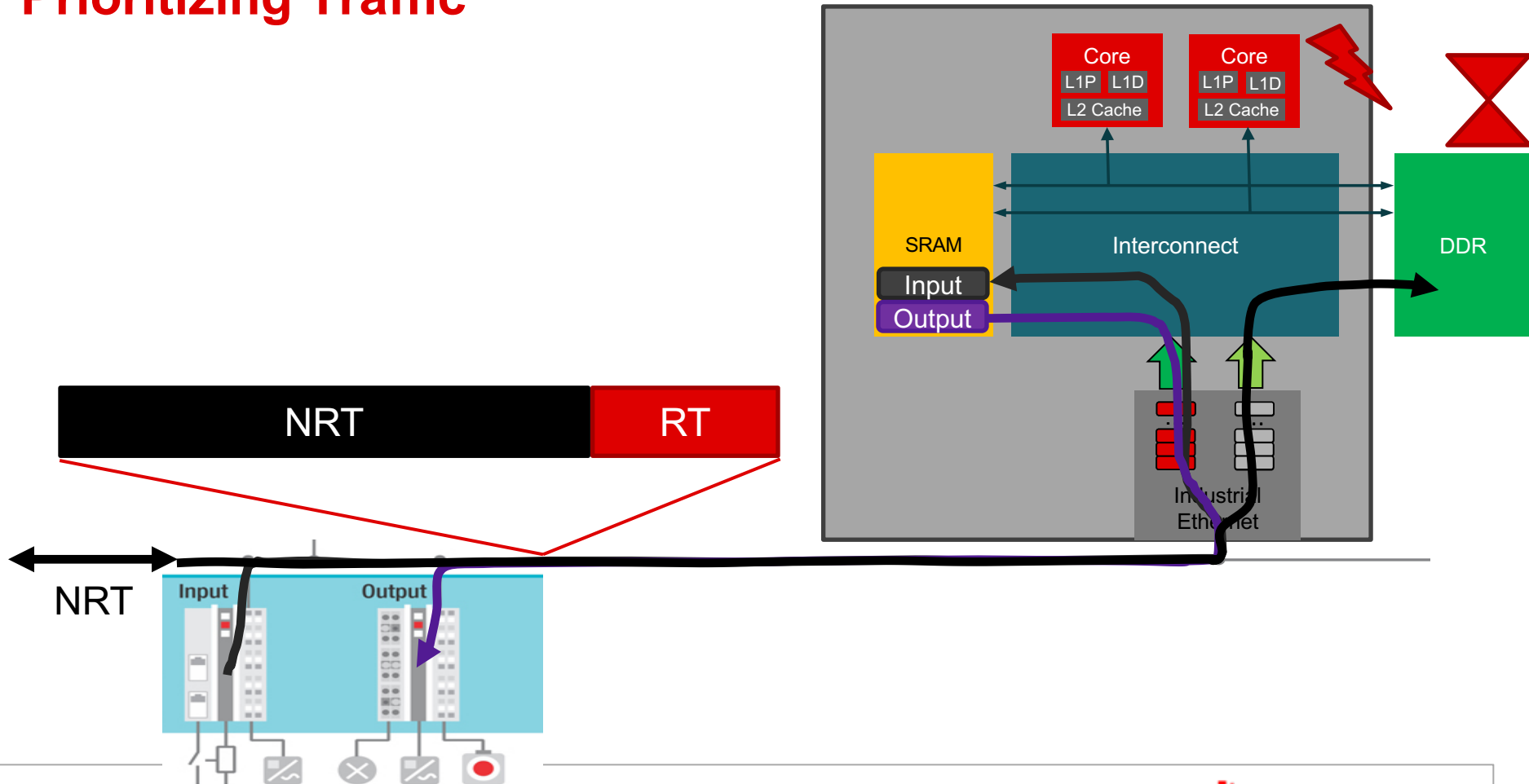
# Access to Ethernet



# Ethernet Delays

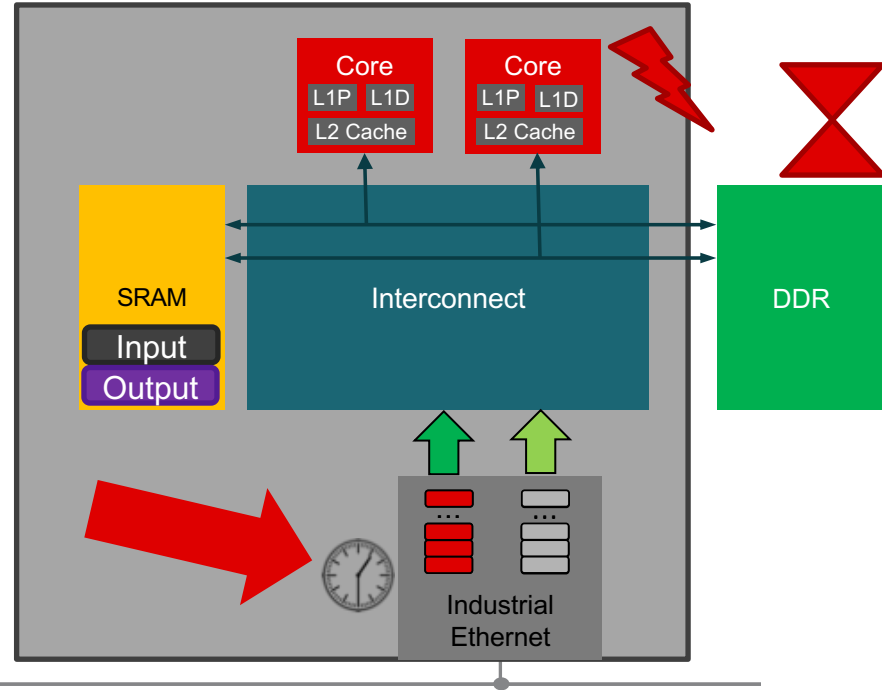
