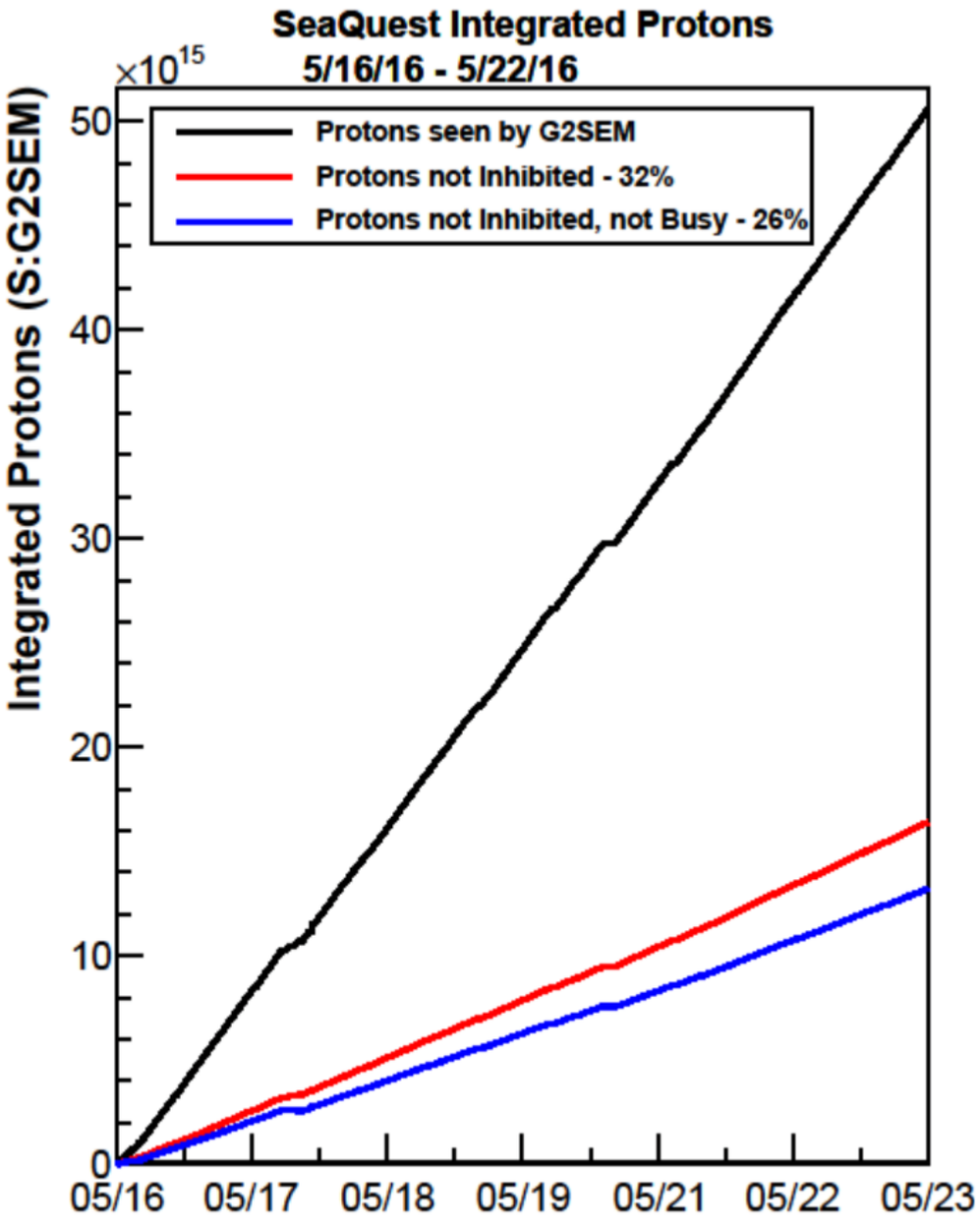
DAQ Upgrade: the Path Forward

Kun, Grass, Xin-Kun, Dave

E906/E1039 Joint Collaboration Meeting
Fermilab, 06/13/2106

What could we gain



Suppose we have unlimited DAQ bandwidth:

- Naively we could at least fill the gap between red and blue — 25% of POT
- Below the blue curve, we could recover the efficiency loss by trigger matrix selection:
  - For high mass DY evens in acceptance (trig. eff. ~55%):
    - 25% is lost due to T/B requirement
    - another 20% is lost due to hot road removal
  - For in-acceptance J/ψ (trig. eff. ~7%):
    - 45% is lost due to T/B requirement
    - another 48% is lost due to hot road removal
- In the far future, with improved tracker, we could recover part of the 68% protons lost by inhibiting high intensity buckets

2x DY and 10x J/ψ are within reach with the rest of spectrometer unchanged!

Los Alamos NATIONAL LABORATORY EST. 1943

Current DAQ workflow and deadtime

v1495 lv-1

Trigger Supervisor (TS)

TDCs

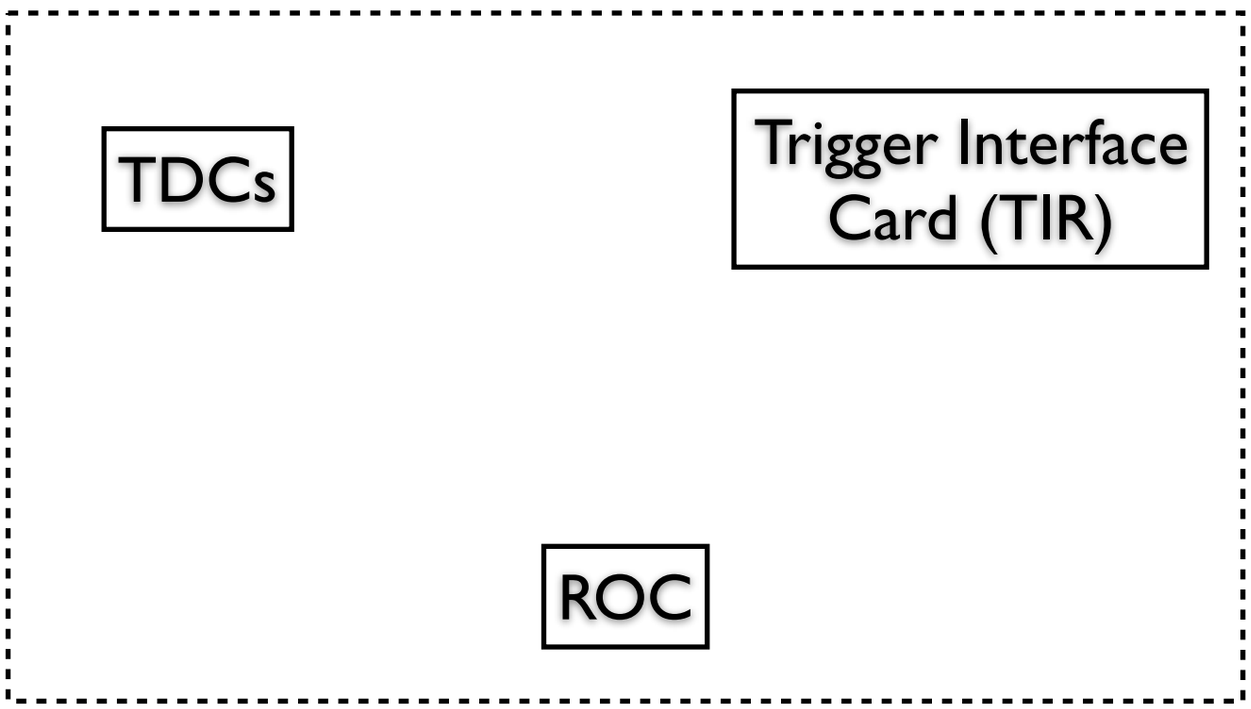
Trigger Interface Card (TIR)

