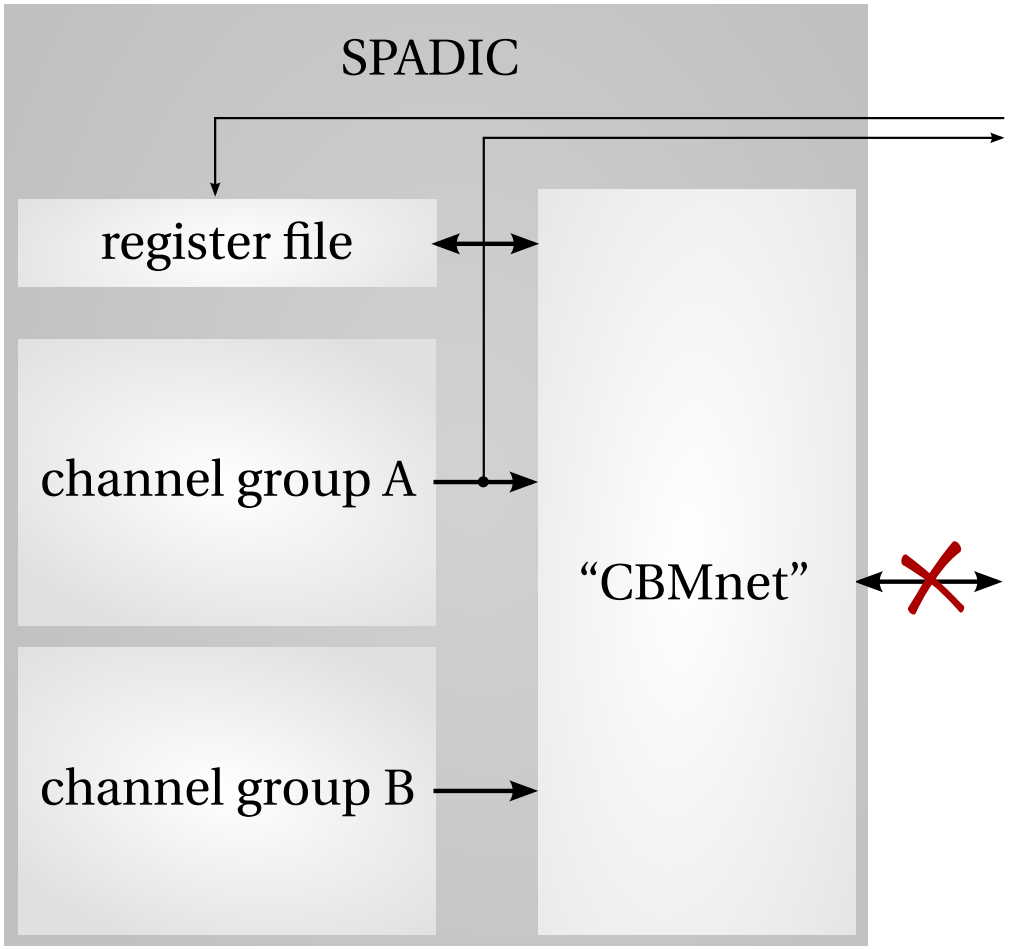
# SPADIC-Susibo communication using CBMnet

Michael Krieger

SuS Meeting, 26.02.2013

# SPADIC-Susibo communication so far

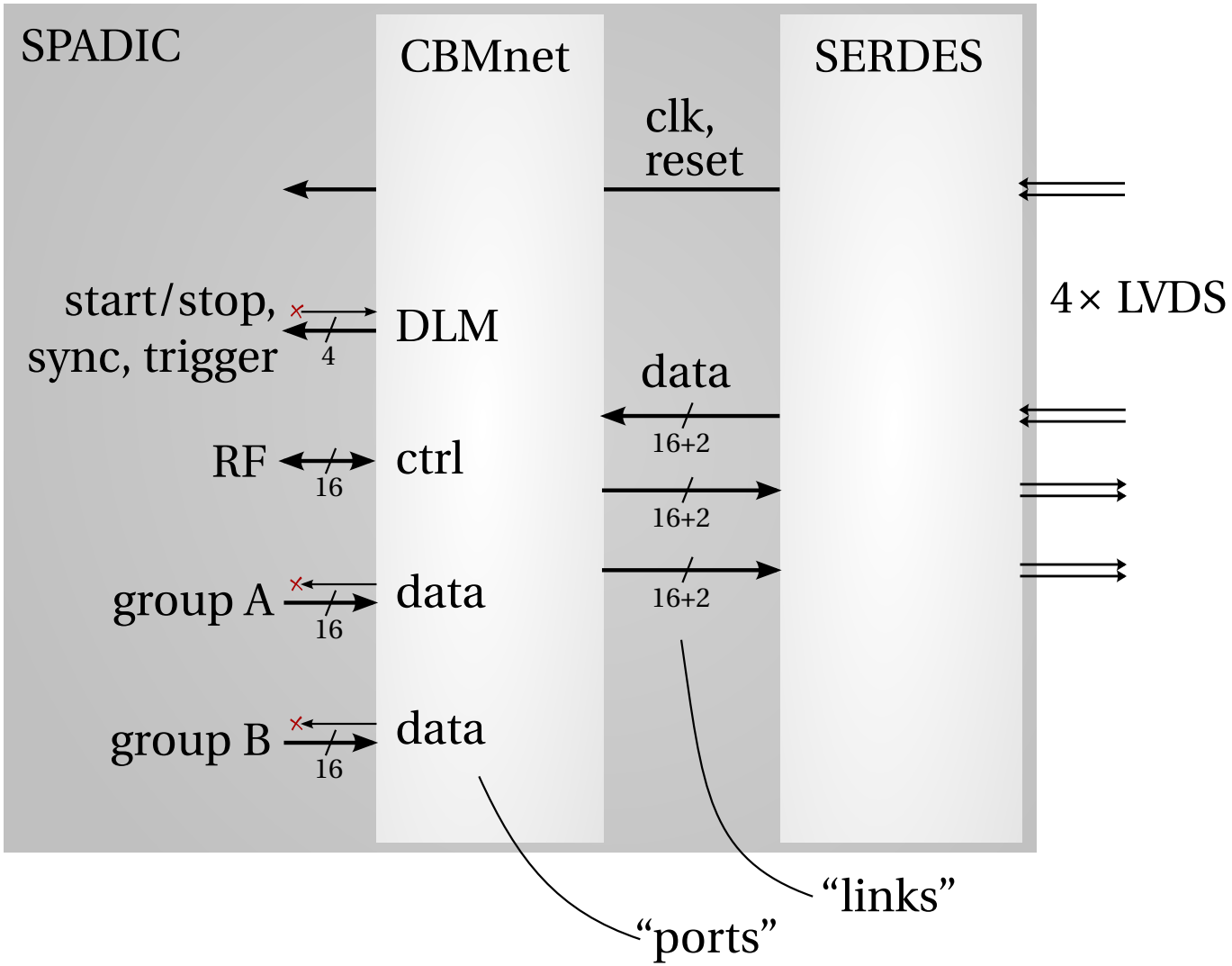



Limitations:

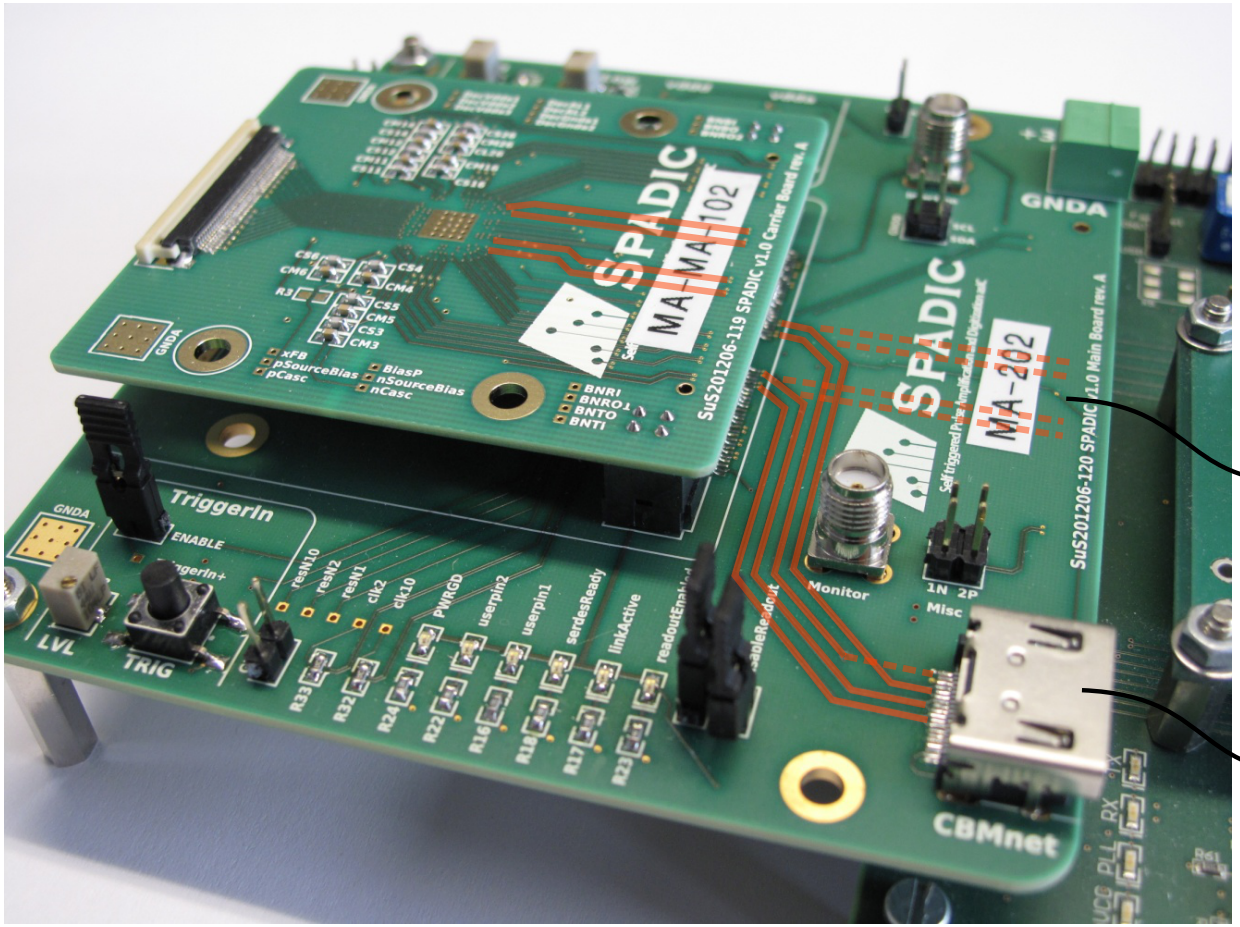
- unreliable (connection breaks down at high data rates)
- cannot read data from group B
- cannot read register values
- epoch channels need DLMs (CBMnet feature)

Susibo needs to
“speak CBMnet”

