OBELIX V1
Digital circuit design

Xu Danwei – CPPM
Maximilian Babeluk -- HEPHY

Content

- Obelix specifications
- Digital Blocks & description
- Data_path & Clk_path
- Conclusions

TJ-MONOPIX2: Digital Specifications

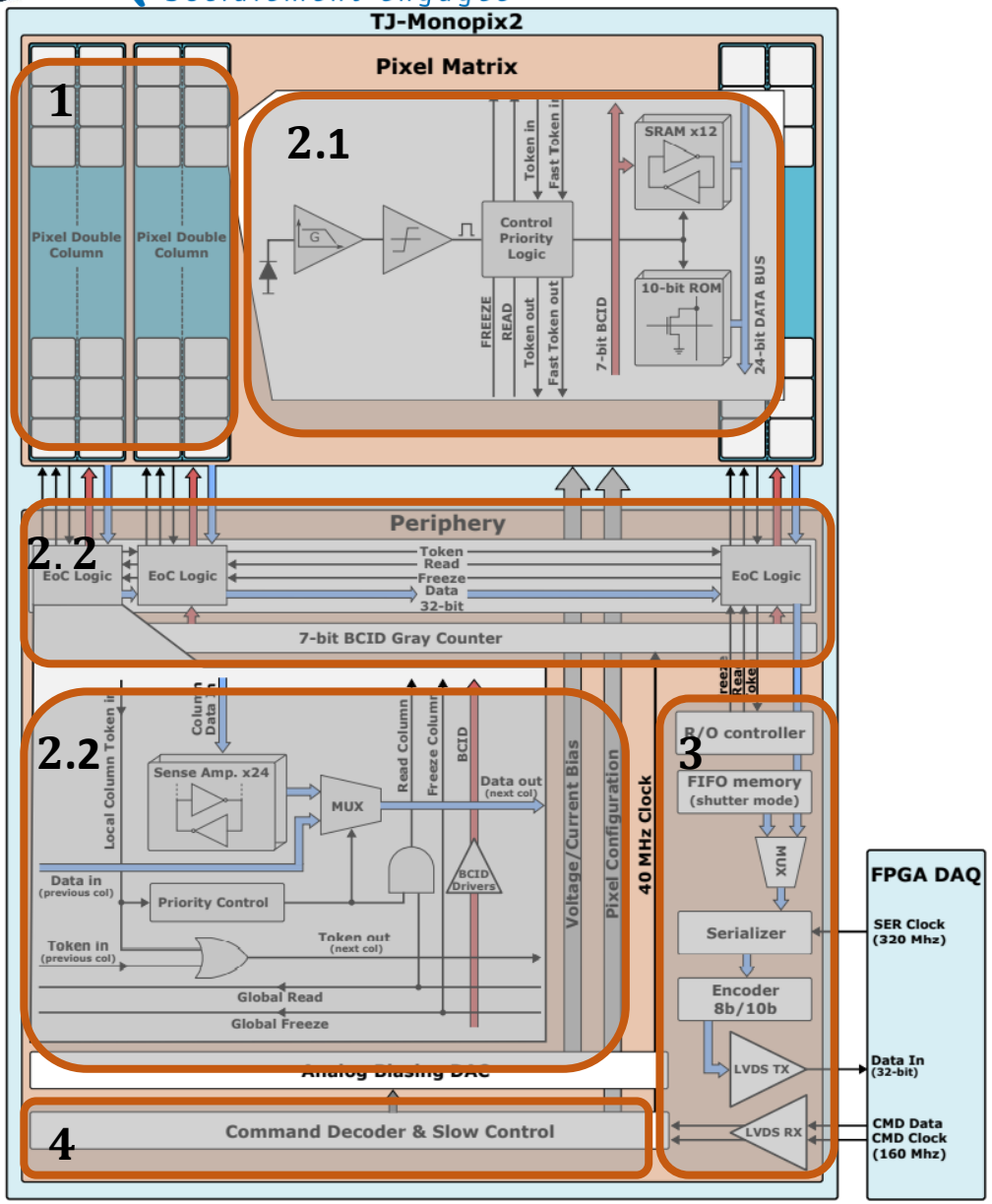


Figure 1: Architecture of TJ-MONOPIX2

- Several parts
- 1. Pixel matrix – double column
- 2. Eocs[1] readout logic –
- 2.1 custom block and data-bus readout circuit
- 2.2 digital logic, contains a token based priority mechanism
- 2 modes : direct readout / shutter mode
- 3. data transmission – transmit output data off-chip
- 4. command decoder and clk control