ROC

CODA

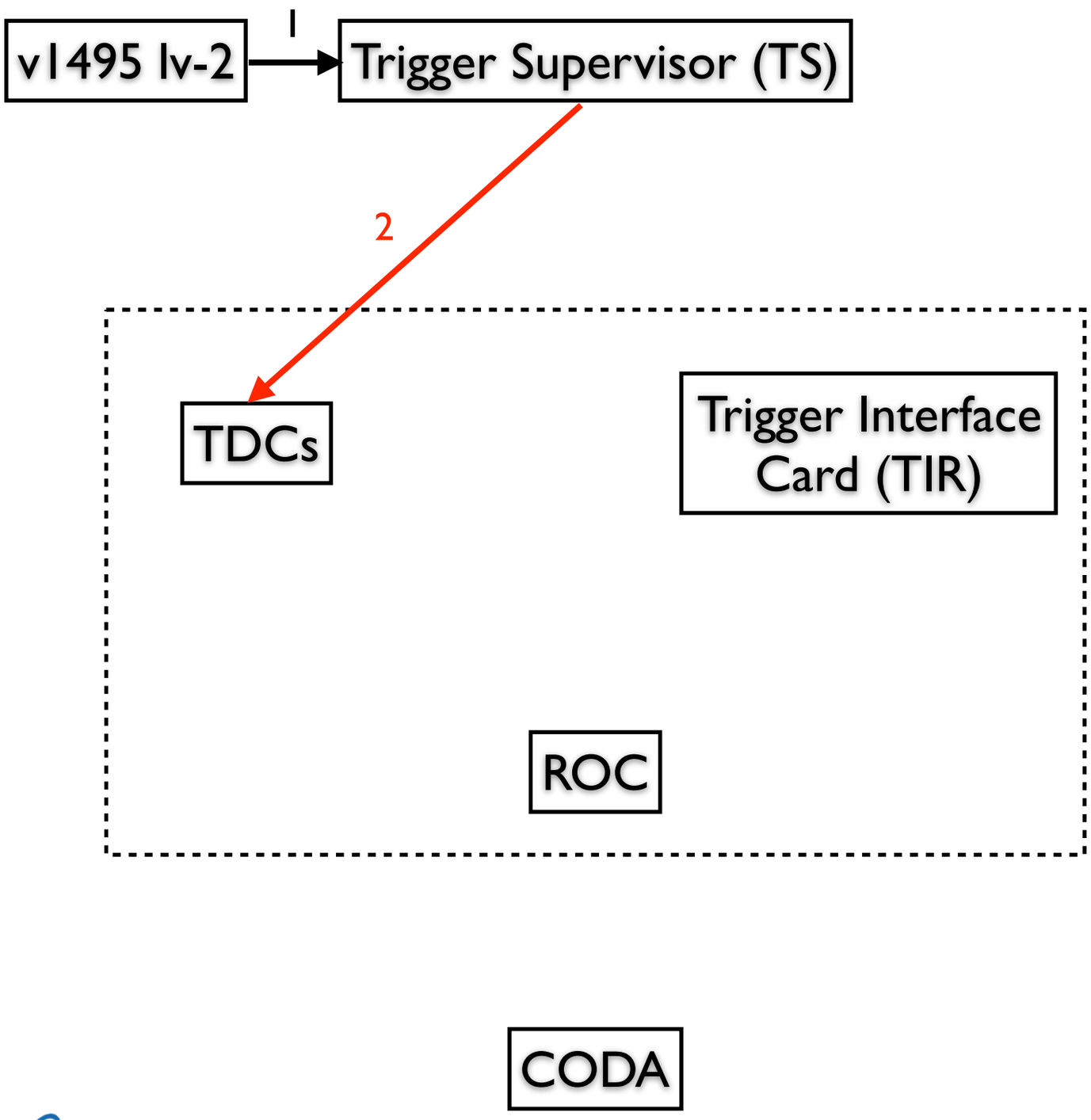
Current DAQ workflow and deadtime

(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy



Los Alamos NATIONAL LABORATORY EST. 1943

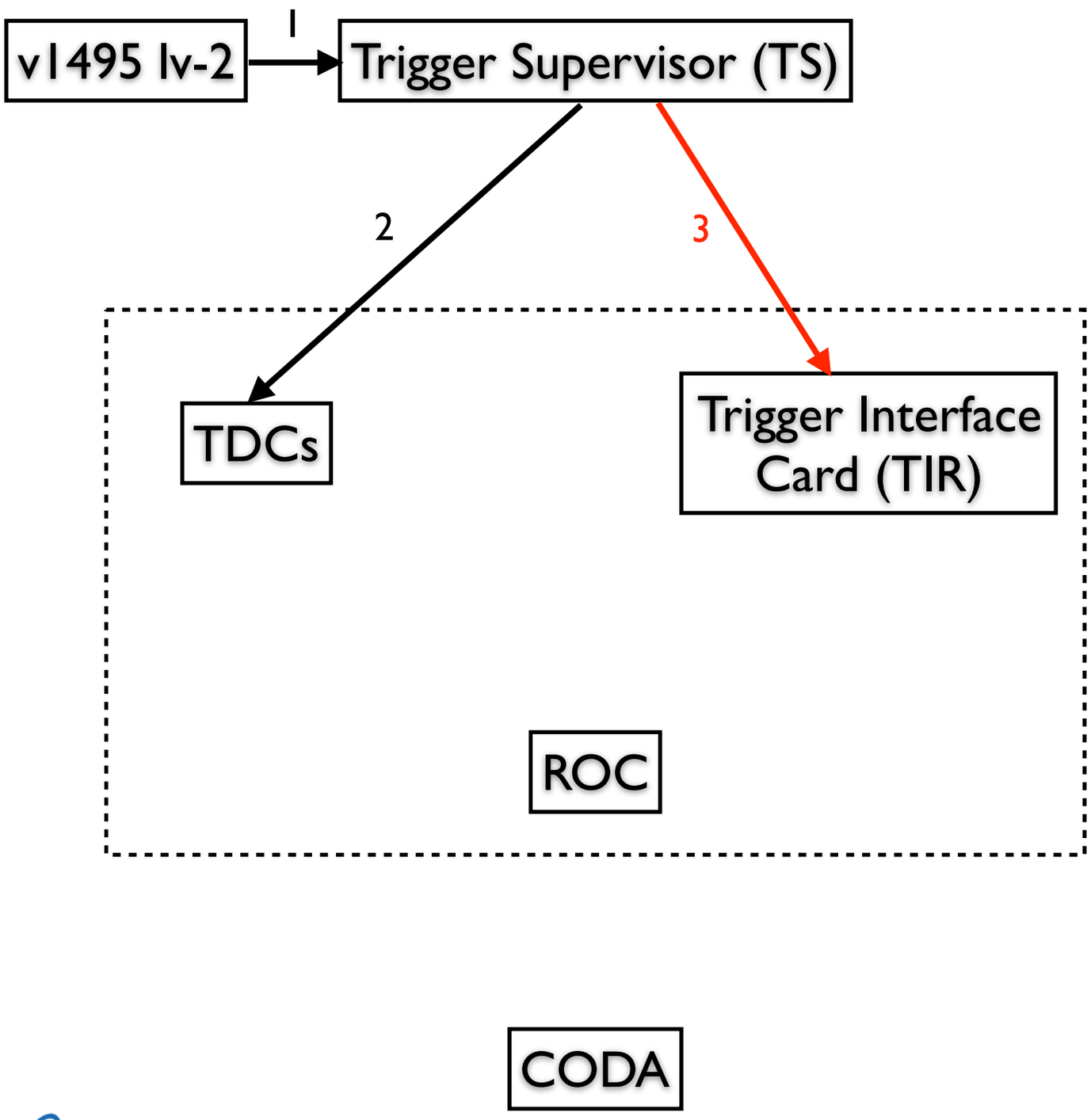
Current DAQ workflow and deadtime



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its ring buffer

Current DAQ workflow and deadtime

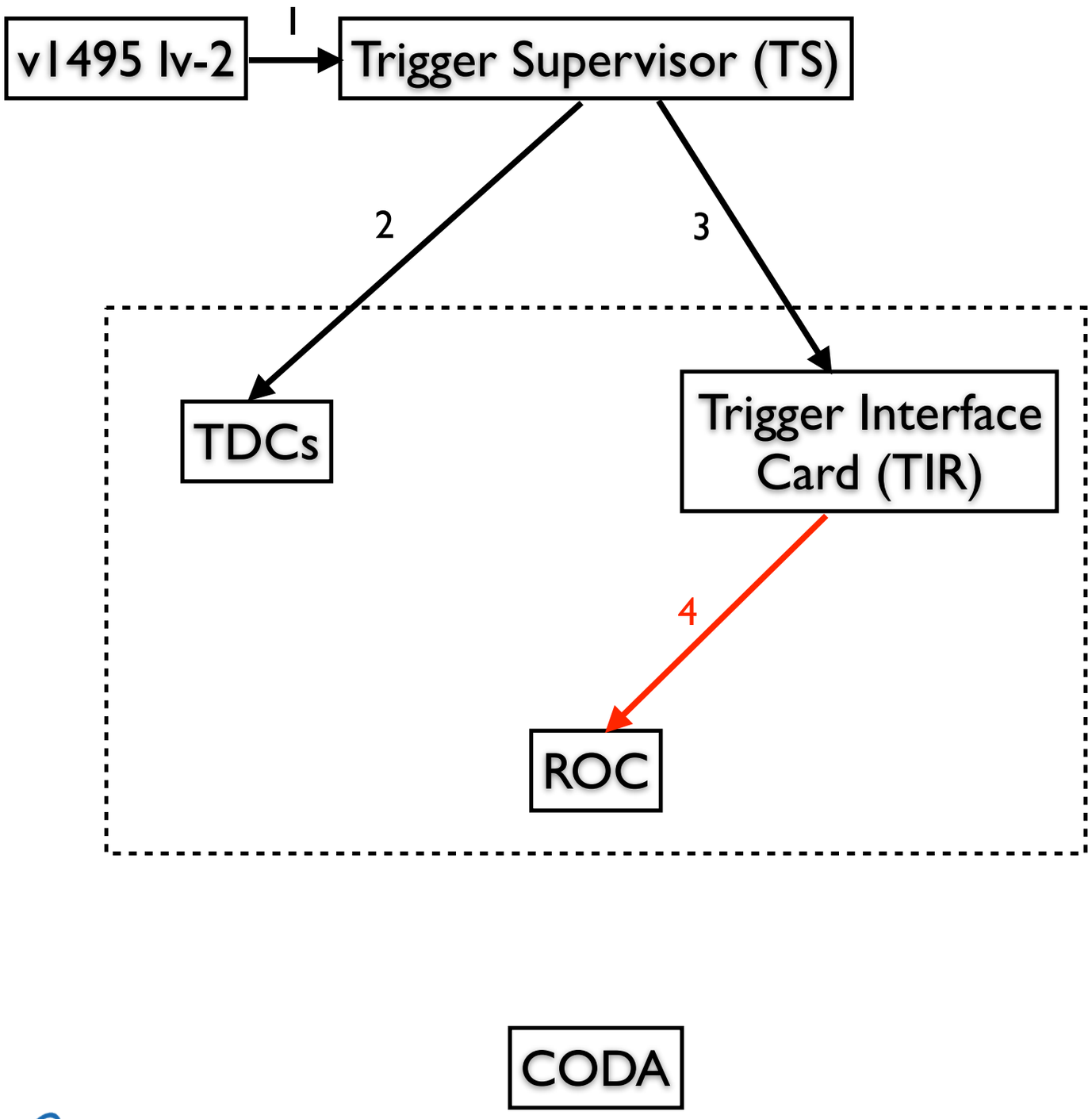


(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its ring buffer

(3) delayed by **32µs**, trigger is sent to all TIRs. This is the '*copy-in-progress*' time

