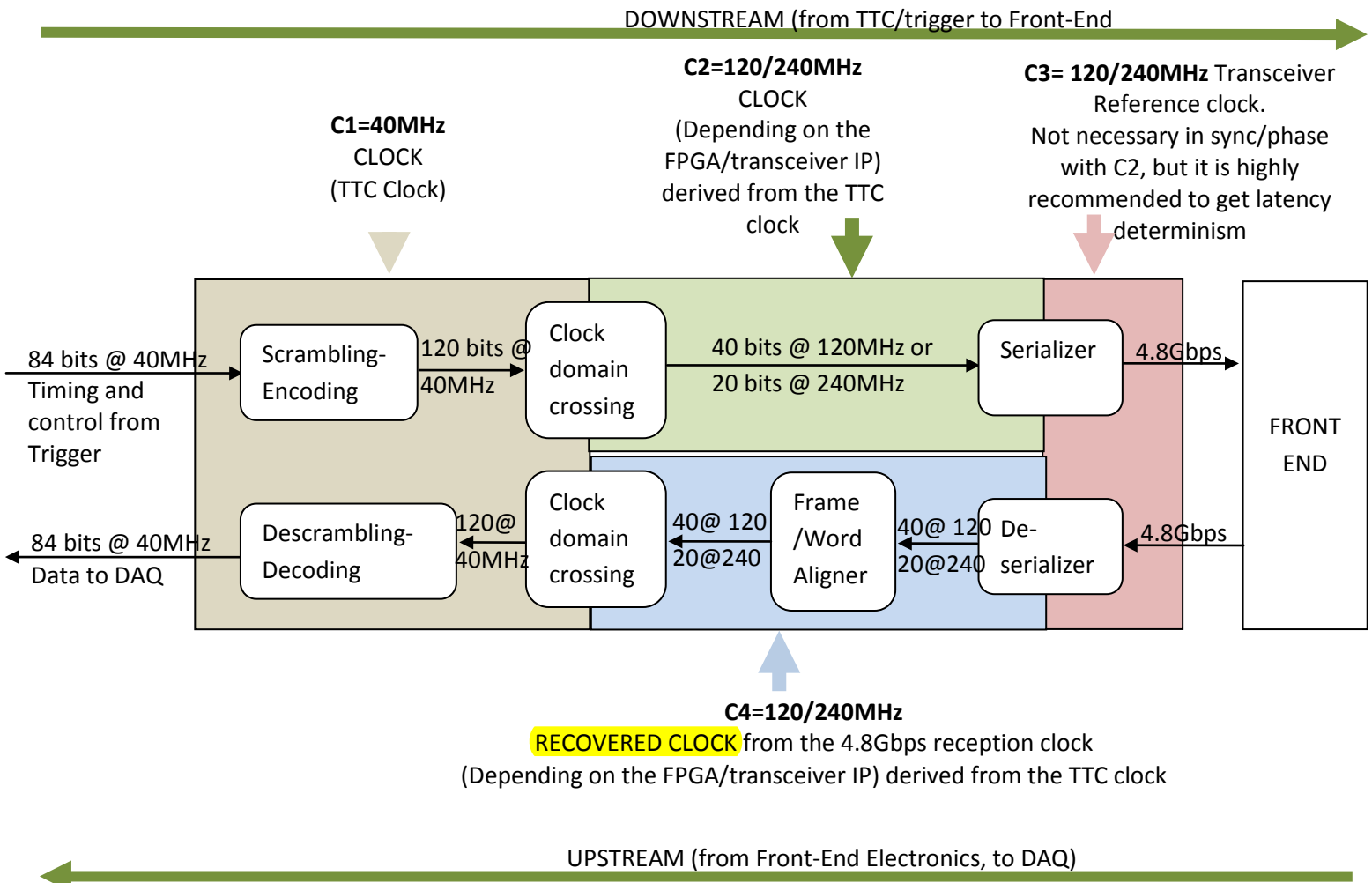


GBT-FPGA

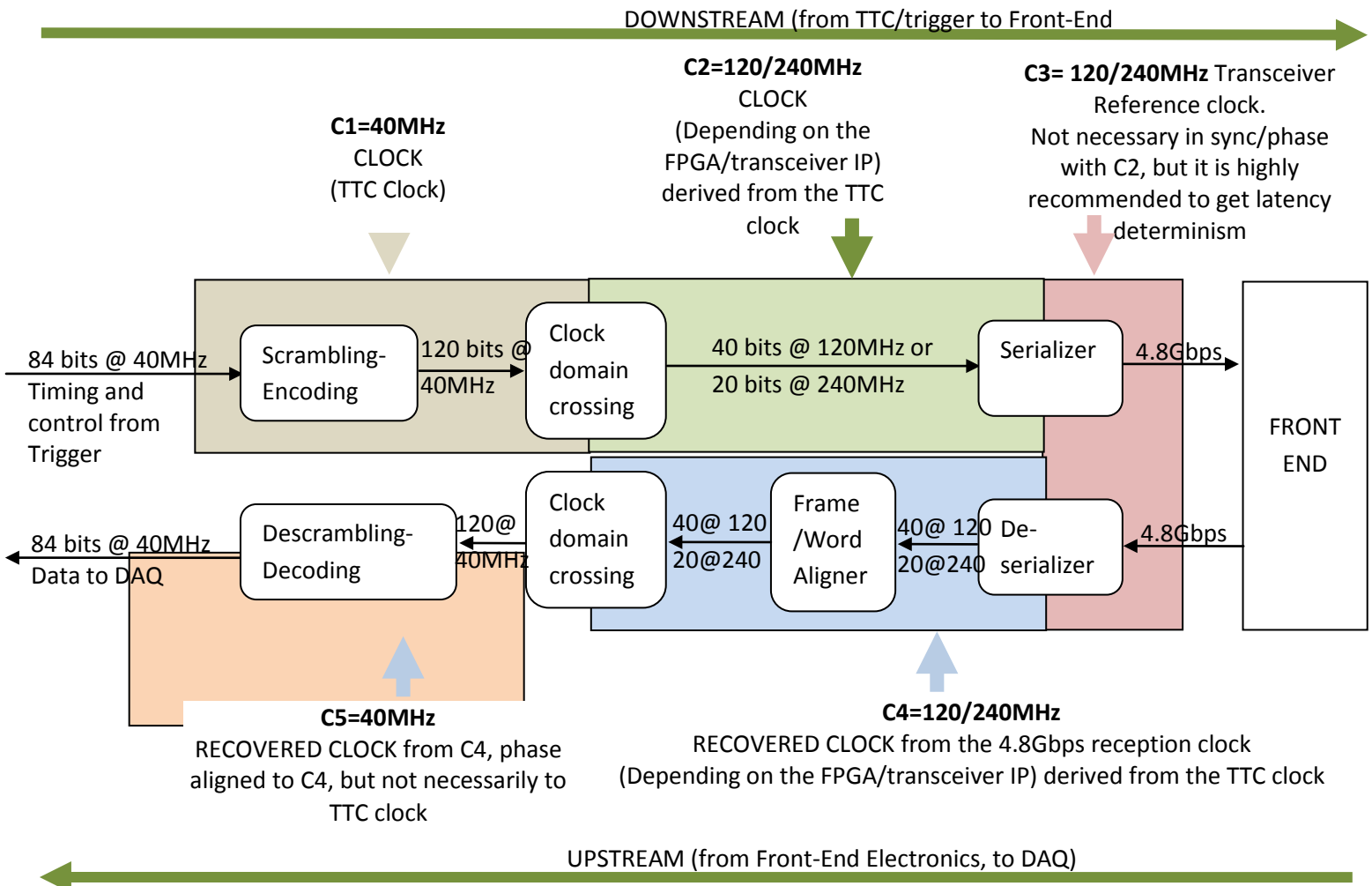
Comments on deterministic latency and recommendations to handle optimization schemes