- Main clkin : 160MHz
- General function clk : 40MHz;
- Single output at 320Mb/s;
- RD53A LHC phase-II upgrade prototype ROIC ;

1: Eoc – End of Column

OBELIX_V1 : Digital Specifications of Belle II

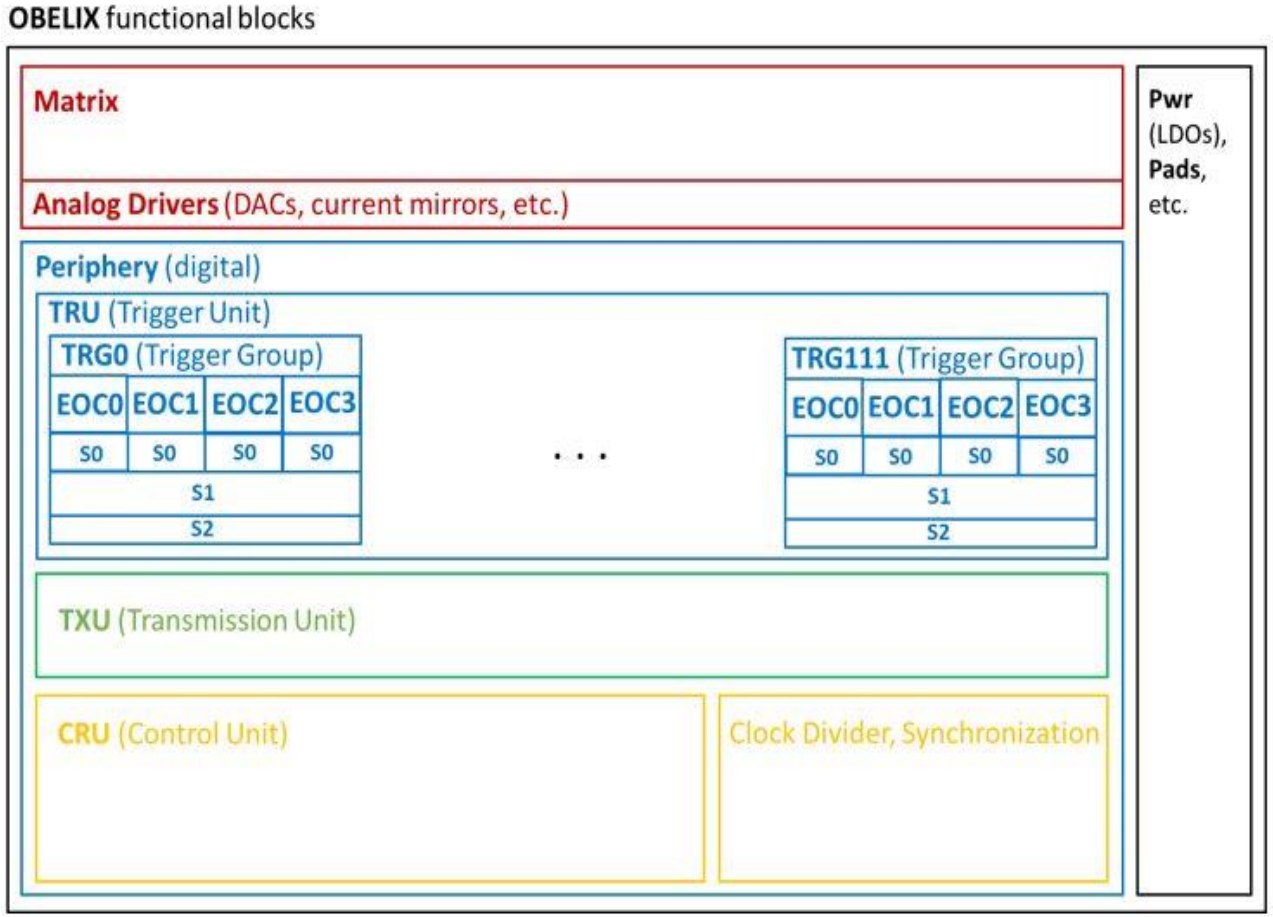


Figure 2 : Functional block of Obelix

- Main specifications of digital Top
- Main clkin : **160MHz**
- General function clk : from 40MHz to **20 MHz**;
- Single output at **320Mb/s**;
- Area limitation and **power consumption** ;
- **New End-of-column** adapted to Belle II trigger
- - Timestamped hits stored in memories
- - Read-out when timestamps matched with trigger
- - hit rate ≤ 120MHz/cm2
- **RD53B** control protocol from ATLAS CMS;
- TJ 180nm technology