Current DAQ workflow and deadtime



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

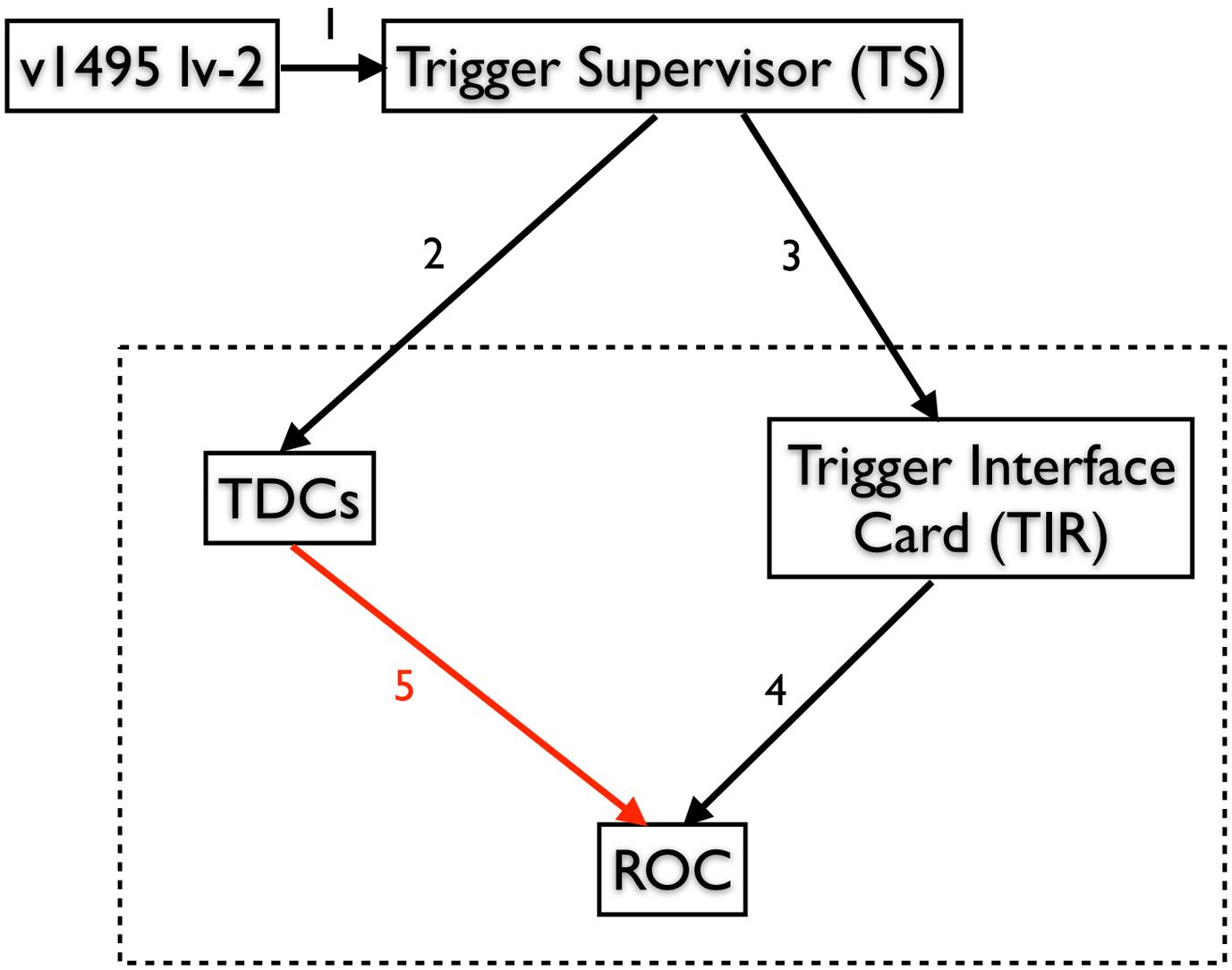
(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its ring buffer

(3) delayed by **32µs**, trigger is sent to all TIRs. This is the '*copy-in-progress*' time

(4) after another **10µs**, TIR instructs ROC to read out TDCs

Los Alamos NATIONAL LABORATORY EST. 1943

Current DAQ workflow and deadtime



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its ring buffer

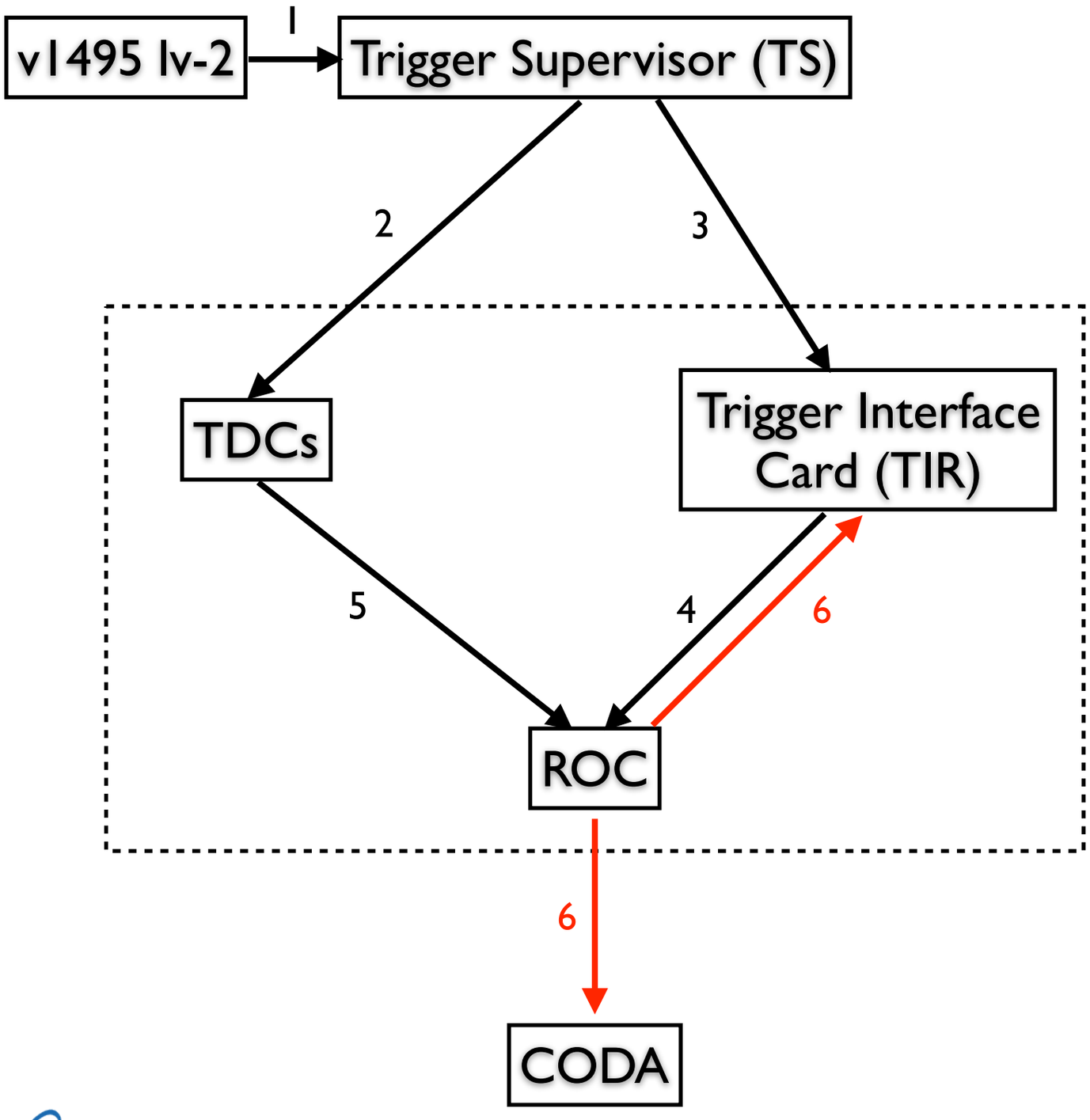
(3) delayed by **32µs**, trigger is sent to all TIRs. This is the '*copy-in-progress*' time