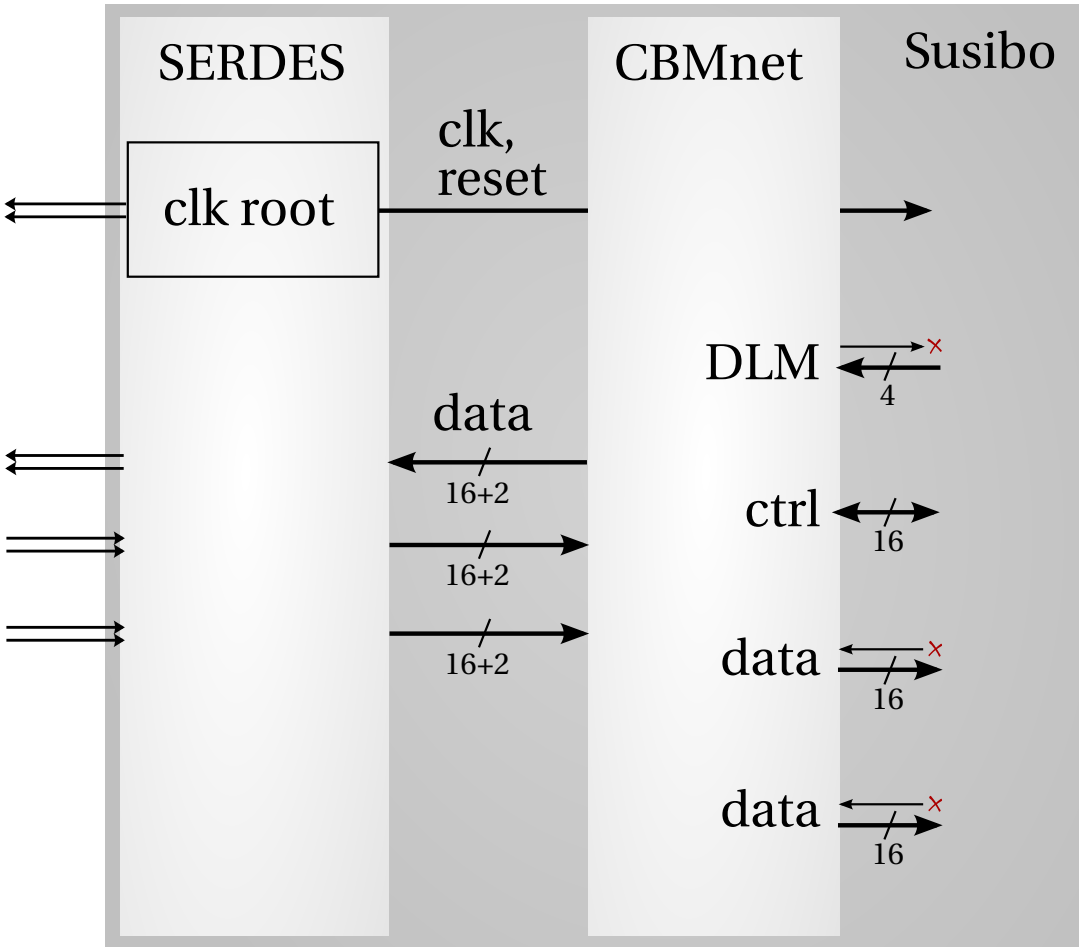
# “CBMnet” in more detail



# CBMnet routing

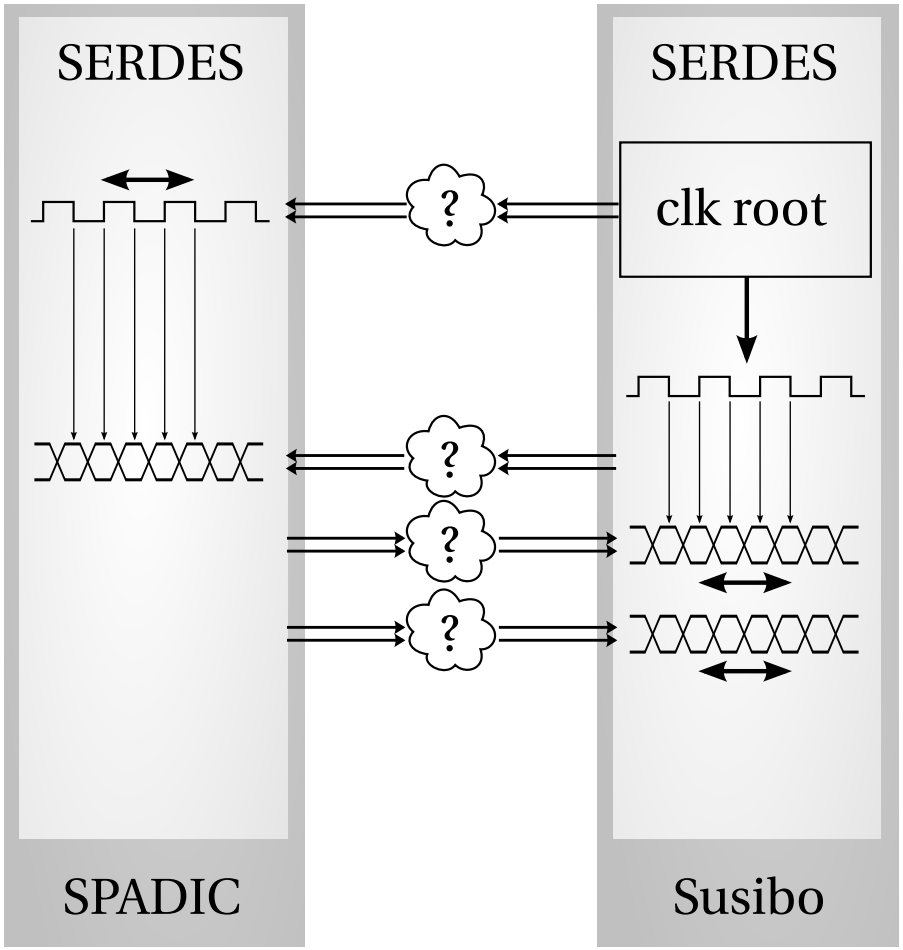


# Susibo point of view




- CBMnet block provided (should be generic enough)
- SERDES block: SP605 implementation provided—not generic enough → need to adapt for Susibo

# Signal delays




250 MHz bit clk, DDR → 2 ns/bit

ensure correct clock/data alignment

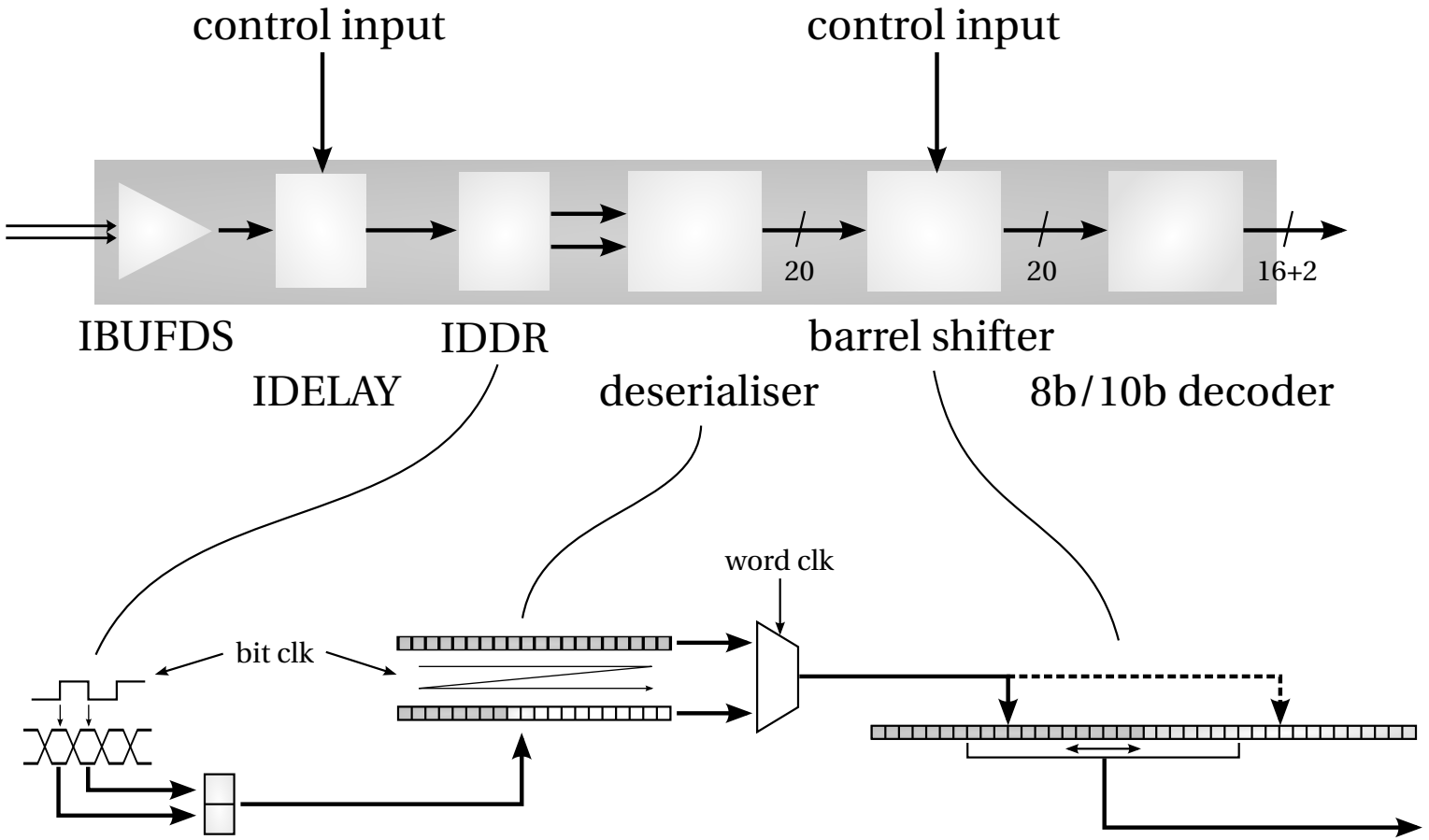
- on both sides
- automatically (independent of PCB routing, cable length, bitfile, ...)