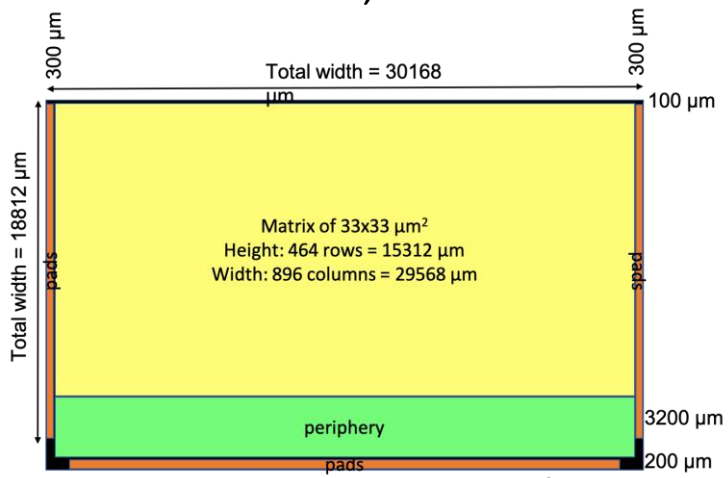


Figure 3 : Geometry and overall layout of Obelix

RD53B protocol:

https://cds.cehttps://cds.cern.ch/record/2665301rn.ch/record/2665301

OBELIX_V1 : Digital Blocks

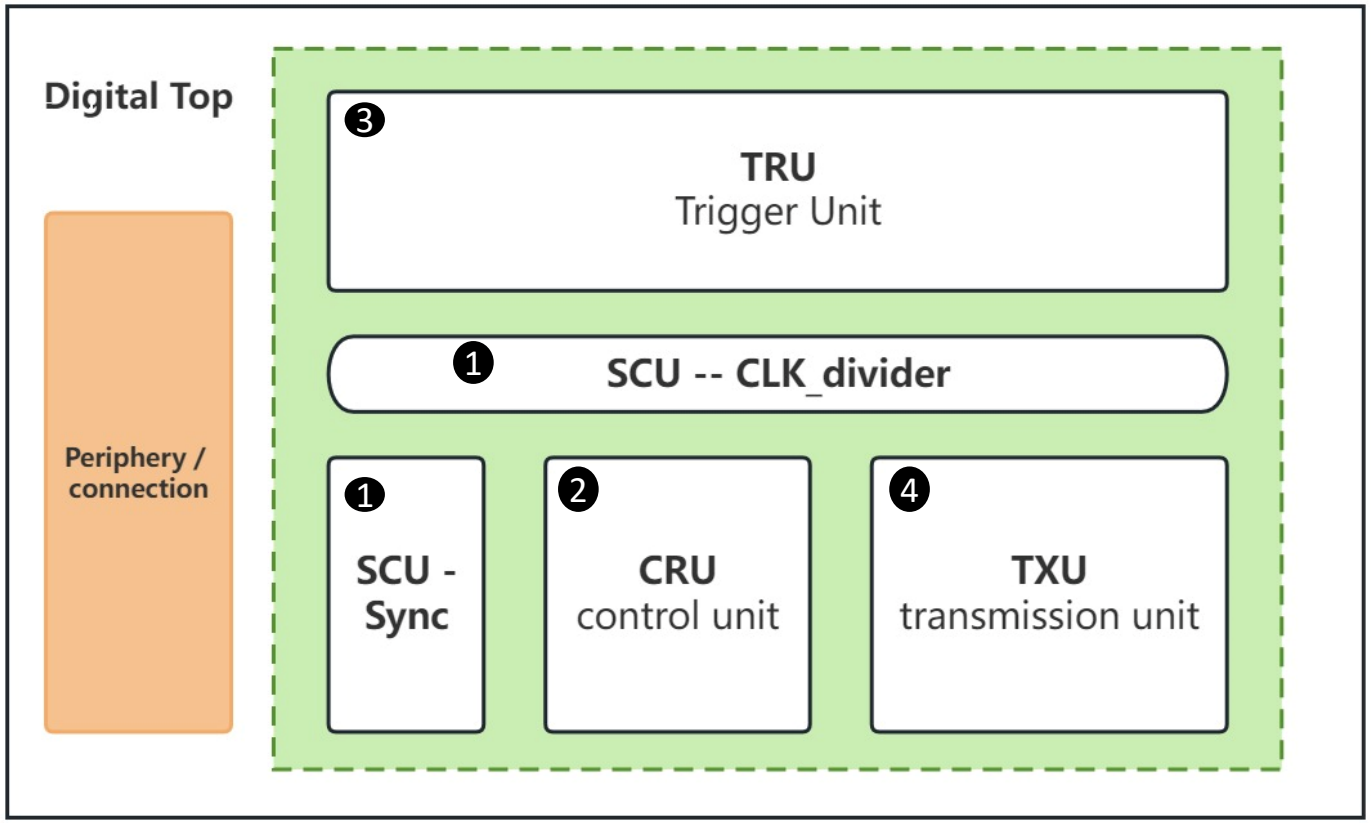


Figure 4 : Module division of Obelix digital top

- Module division : 4 main parts
- ❶**SCU – sync & clk divider**: digital clk divider, synchronize circuit & clk divider, RxDat format StoP conversion, main function: clock divider, Rx_data SIPO synchronization
- ❷**CRU – Control Unit**: Implementation RD53B interface, main functions: command decoder, global register configuration