(4) after another **10µs**, TIR instructs ROC to read out TDCs

(5) TDCs send hits to ROC through VME backplane, this takes **~100µs**

Los Alamos NATIONAL LABORATORY EST. 1943

Current DAQ workflow and deadtime



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its ring buffer

(3) delayed by **32µs**, trigger is sent to all TIRs. This is the '*copy-in-progress*' time

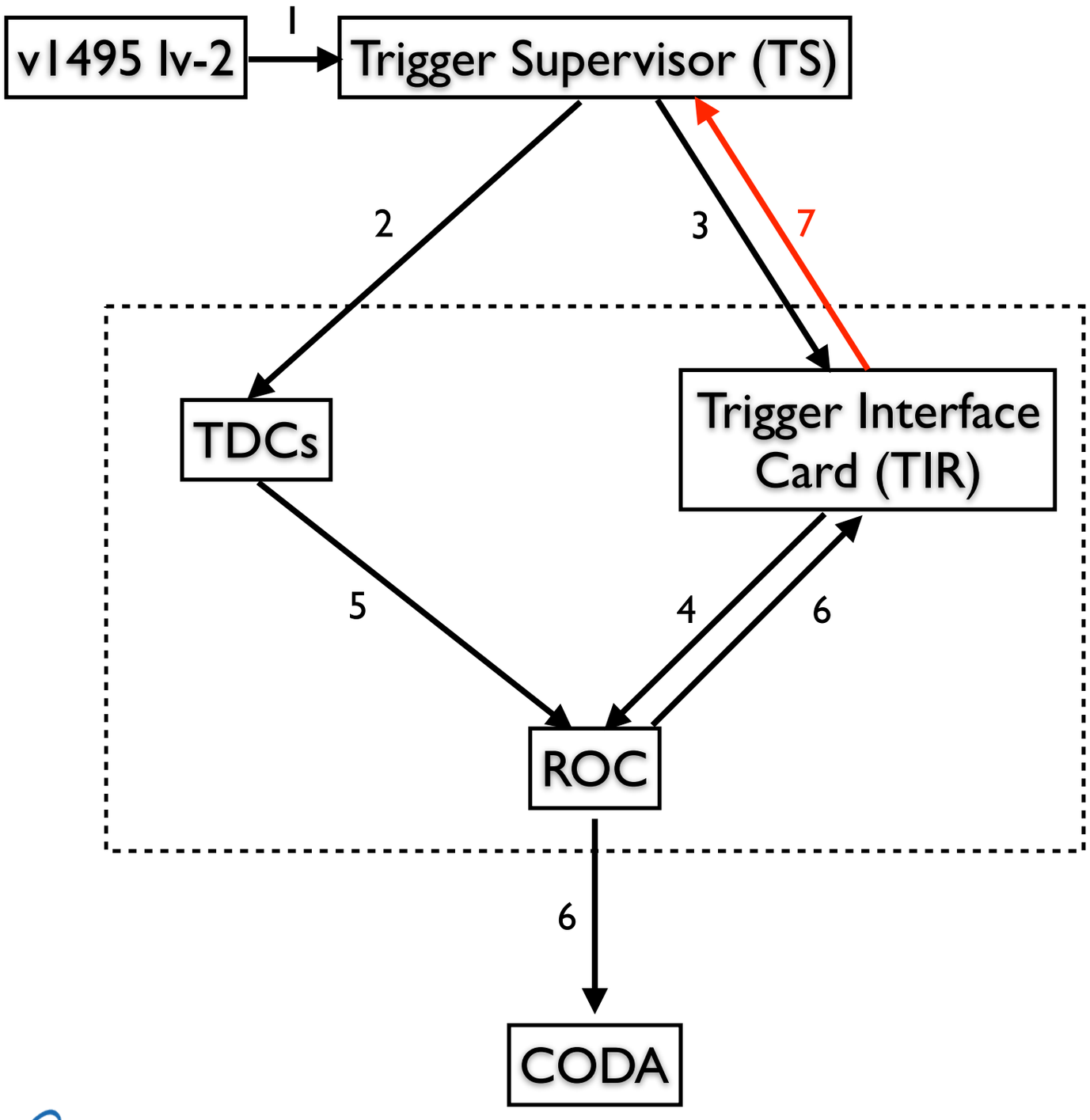
(4) after another **10µs**, TIR instructs ROC to read out TDCs

(5) TDCs send hits to ROC through VME backplane, this takes **~100µs**

(6) ROC tells TIR it's done reading, and sends data to upstairs through ethernet

Los Alamos NATIONAL LABORATORY EST. 1943

Current DAQ workflow and deadtime



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its ring buffer

(3) delayed by **32µs**, trigger is sent to all TIRs. This is the '*copy-in-progress*' time

(4) after another **10µs**, TIR instructs ROC to read out TDCs

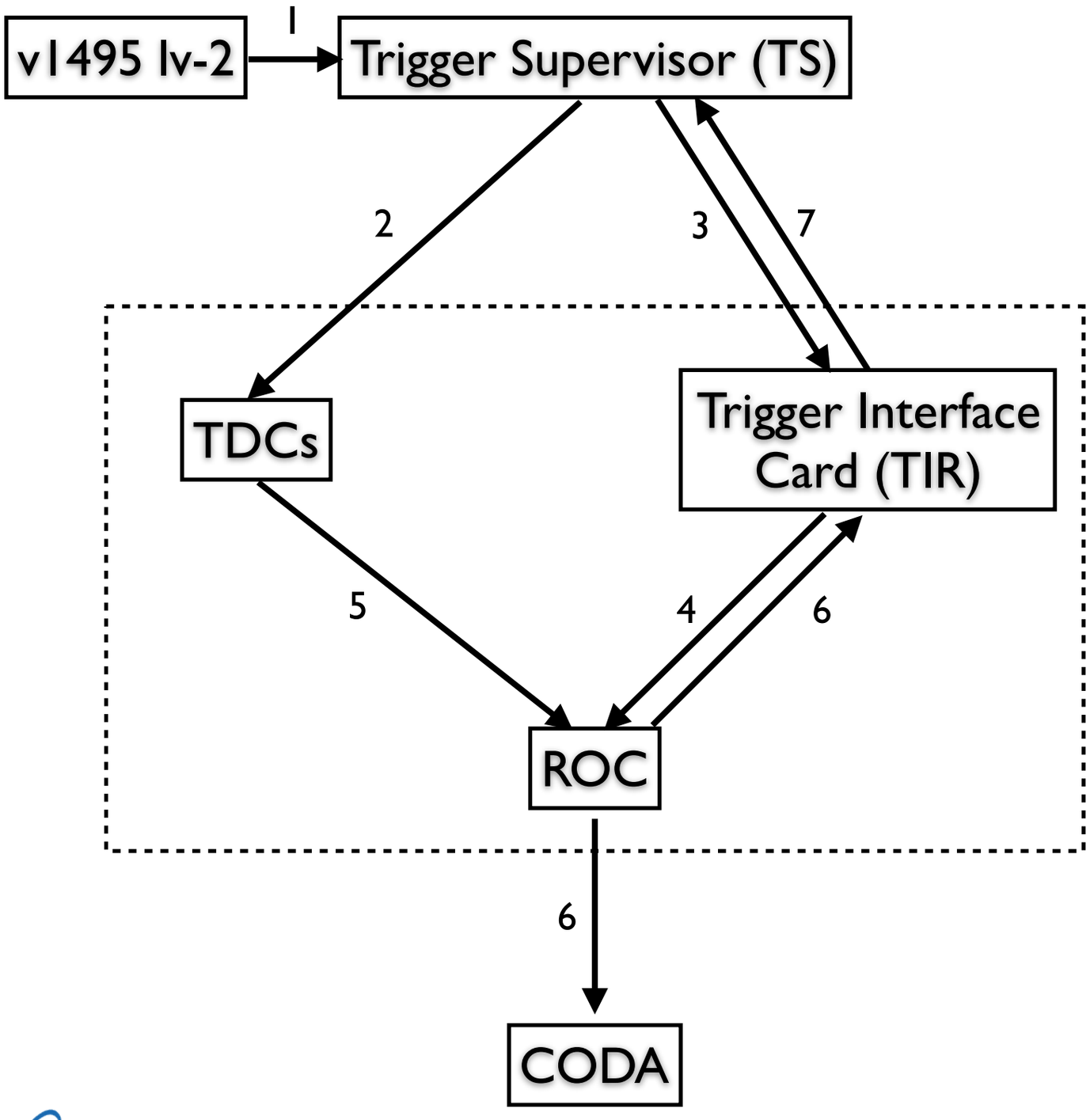
(5) TDCs send hits to ROC through VME backplane, this takes **~100µs**

(6) ROC tells TIR it's done reading, and sends data to upstairs through ethernet

(7) TIR issues ACK back to TS saying this VME is done. After receiving ACKs from all VMEs, TS is reset and ready to take next trigger

Los Alamos NATIONAL LABORATORY EST. 1943

Current DAQ workflow and deadtime



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its ring buffer

(3) delayed by **32µs**, trigger is sent to all TIRs. This is the '*copy-in-progress*' time