Virtex 5 provides

- IDELAY (variable)
- ODELAY (fixed)
- DCM (variable)

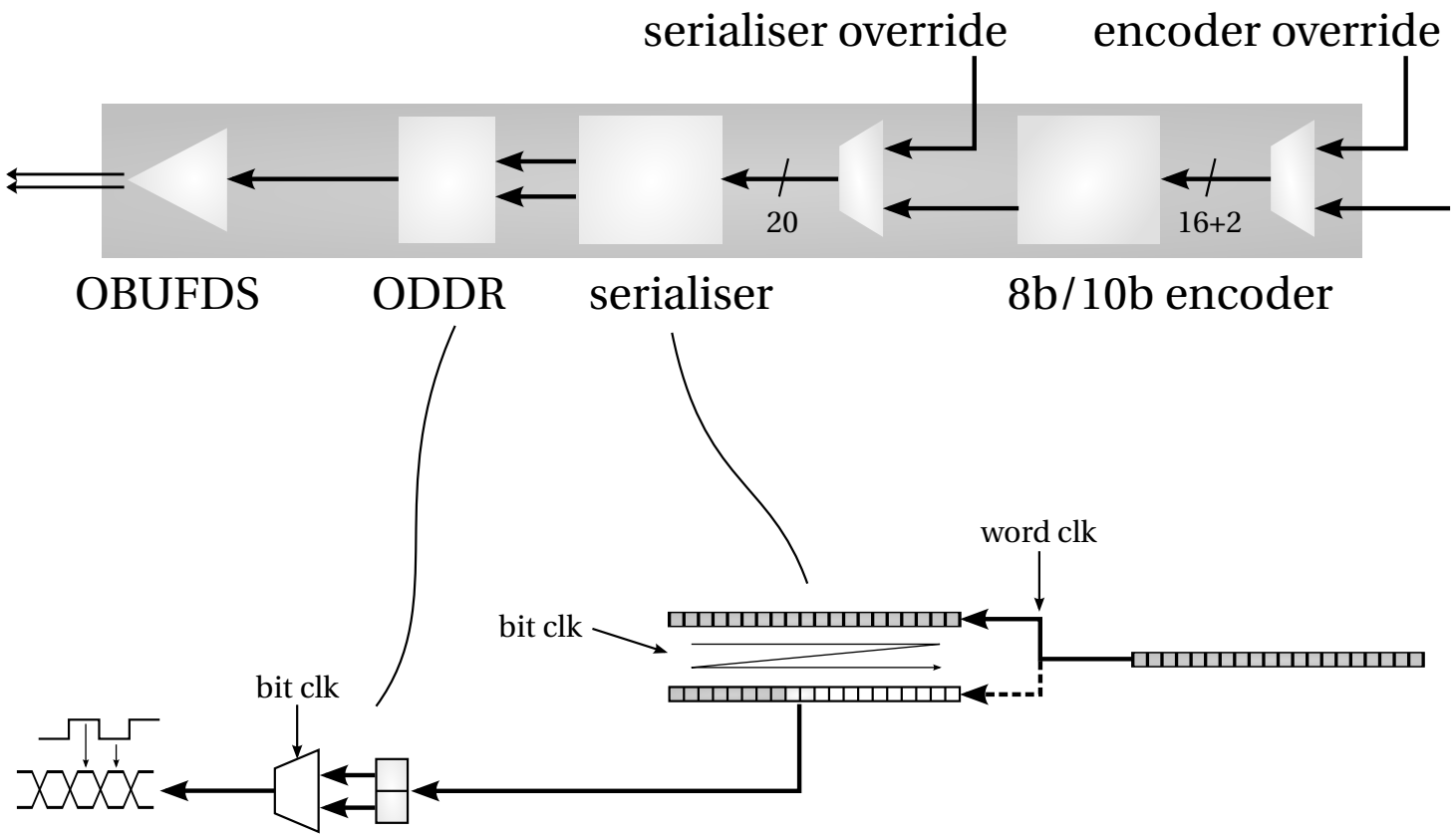
# SERDES building blocks

receiver module:



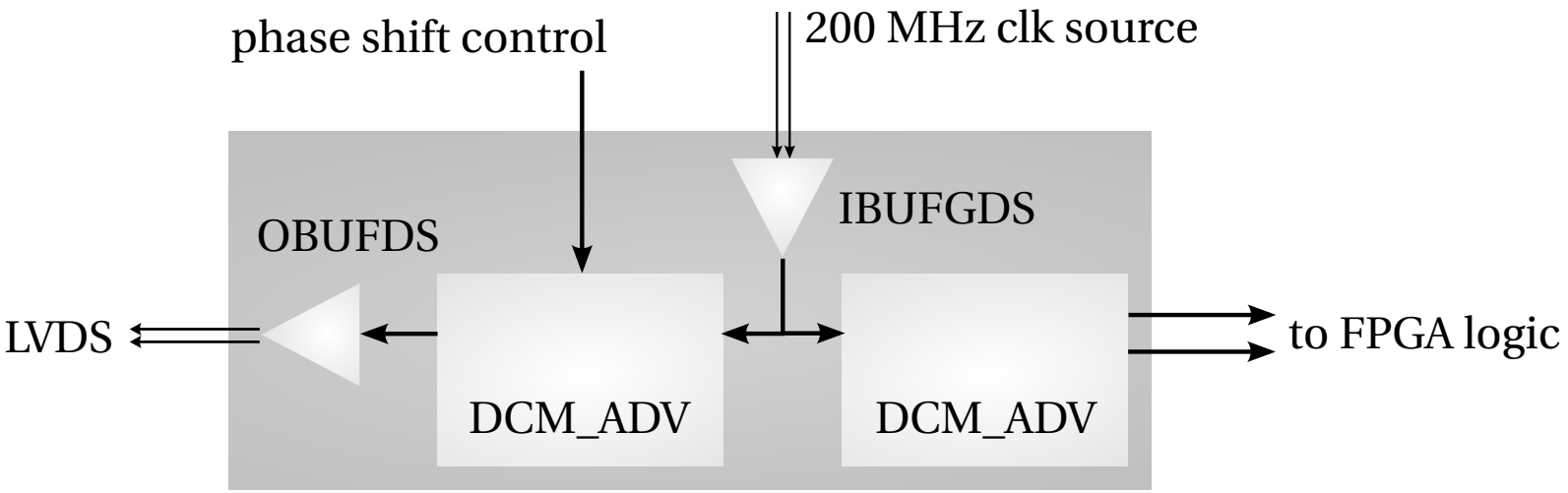
# SERDES building blocks

transmitter module:



# SERDES building blocks

clock root:

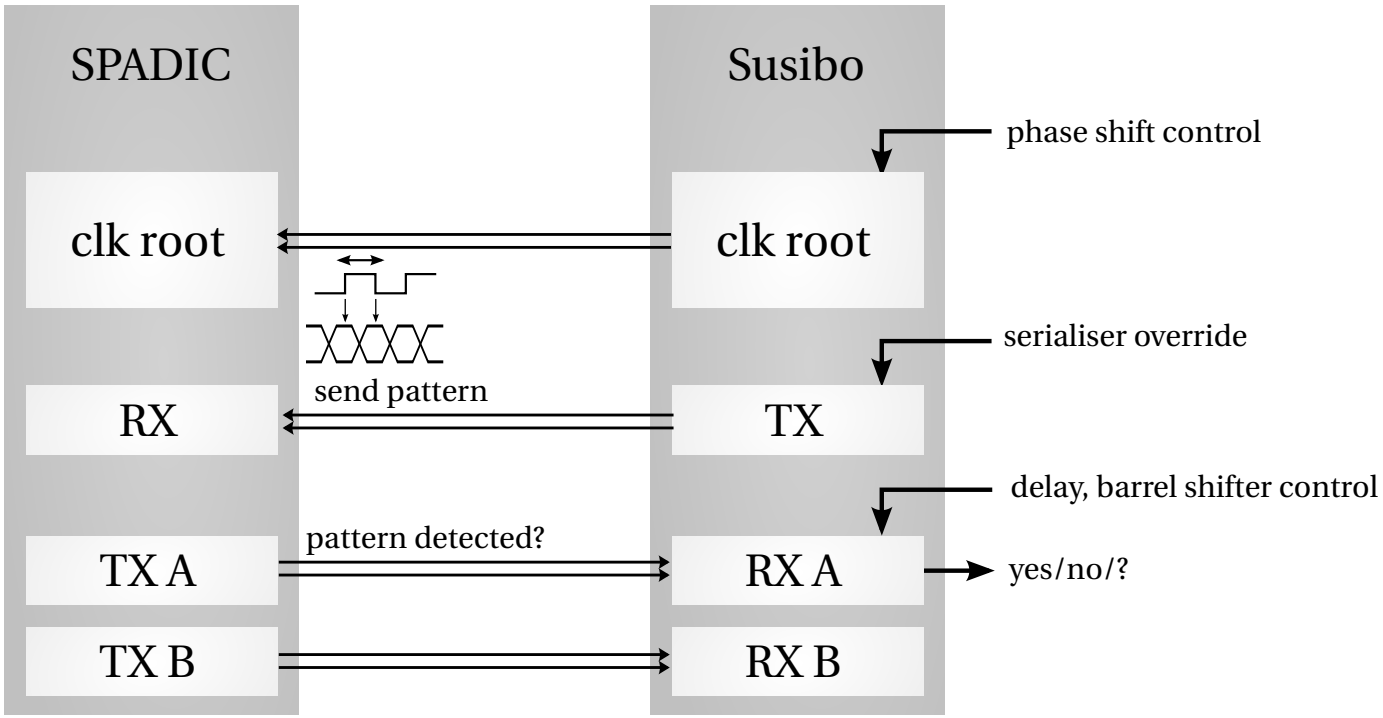



currently: 200 MHz bit clock, 20 MHz word clock

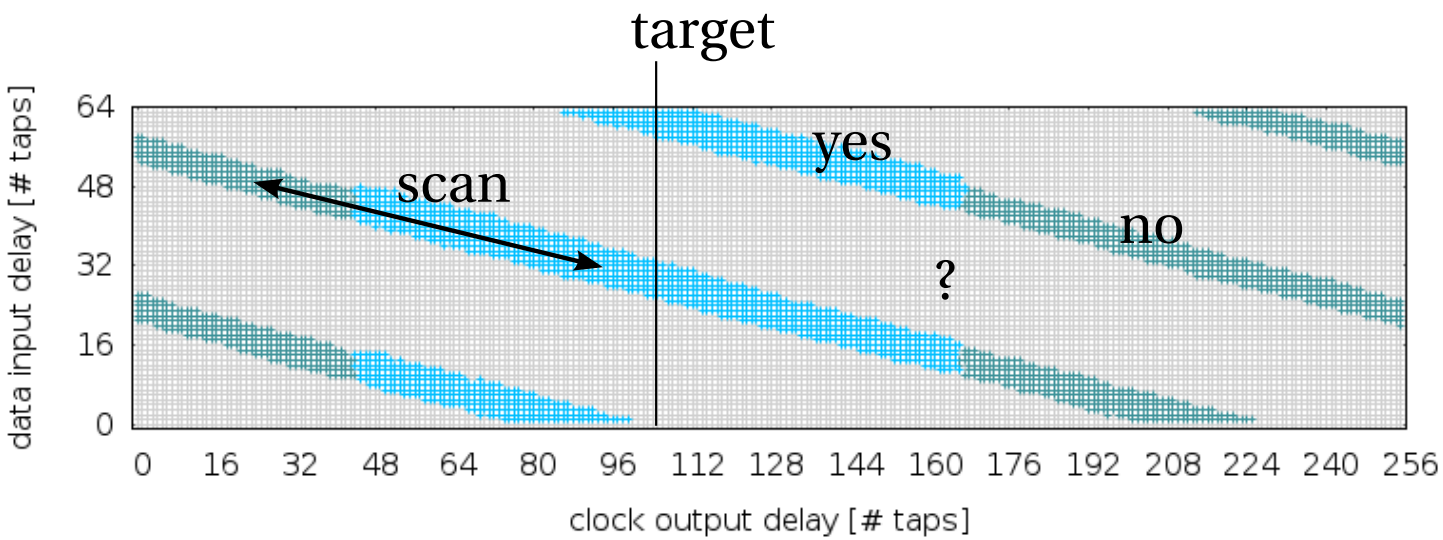
# clock phase shift adjustment

## send pattern to SPADIC

- is the pattern received?
- can we receive the answer?