- ❸**TRU – Trigger Unit**:Manage pixel data from the matrix-EOC and wait for the trigger to pick them for output
- ❹ **TXU – TX Unit**: generate output data and sequential output, main functions: data framing, serializer

SCU -- CLK Generator & Sync

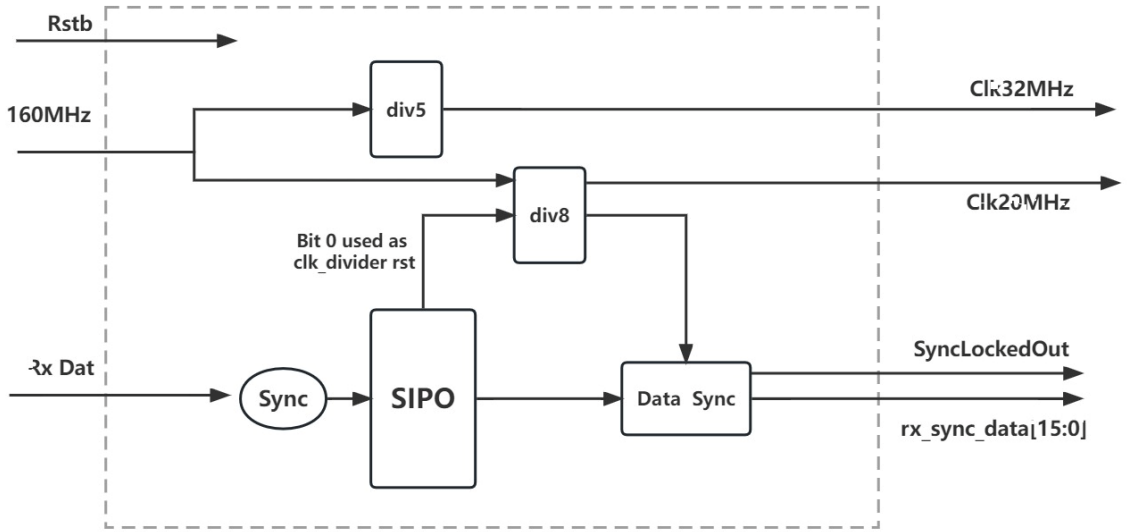


Figure 5 : Structure of SCU

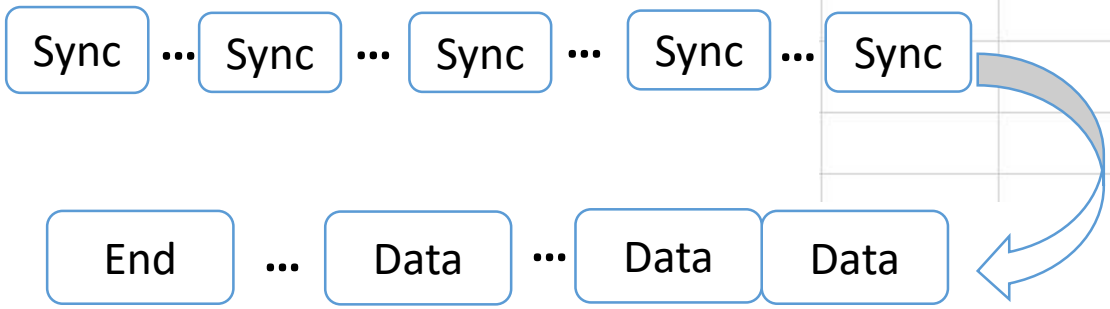
- Synchronization
- CLK_generator
  - Main input 160MHz from external
  - 32MHz -- clk div 5 : internal clk signal for TXU
  - 20MHz -- clk div 8 : for main digital circuits