(4) after another **10µs**, TIR instructs ROC to read out TDCs

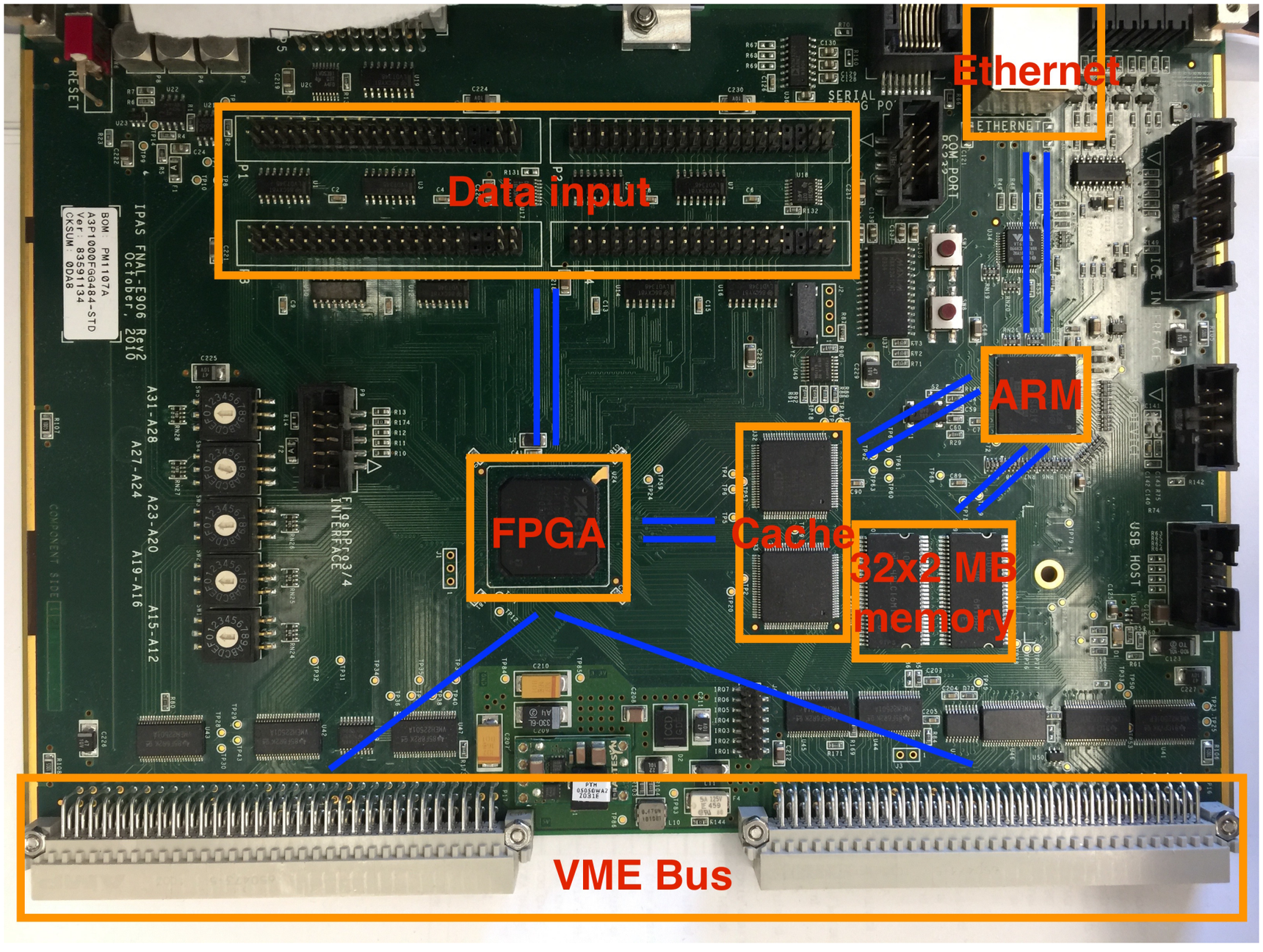
(5) TDCs send hits to ROC through VME backplane, this takes **~100µs**

(6) ROC tells TIR it's done reading, and sends data to upstairs through ethernet

(7) TIR issues ACK back to TS saying this VME is done. After receiving ACKs from all VMEs, TS is reset and ready to take next trigger

In total we have ~140µs deadtime per event

Proposed modification to TW-TDC microcode



Current setup:

- Input - FPGA - VME
- overall dead time ~140 µs
- Copy-in-progress time 42 µs

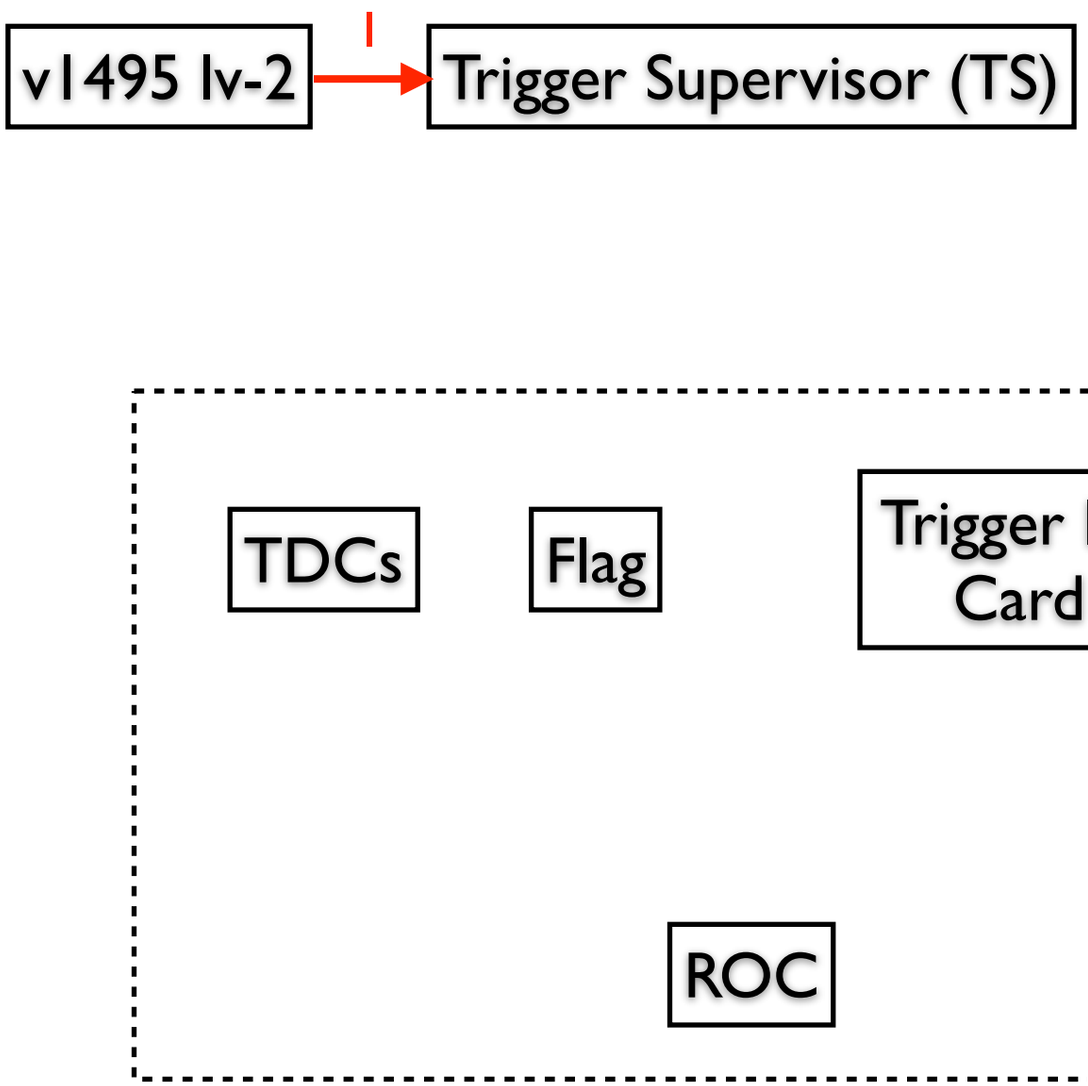
Step 1:

- during the 4.2s spill on time, FPGA send all data to memory through the ARM chip
- FPGA sets a register for ROC to know the readout is finished
- 2 options to maintain data alignment:
  - each TDC maintain an internal counter as eventID
  - ROC writes a centralized eventID to TDC (+2µs)

Step 2:

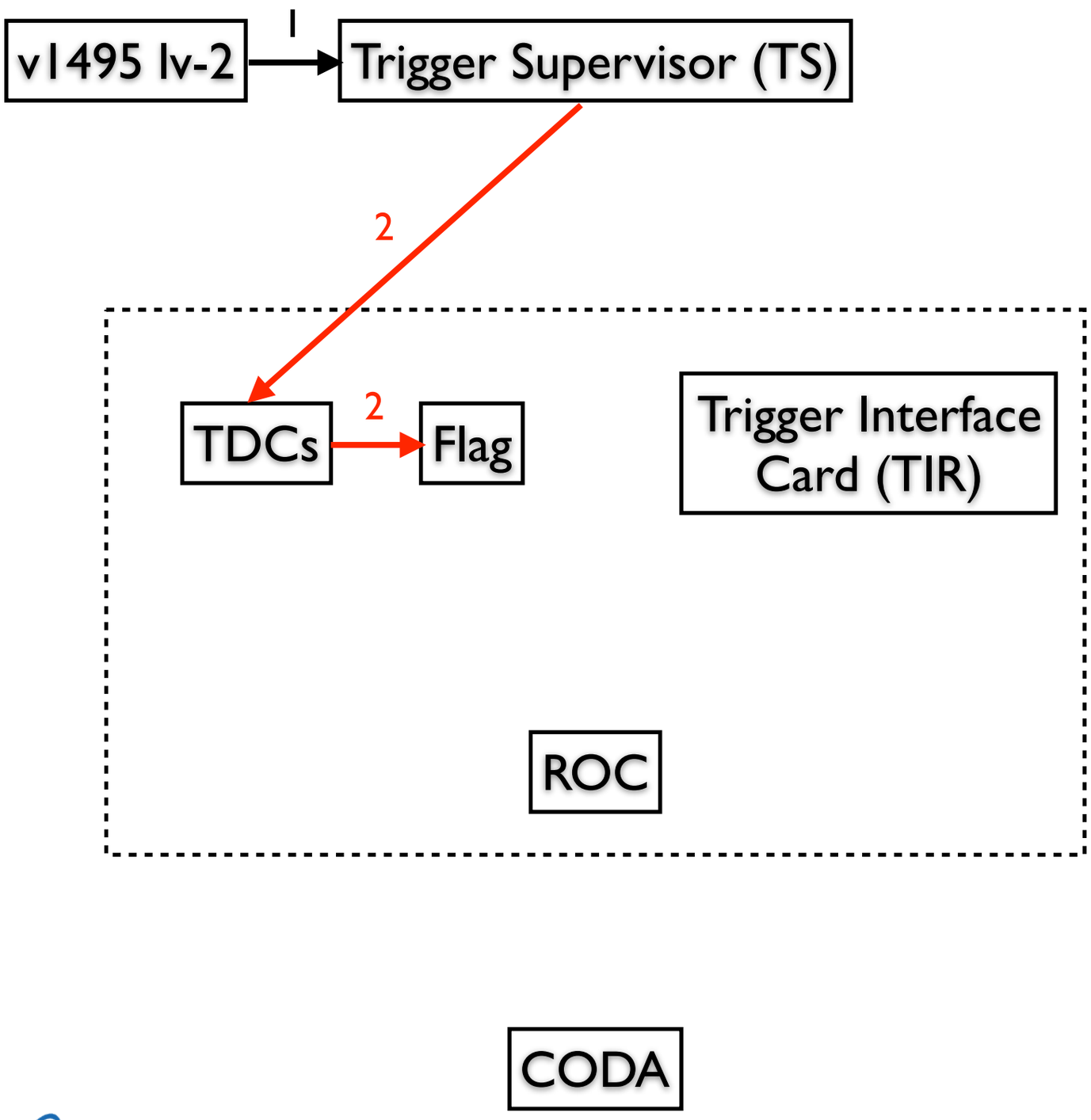
- upon receiving EOS, ARM starts to send data back to VME backplane through FPGA
- if VME bandwidth is still a bottleneck, we could program ARM to directly send data through ethernet port (more challenging)

Adjusted DAQ workflow



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

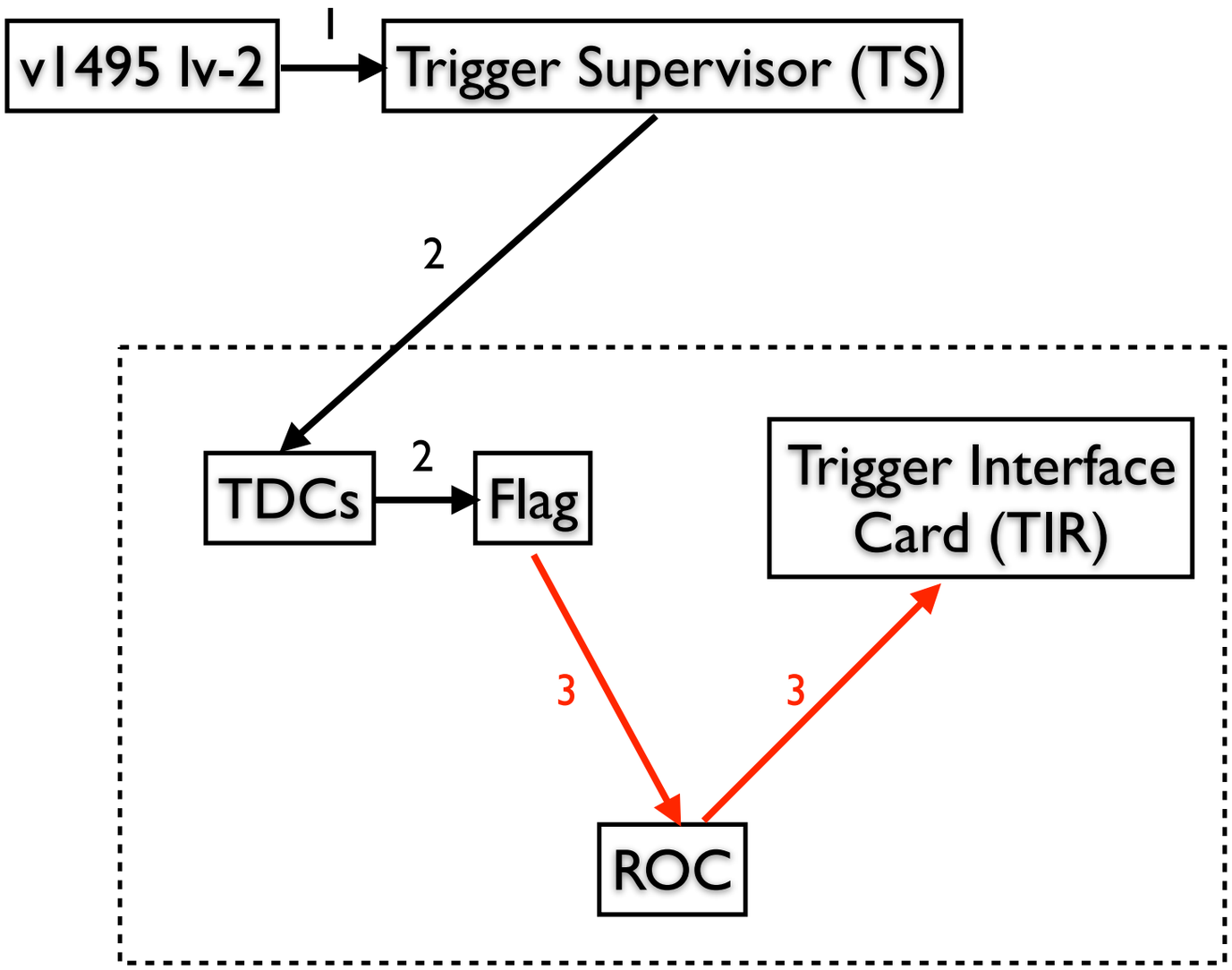
Adjusted DAQ workflow



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its onboard memory, then sets a register to indicate it's finished