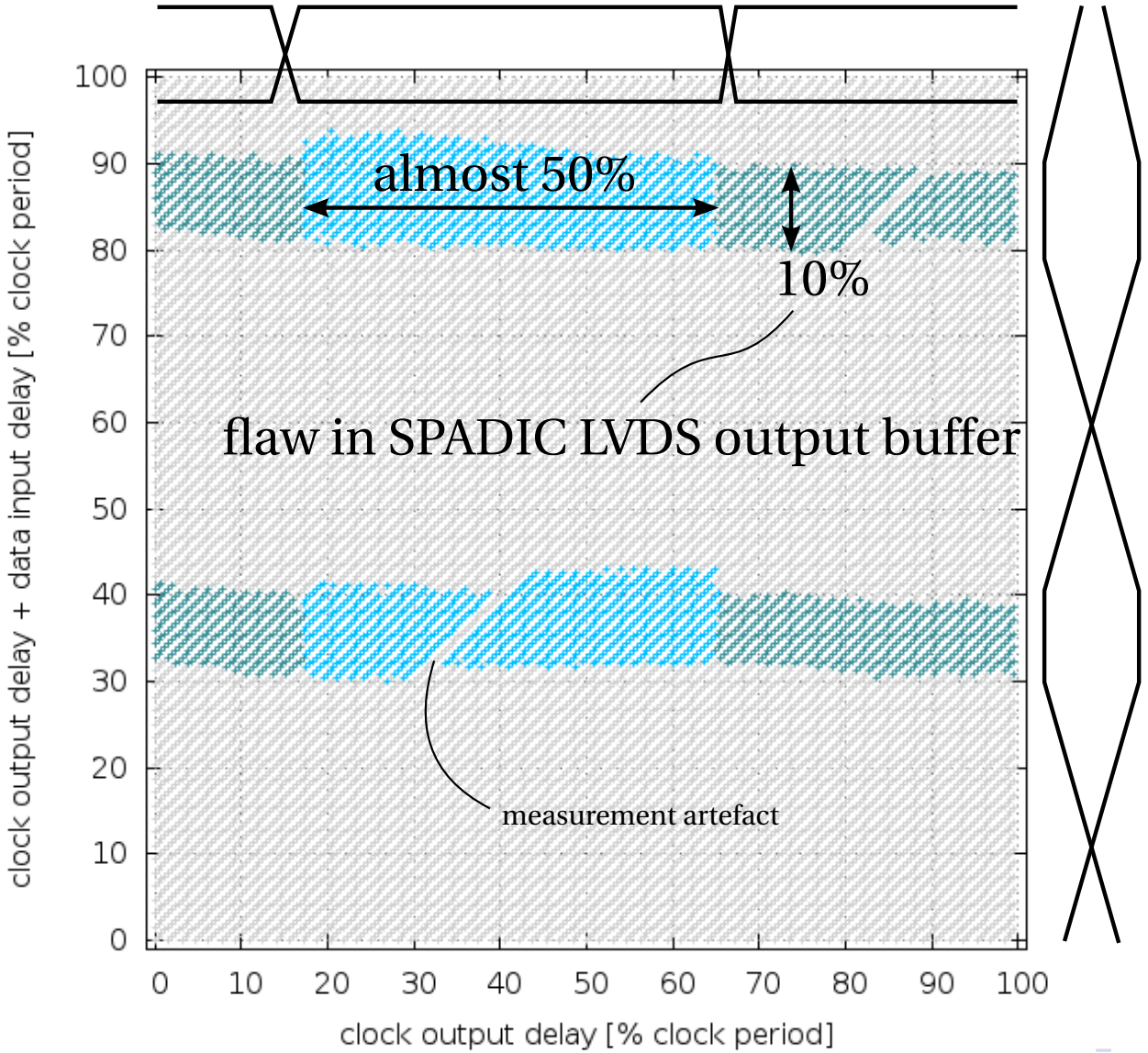


# clock phase shift adjustment

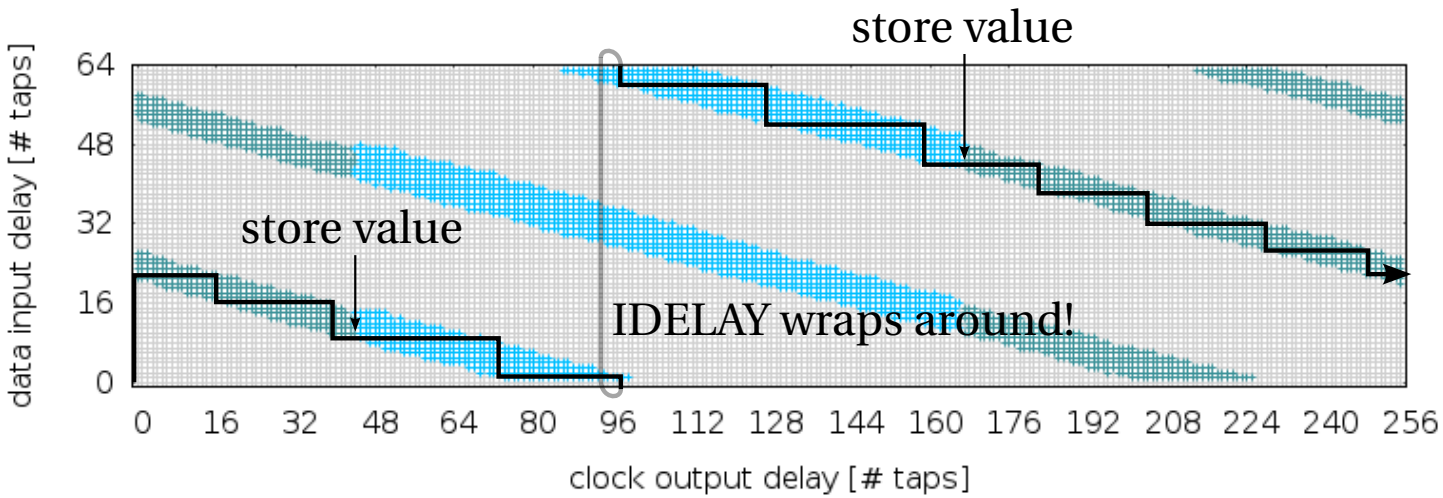



- scan for yes/no border
- go to center
- how to stay inside (slope)?

# side note: LVDS signal quality



# clock phase shift adjustment



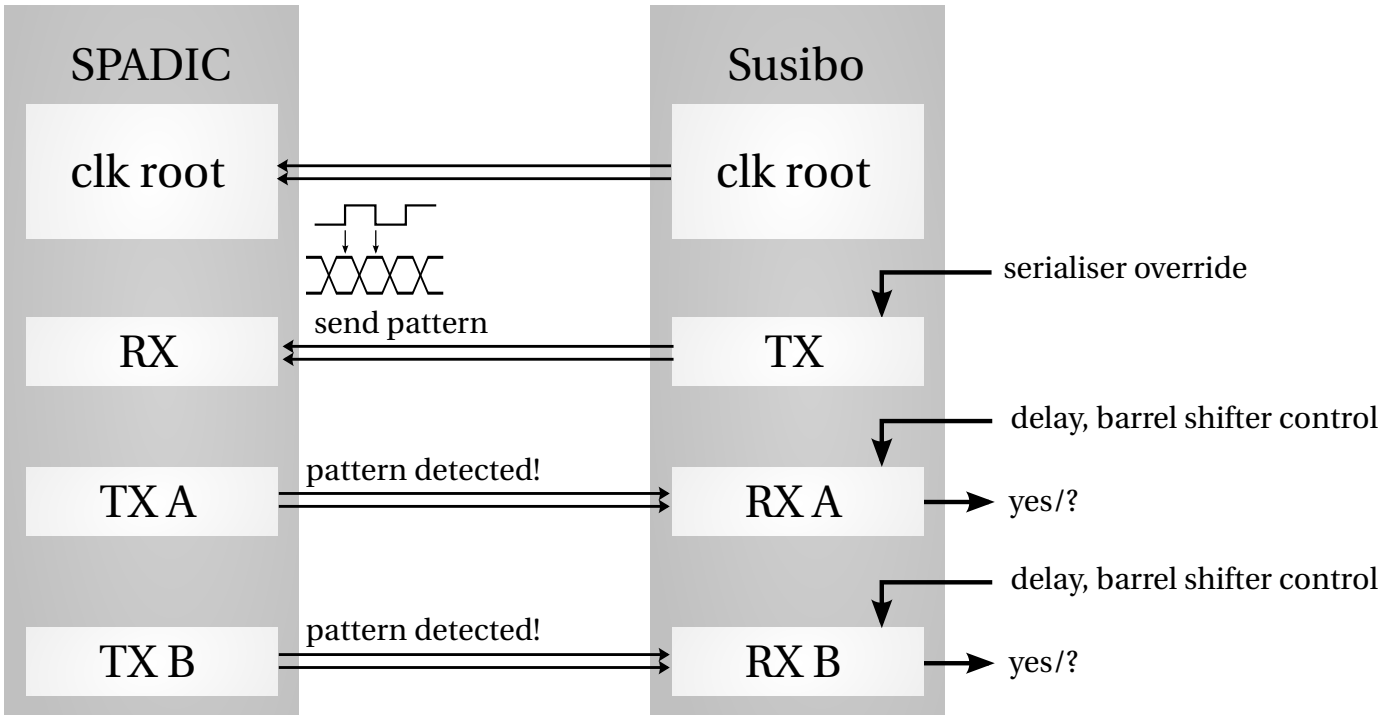

relatively simple and fail-safe algorithm:

- go right/down until outside “yes”/“no”
- independent of slope (no need to even know the slope)
- DCM phase shift range only scanned once (does *not* wrap around → rewind logic saved)
- scan completely → end position known