- **SCU**: divider into 2 parts, general specs:
- Input – clk 160MHz, sync Rstb, RxDat (serial)
- Output – clk 20MHz, clk 32MHz for other parts, 16-bits RxSyncData(parallel)
- **clk_gene**: clk div modify – div5 & div 8 – counter divider ;
- **Sync**: RxDat SIPO converts – serial string input to 16-bit parallel data output;
- match keywords ‘Sync’
- data alignment;
- output ‘SyncLockedOut’;
- determine the start point of data;
- ‘Rst’ signal generation for 20MHz

Sync [15:0]: ‘10000001 01111110’

SCU -- CLK Generator & Sync

- **SCU**:
- Synchronize flow:
- ‘Sync’ detect ,
- Data alignment ,
- Sync complete
- output ‘SyncLockedOut’
- format below



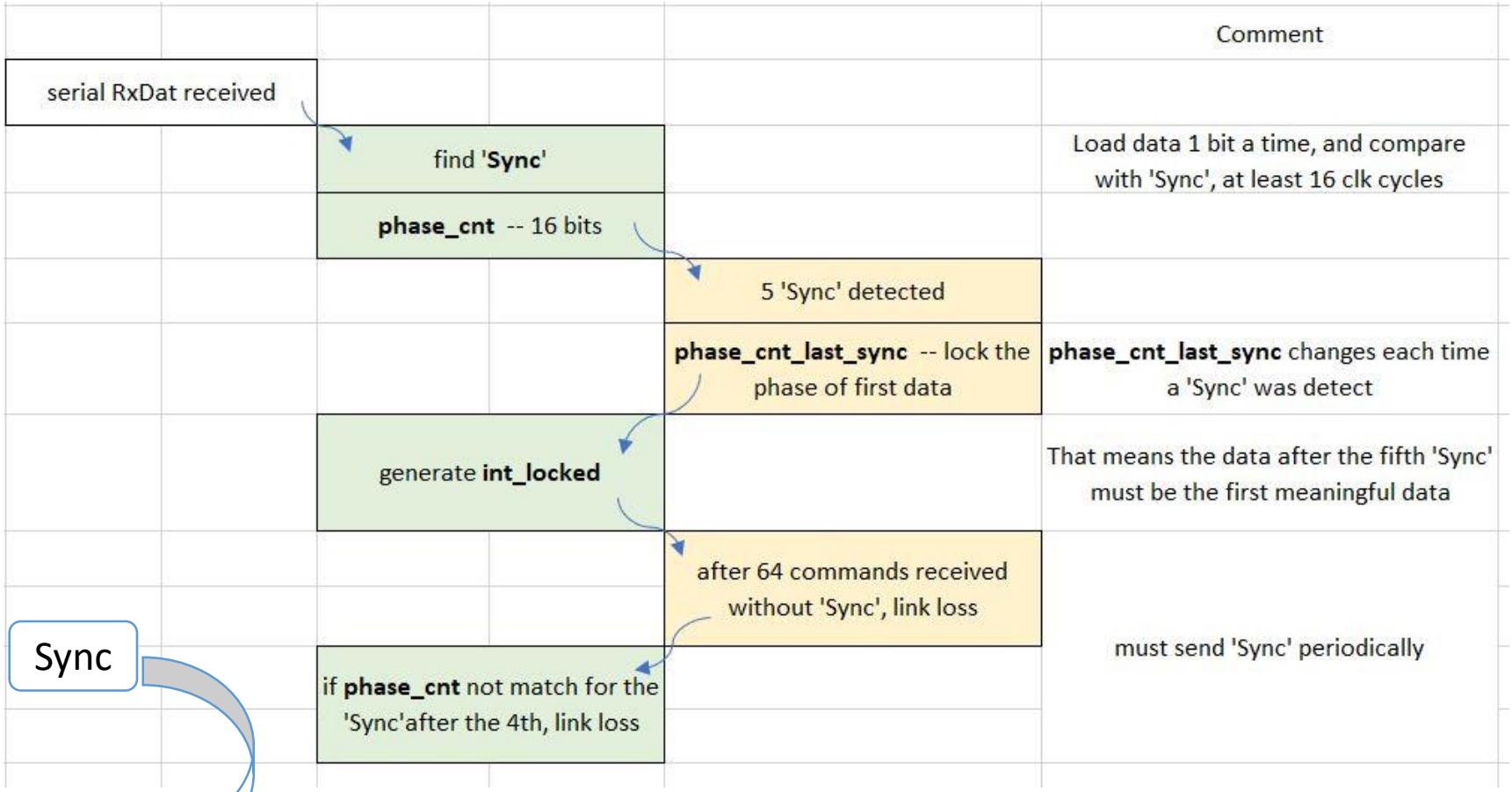


Figure 6 : Synchronization flow

*Case of ‘Lock_Loss’