Adjusted DAQ workflow



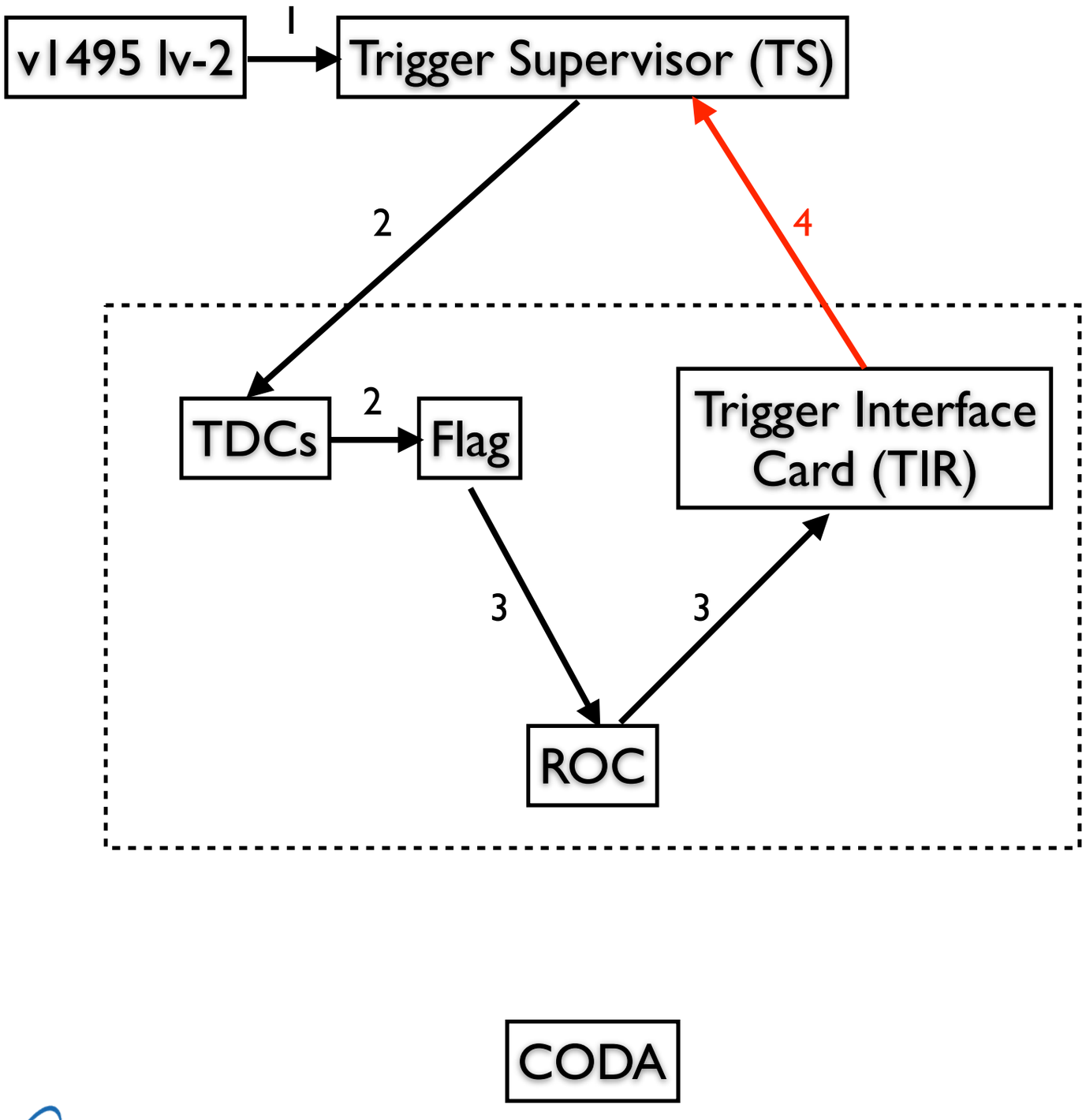
(1) Trigger supervisor receives trigger from v1495 Iv-2, TS is set to busy

(2) Iv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its onboard memory, then sets a register to indicate it's finished

(3) ROC keeps scanning all TDCs, and tells TIR all TDCs are done

Los Alamos NATIONAL LABORATORY EST. 1943

Adjusted DAQ workflow



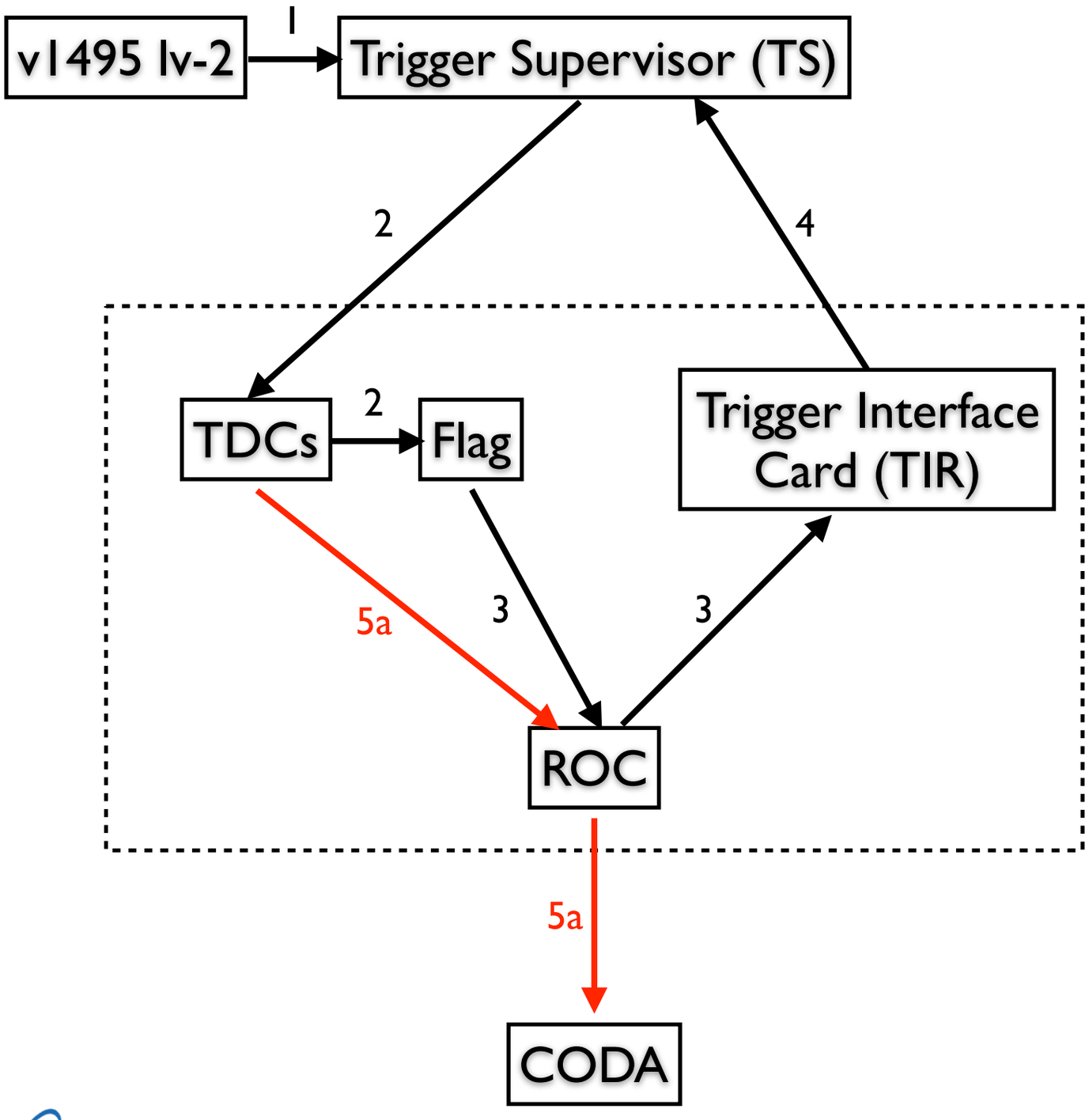
(1) Trigger supervisor receives trigger from v1495 Iv-2, TS is set to busy

(2) Iv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its onboard memory, then sets a register to indicate it's finished

(3) ROC keeps scanning all TDCs, and tells TIR all TDCs are done

(4) TIR issues ACK back to TS saying this VME is done. After receiving ACKs from all VMEs, TS is reset and ready to take next trigger

Adjusted DAQ workflow



(1) Trigger supervisor receives trigger from v1495 lv-2, TS is set to busy

(2) lv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its onboard memory, then sets a register to indicate it's finished

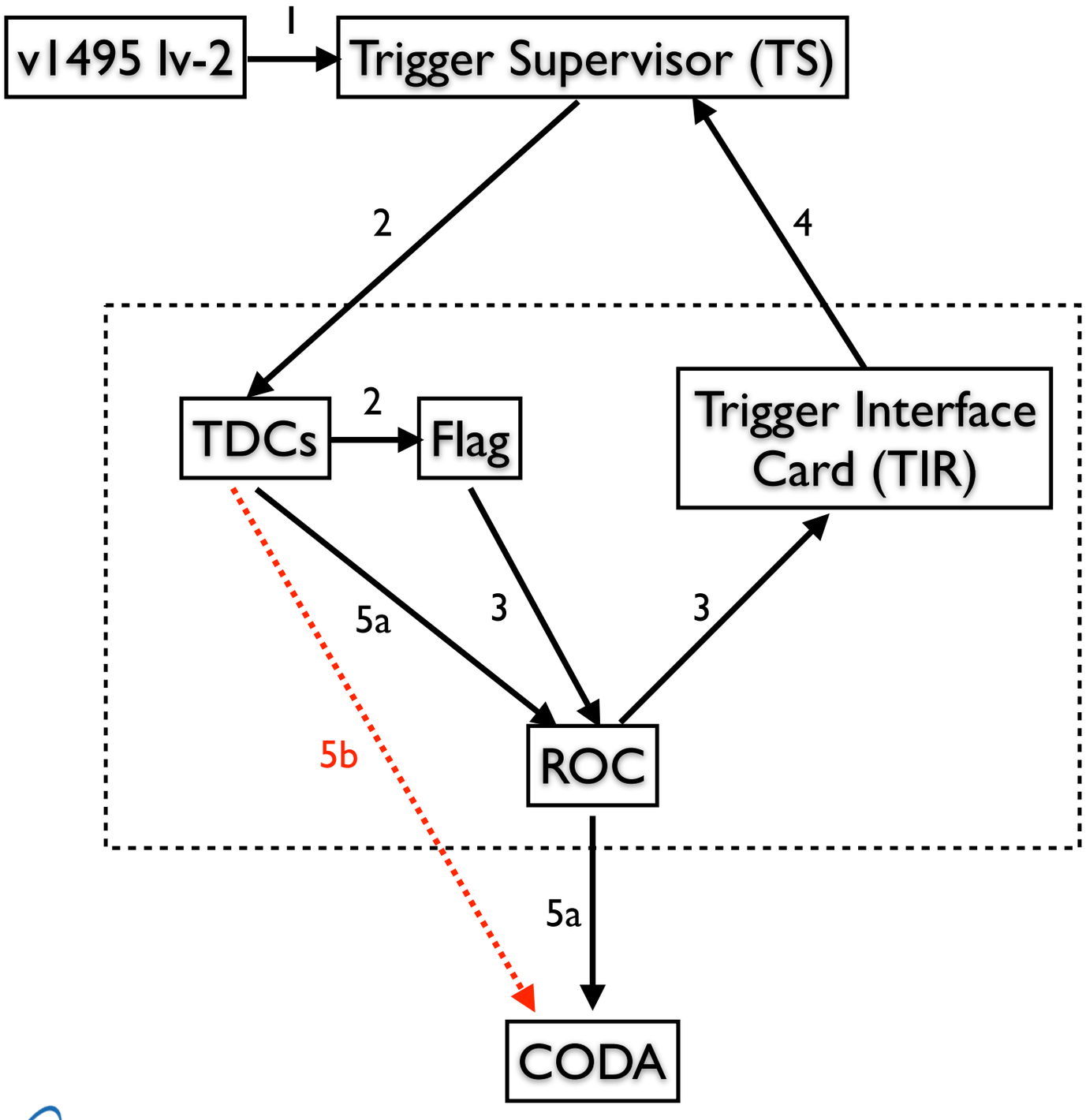
(3) ROC keeps scanning all TDCs, and tells TIR all TDCs are done