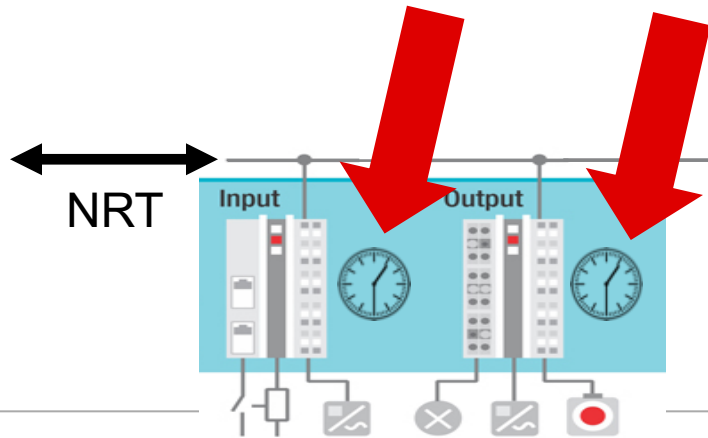


# Prioritizing Traffic



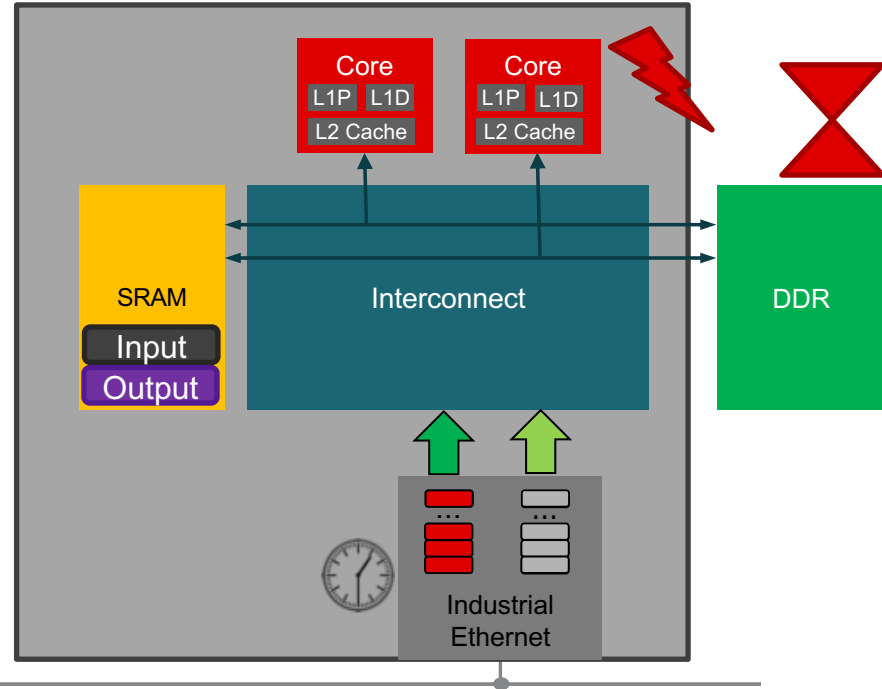
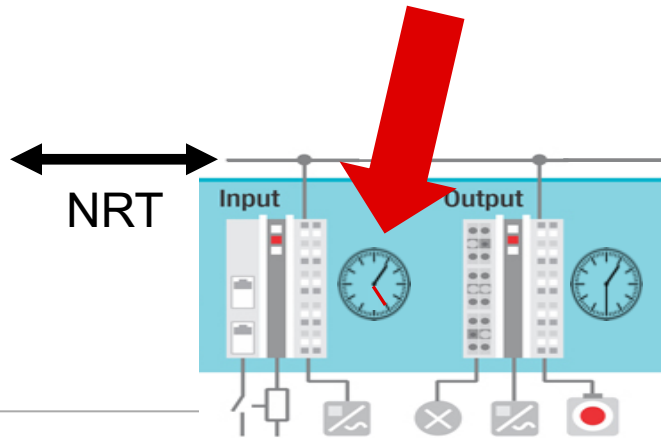