-- 64 bytes Rx data without ‘Sync’

-- phase not match between two ‘Sync’

CRU -- control unit

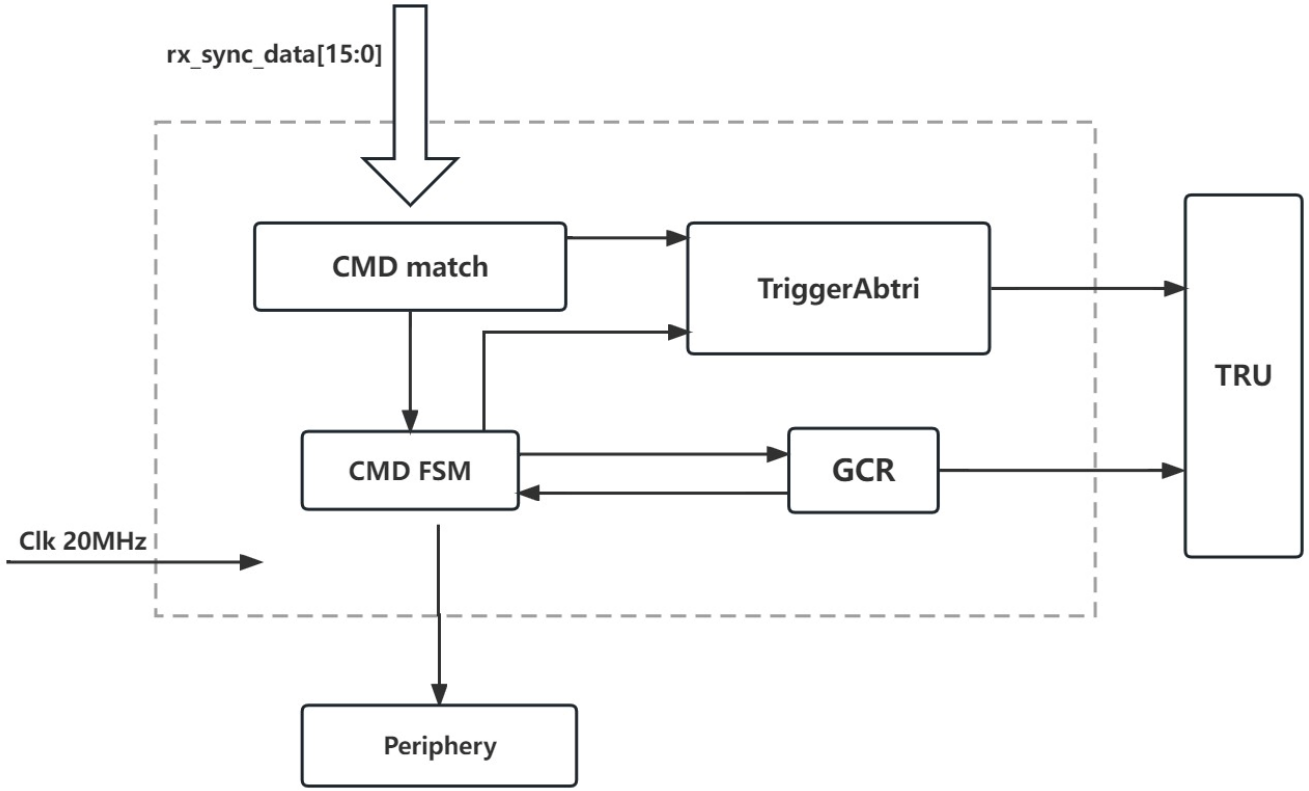


Figure 7 : Structure of CRU

- **CRU**: Use RD53B control protocol from ATLAS CMS;
- **CMD**: almost kept, decoder all the instructions from RxDat to deliver all the commands to other parts of the chip. Most important – **verify the handshake for trigger** between control and chipbottom.
- **GCR**: after collecting all the infos, decide whether to increase or change the default/set register values
- doc/registers.md · main · VTXUpgrade / obelix1 · GitLab (desy.de)