(4) TIR issues ACK back to TS saying this VME is done. After receiving ACKs from all VMEs, TS is reset and ready to take next trigger

(5a) Upon receiving EOS, ROC reads out all TDCs and send data to CODA (in a much less time-sensitive way)

Los Alamos NATIONAL LABORATORY EST. 1943

Adjusted DAQ workflow



(1) Trigger supervisor receives trigger from v1495 Iv-2, TS is set to busy

(2) Iv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its onboard memory, then sets a register to indicate it's finished

(3) ROC keeps scanning all TDCs, and tells TIR all TDCs are done

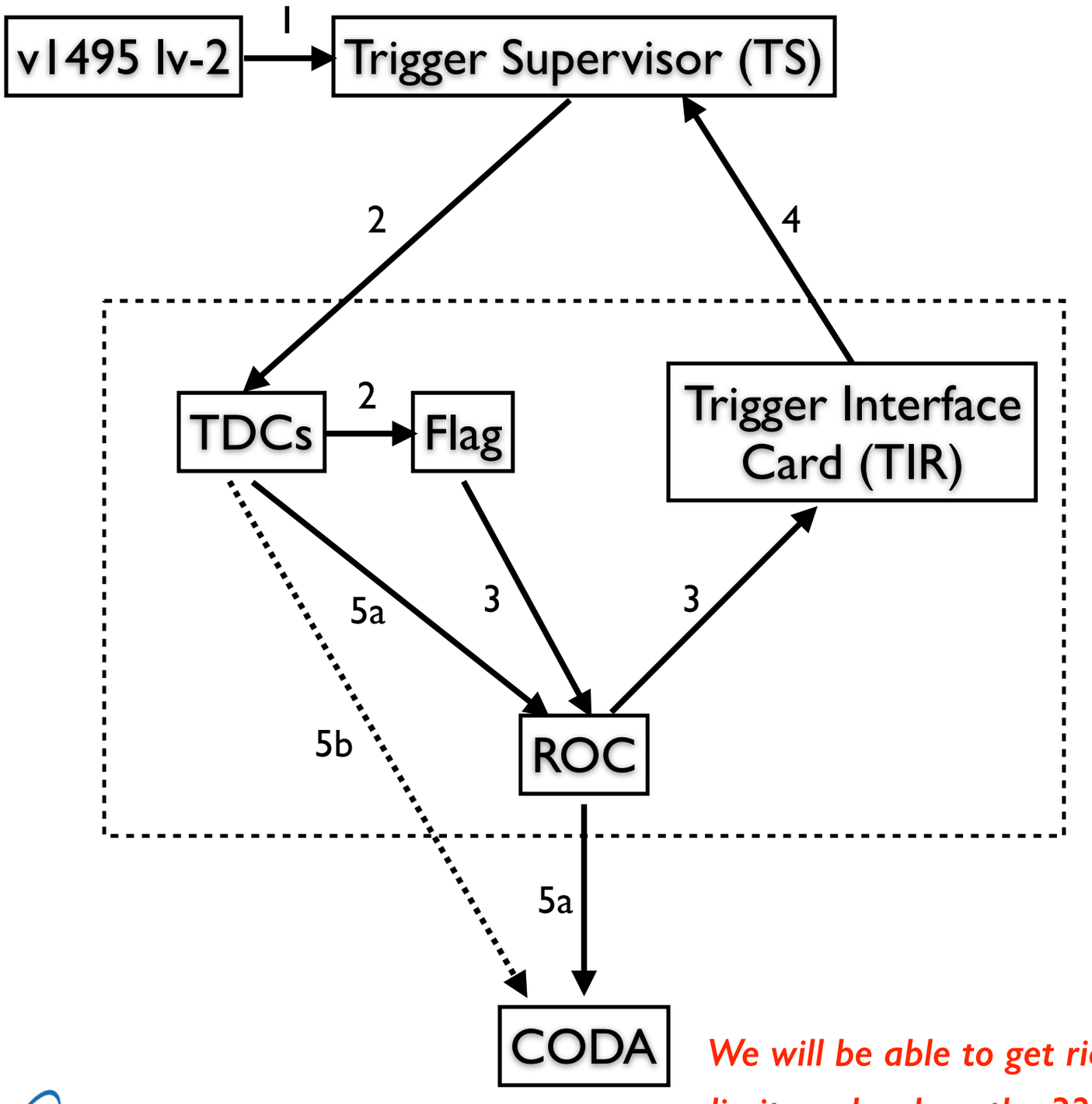
(4) TIR issues ACK back to TS saying this VME is done. After receiving ACKs from all VMEs, TS is reset and ready to take next trigger

(5a) Upon receiving EOS, ROC reads out all TDCs and send data to CODA (in a much less time-sensitive way)

(5b) Alternatively, each TDC could send data directly to CODA through onboard ethernet

Los Alamos NATIONAL LABORATORY EST. 1943

Adjusted DAQ workflow



(1) Trigger supervisor receives trigger from v1495 Iv-2, TS is set to busy

(2) Iv-1 accept is fanned out to all TDCs. TDCs stops taking data and save all hits in its onboard memory, then sets a register to indicate it's finished

(3) ROC keeps scanning all TDCs, and tells TIR all TDCs are done

(4) TIR issues ACK back to TS saying this VME is done. After receiving ACKs from all VMEs, TS is reset and ready to take next trigger

(5a) Upon receiving EOS, ROC reads out all TDCs and send data to CODA (in a much less time-sensitive way)

(5b) Alternatively, each TDC could send data directly to CODA through onboard ethernet

We will be able to get rid of the 100+10μs deadtime from VME bandwidth limit, and reduce the 32μs 'copy-in-progress' deadtime as much as possible

A (almost) guaranteed factor of 5 improvement

Los Alamos NATIONAL LABORATORY EST. 1943

Tasks and timeline

Overall objective: *have a buffered DAQ working before next run*

1. Proof of principle: program the ARM processor to read/write data between FPGA and memory (Terry K. already succeeded the first step)
2. Development of ARM code and modification to existing FPGA code
3. Have a working TDC board on test bench by September, and start CODA integration in the hall
4. Have a working system by November!

Task force: Xin-Kun, Grass, Kun, Dave, Jin-Yuan, Terry, *and more?*

Los Alamos NATIONAL LABORATORY EST. 1943

Backup slides

Data acquisition phase.

- Trigger distribution:
  - Trigger still fanned out to each TDC (LEMO input #1? What is logic level? Schematic implies NIM.)
    - TDC microcode need to be changed
    - TDC computer code needs to be written
    - Events buffered to memory on TDC board
  - Want to time stamp each event to ensure data alignment… send trigger to ROC also, but ROC shouldn't read out data; only send trigger # & time stamp to every TDC on VME back plane; TDC should store trigger # and time stamp with data record.
    - ROC code needs to be modified.
    - Needs to operate in less time than fixed dead time.
- Dead time signal.
  - Fixed dead time determined by "copy in progress" time?
  - Created & output by 1TDC/crate (TTL output 3.3V TTL?).
    - Output driver spec says output voltage = Vcc. Schematic says Vcc=3V3.

Readout Phase

- Readout started by End-of-Spill.
  - Signal from BIM (QIE board)?
  - Signal from computer in control room?
  - Sent to every TDC in each VME crate (LEMO input#2)?
- Requires new TDC code & microcode.
  - Last TDC in VME readout chain sends trigger to ROC (LEMO TTL Output).
    - When EOS is received.
    - After every event is read out except for last buffer event.
  - ROC sends buffer of 100? (what number?) triggers to event builder.
  - How do we tell that read out of spill is complete?
    - Maybe each TDC should count triggers & output enough triggers to exactly fill NNN ROC buffers; the extra "events" could contain nothing except a total trigger count of triggers in the spill & perhaps a total word count