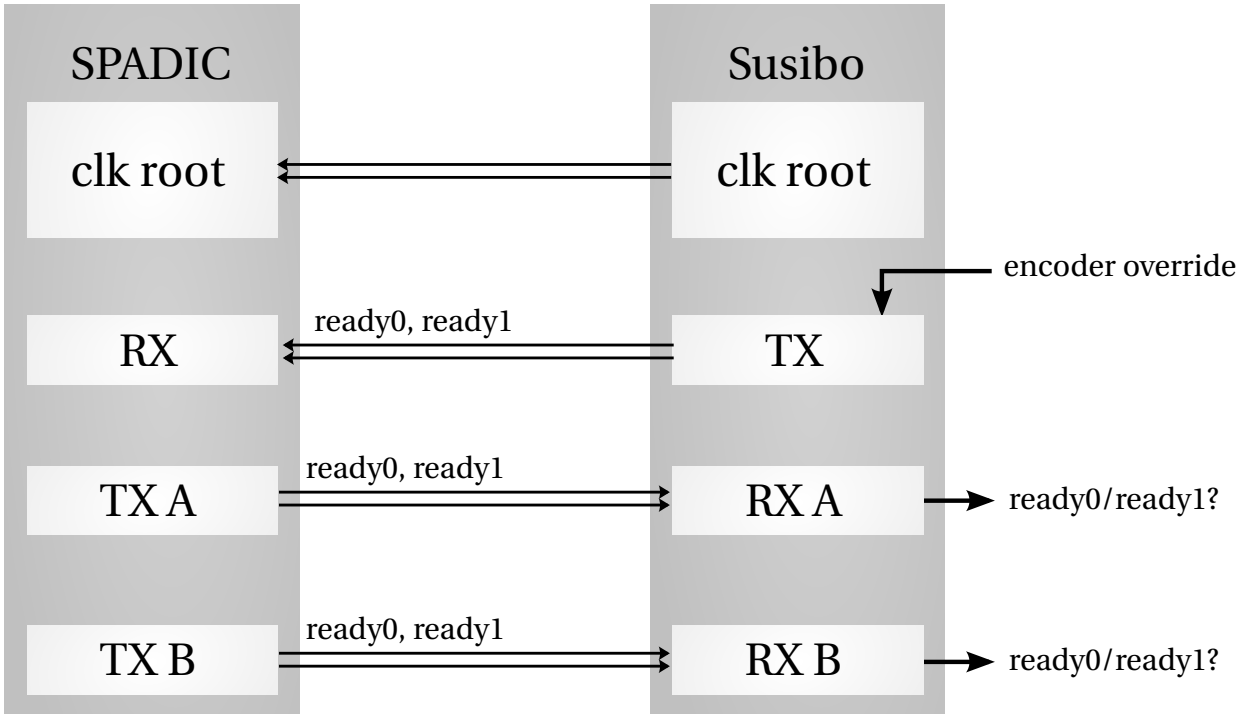
# input delay adjustment

## send pattern to SPADIC

- the pattern should now be received
- can we receive the answer?

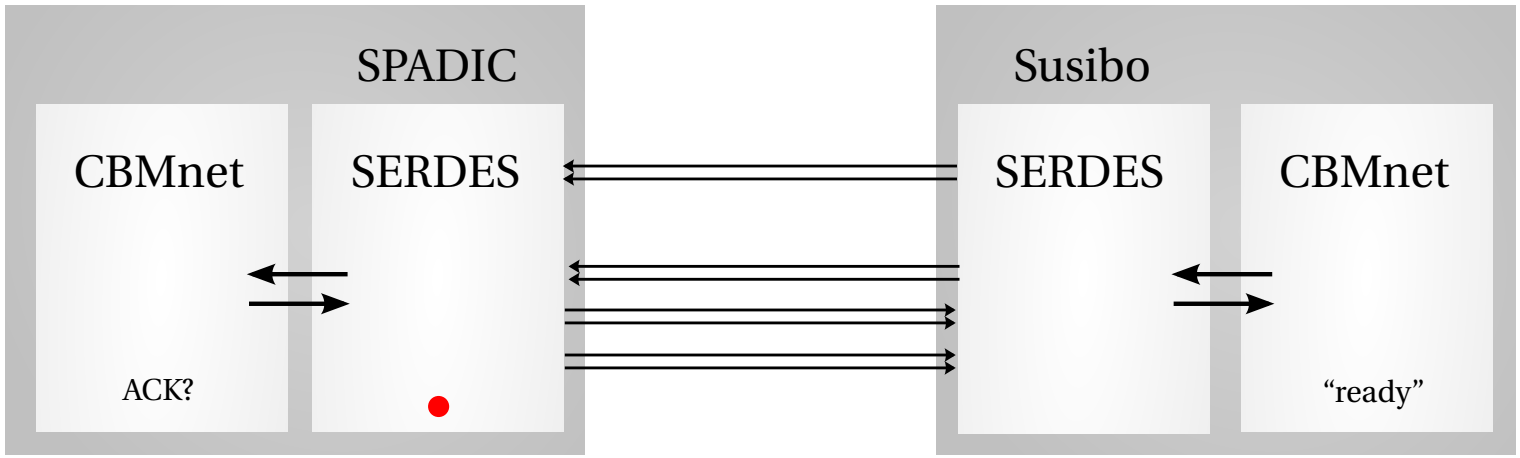


# SERDES handshake



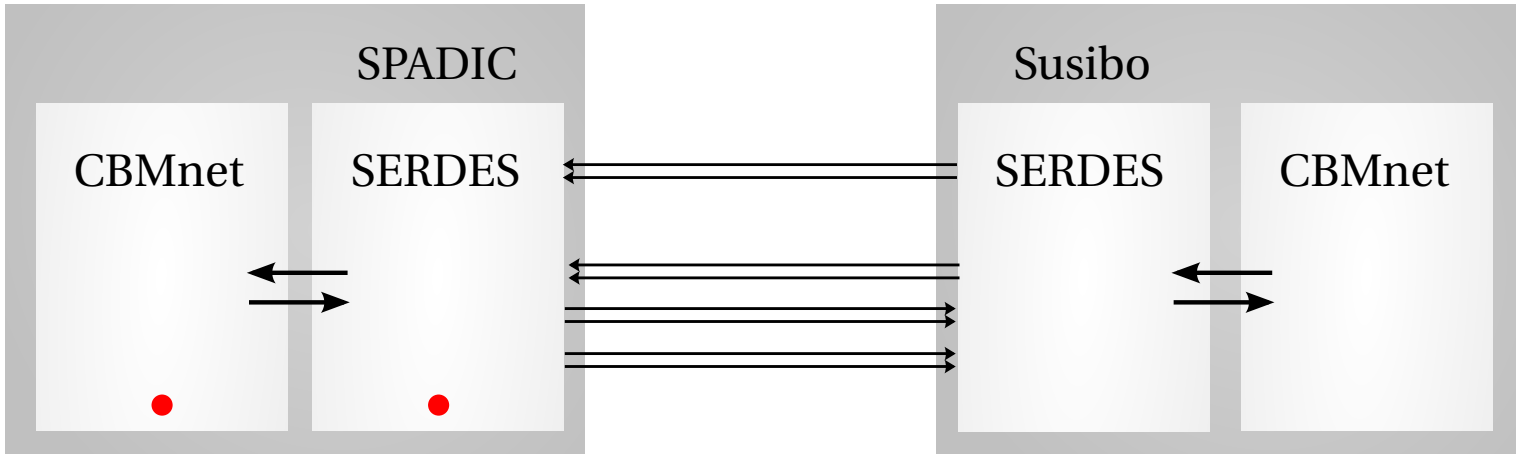

- send ready0 → SPADIC stops sending “yes”/“no”
- wait for incoming ready0 characters
- send ready1, wait for ready1 → finish

# CBMnet initialization




- after SERDES is ready (LED), CBMnet can use it
- again handshake
  - SPADIC-side handshake FSM often stuck in infinite loop: waiting for “ACK”
  - Susibo FSM assumed “ACK” was sent, but it wasn’t
  - handshake specification?
  - cannot decide if wrong concept or wrong implementation → hack: send some more “ACK”s…

# Status




SPADIC signals “CBMnet ready” most of the time (LED)

- first experiment: write register file
- works *almost* always (why not *always*?)
  - correct control characters are sent (at least they enter the 8b/10b encoder), but register is not written
- ACK01, ACK03 are returned, ACK00, ACK02 not