TRU -- trigger unit

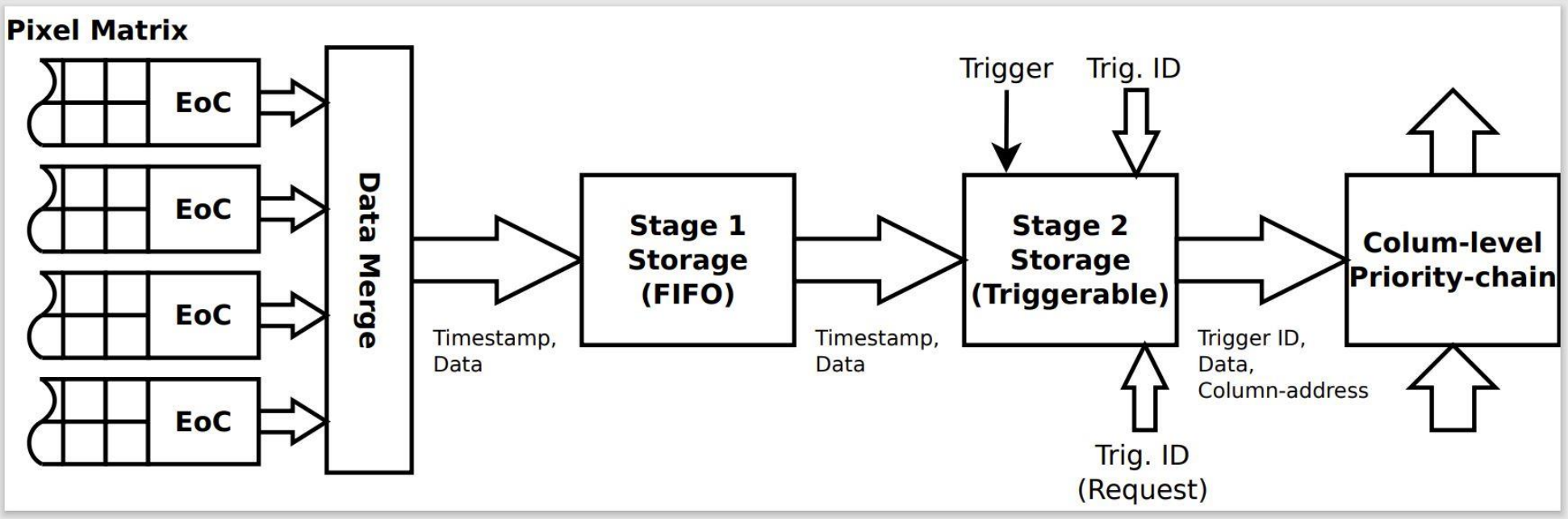


Figure 8 : Structure of Trigger Group

- New End-of-column adapted to Belle II trigger
- - Timestamped hits stored in memories
- - Read-out when timestamps matched with trigger
- - spec: hit rate ≤ 120MHz/cm2
- Trigger memory organized in 112 Trigger Groups(TRGs), each connected to 4 DCs(EoC)
- Eoc: controls of pixel readout, extends 'Le' BCID to 9 bits
- Process the pixel data and wait for trigger commands to transmit

DCs – Double Columns
Le / Te – Leading edge / Trailing edge
BCID – Bunch Crossing ID