# Synchronization



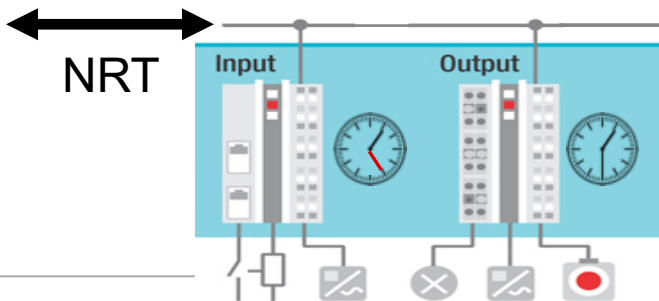
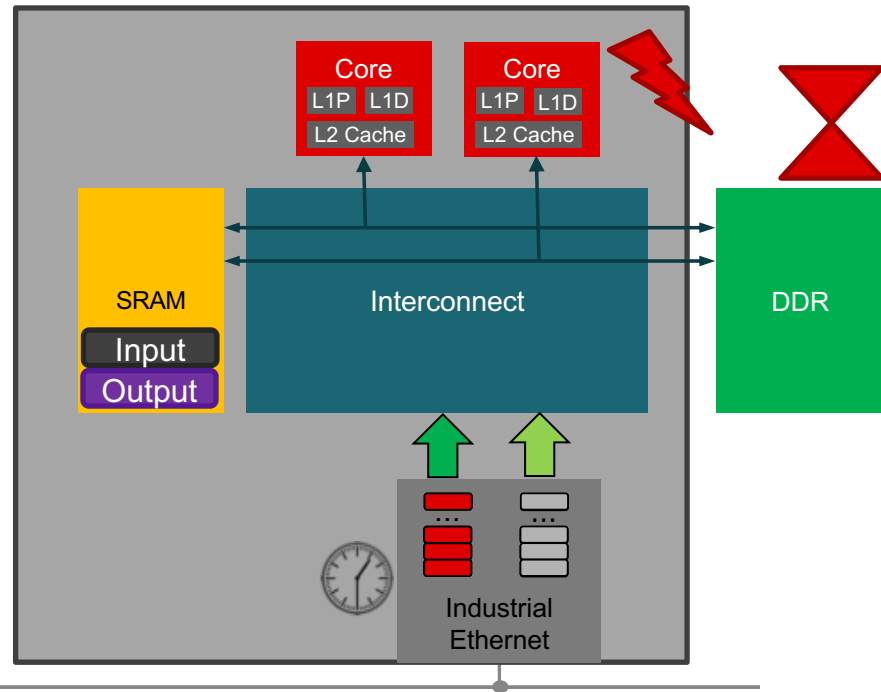
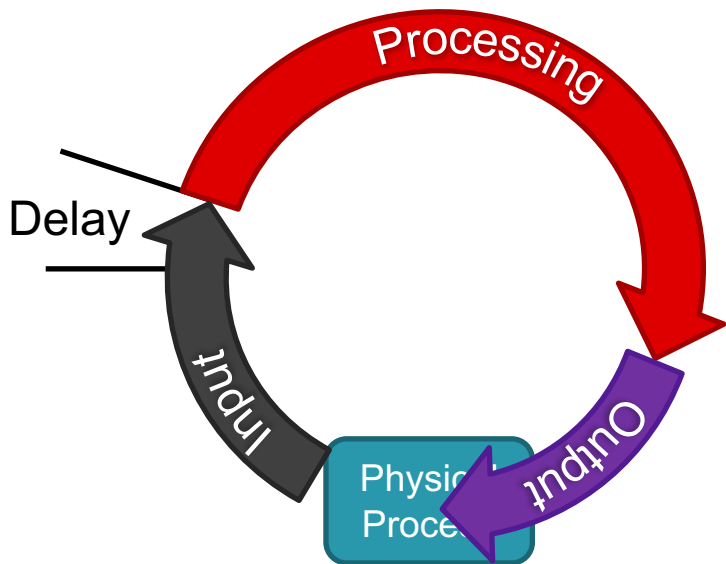
- Everyone agrees to do something at the same time