General remarks:

- The GBT-FPGA source code provided here is focused on the implementation of the GBT protocol in the **BACK-END area**. Therefore, **no 40MHz clock phase recovery solution is proposed in the Deserializer block**.
- Two different clock schemes are possible/required for the GBT-FPGA serdes IP in the framework of the back-end systems, depending on the type of clock domain crossing blocks, and on the type of device used:
 - **Scheme 1** : TTC clock used for both input and output busses of the IP (downstream and upstream links)



- **Scheme 2:** different 40MHz clocks for input and output busses of the IP (downstream and upstream links)



Typical scheme:

- Provided with the starter kit
- Type of domain-crossing block: dp-RAM based.
- Preferred clock scheme: scheme 1. Scheme 2 is also possible.
- Consequence on latency:
 - Downstream latency is deterministic, but depends on:
 - Fiber length
 - Phase between C2 and C3 (which justifies that C3 is phase-aligned to C2). Having a variable phase between C2 and C3 could create a difference of 1 clock cycle in the downstream transmission latency.
 - Upstream latency is deterministic, but depends on:
 - Fiber length
 - Phase between C4 and C3 (which justifies that C4 is phase-aligned to C3). Having C4 and C3 not phase aligned could create a difference of 1 clock cycle in the upstream transmission latency.
 - Phase between C5/C1 and C4, depending on the scheme. Having a variable phase between C5/C1 and C4 could create a difference of 1 clock cycle in the upstream transmission latency.
 - Latency is high – typical loopback latency = 23 TTC clock period
- Recommendations:
 - Extract C2 and C3 from C1 with the same device, with a deterministic and controlled phase (typically, CDCE62005 clock synthesizer from TI)
 - Pay attention to the dp-RAM specification of your devices. Some (StratixII and IV) are less sensitive to phase between rdclock and wrclock in term of latency than some others (Virtex 5 and 6). However, their internal delays are higher.
 - Some internal features of the transceivers IPs can fix the phase between recovered C4 and ref clock C3.

Resources Optimization architecture:

- 4 versions are proposed to optimize the resources used by the decoder for a high number of links.
- Typical loopback latency is between 22 and 25 clock cycles (including the transceiver and a loopback cable of about 40cm).
- All are based on the idea of sharing one decoder between several links by multiplexing the data and multiplying the speed of the decoder.
 - Optimization version 0: No sharing, each link has its own decoder. Latency identical for all the links, 23 clock cycles.
 - Optimization version 2: 1 decoder for 2 links
 - Optimization version 3: 1 decoder for 3 links
 - Optimization version 4: 1 decoder for 4 links
- Various links implemented in the same 'resources optimization design' can have various loopback latencies. For example, the optimization by 4 can give links with loopback latencies of 24, 23 or 22 TTC clock cycles. Moreover, as previously said, the latency can vary by one clock cycle after a reset.

Latency optimization architecture:

- 3 versions are available to optimize the latency:
 - Version1: with dpRAM for the clock-domain-crossing, but reduced clocking steps in the counters implementation: typically, 8-9 clock cycles.
 - Version2: with registers instead of dpRAM for the clock domain crossing: typically 6-7 clock cycles.
 - Version3: same as version2, but with less synchronization steps: typically 5-6 clock cycles.
- These 3 versions are proposed as examples only, and have to be used with extra care.
- Implementing the latency optimization with clock-domain-crossing blocks based on registers requires using the scheme 2 to have a constant phase between 120/240 MHz and 40MHz clocks. This means not having the trigger/TTC control and the DAQ running with the same clock sources (even though they both are 40MHz, and are, ultimately, derived from the same TTC source, they can have changing phase difference).
- Consequently, having N links in a system implementing this 'latency optimization' solution using registers will result in N data busses with not deterministic phases between them.

Combining the 2 optimization architectures:

- As the latency optimized versions are based on the idea of reducing the number of synchronization steps, combining the 2 optimization schemes will result in having timing problems during compilation, and ultimately links which never lock. It is thus not recommended.