# Not Synchronized



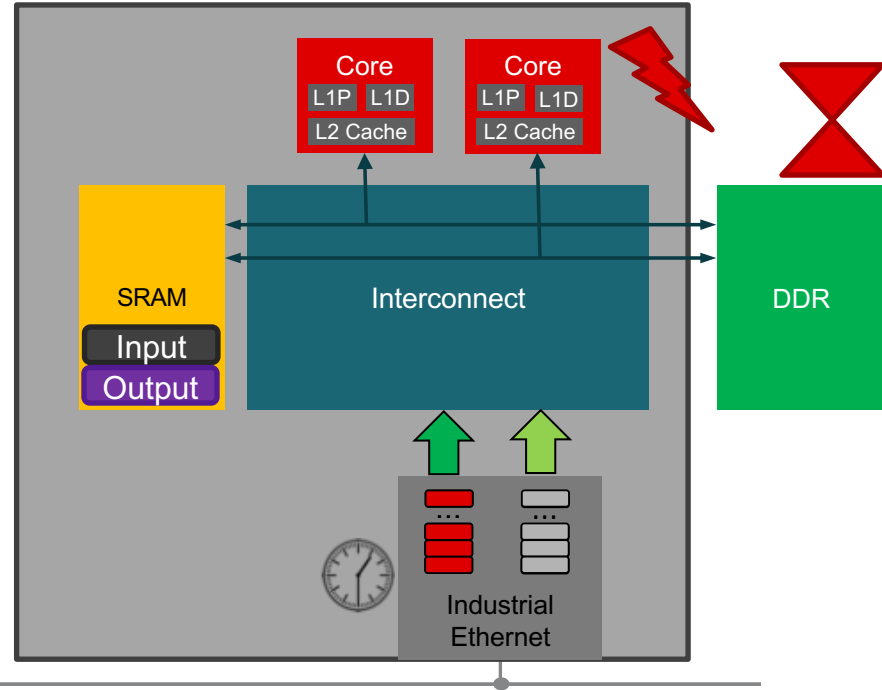
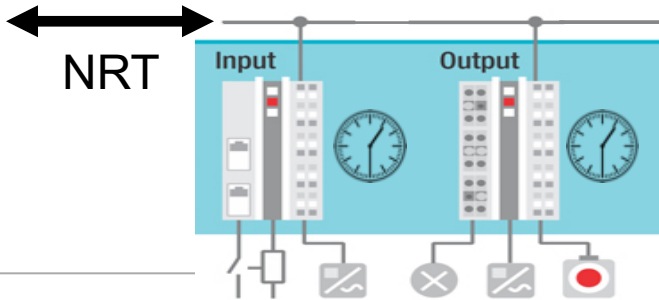
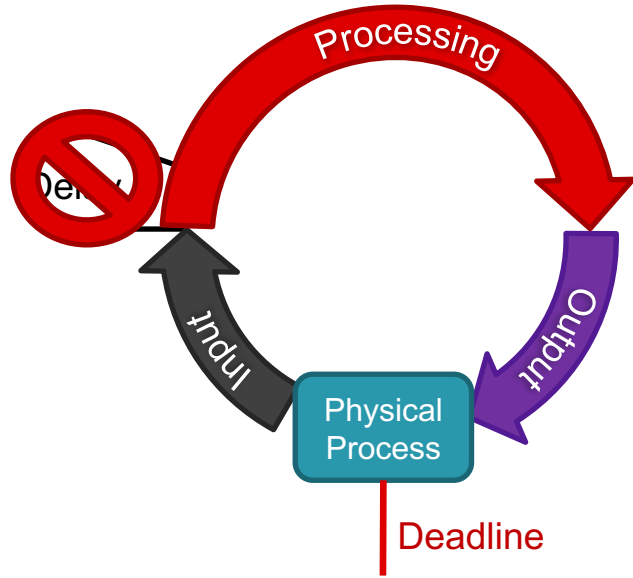
- Everyone agrees to do something at the same time
- But, Input has a slow watch...

# Unsynchronized Delay



- Everyone agrees to do something at the same time
- Input has a slow watch...
- Input read late relative to the rest of system...delay!!

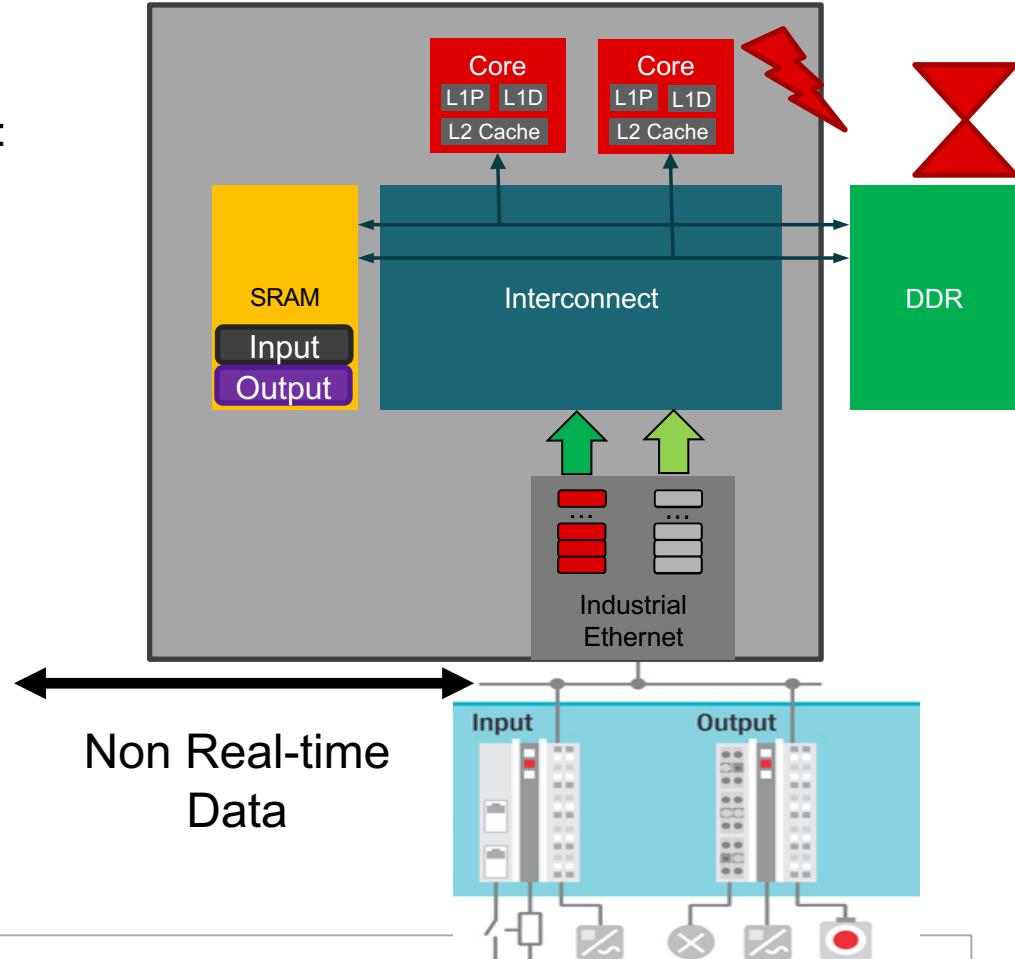
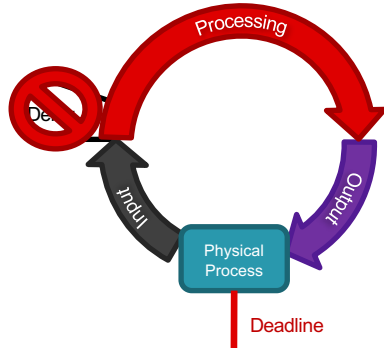
# Synchronize Watches



- Everyone agrees to do something at the same time
- Input has a slow watch...
- Input read late relative to the rest of system...delay!!
- Synchronize watches to avoid delays

# Summary

- Manage delay with determinism
- For Determinism, add RT and Non-RT for:
  - Cores
  - Memory
  - Packet Management (Queues)
  - Busses
  - Interconnect
  - Industrial Ethernet
  - Time Synchronization



# For more information...

- [AM654x Training Series](#)
- [AM654x on the web](#)
  - [Data Sheet](#)
  - [Technical Reference Manual](#)
- [Processor Software Development Kit \(SDK\)](#)
- [Evaluation Module](#)
- [Industrial Development Kit](#)
- [Support Forum](#)



© Copyright 2019 Texas Instruments Incorporated. All rights reserved.

This material is provided strictly “as-is,” for informational purposes only, and without any warranty.  
Use of this material is subject to TI’s **Terms of Use**, viewable at [TI.com](https://www.ti.com)