TRU -- trigger unit

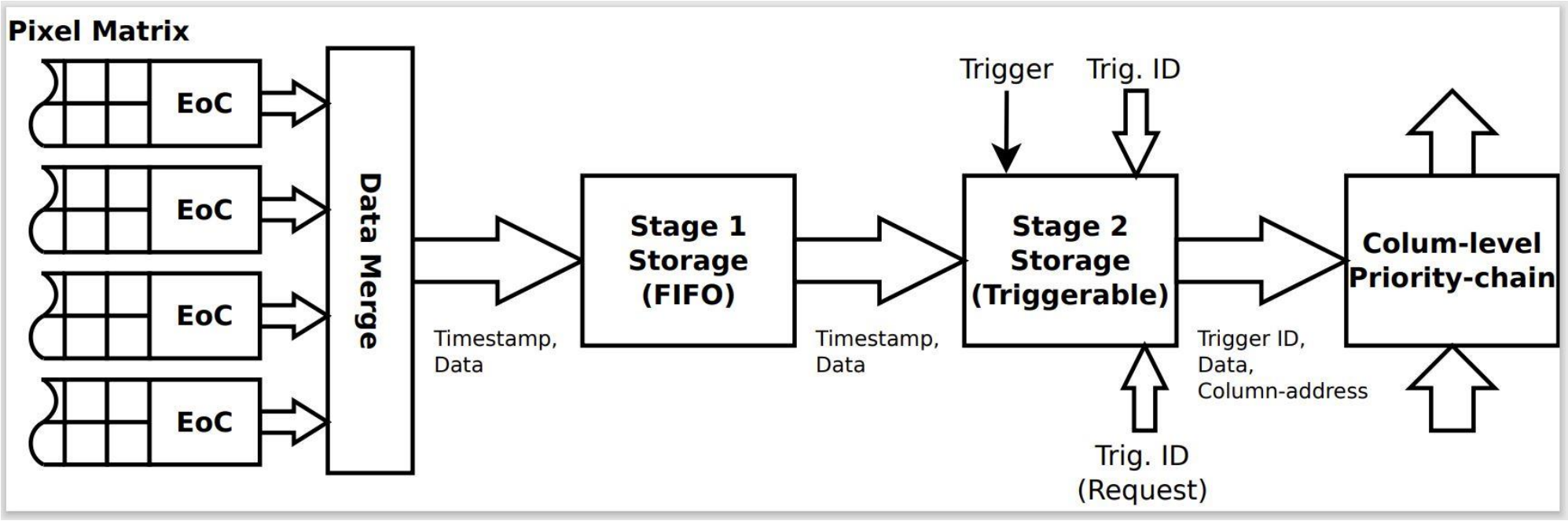


Figure 8 : Structure of Trigger Group

Data Merge:

- Collect hit data from 4DCs via 'round-robin' arbiter

Stage 1 Storage :

Simple power friendly SRAM FIFO

Data is buffered here until a spot in S2 is free

Overload protection: discard data on output if it expired in S1

Stage 2 Storage :

- Data is matched to trigger
- Wait for readout after trigger
- Only works at the end of latency for each hit
- Small, more logic per data-word
- Combine pixel address + data

TXU -- transmission unit

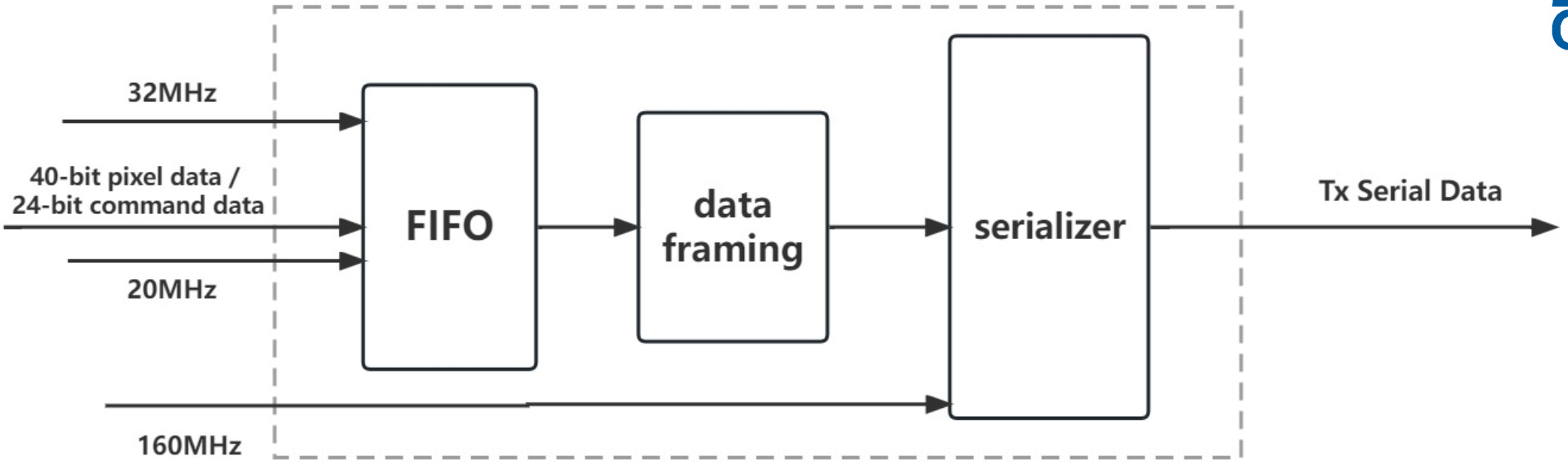


Figure 9 : Structure of Transmission Unit

- **TXU**: divider into 3 parts, general specs:
- FIFO: data buffering, transfer data_rate,
- input clk 20MHz / output clk 32MHz
- Frame Generation: generate data framing necessary
- Serializer: 160MHz, output data to FPGA
- Two packages to transfer :
- command package : IDLE + SOF_C + cmd_addr + cmd + EOF_C + IDLE
- pixel data package : IDLE + SOF_H + pixel data + EOF_H + IDLE

Frame Generation: 4 data types

Pixel data – 35 bits+ 5 bits:

19-bit address – 10 bits 'row' + 3 bits 'DC' inside block + 6 bits 'block'

16-bit timing – 9 bits 'Le' BCID + 7 bits 'Te' BCID

5 bits for selecting – each frame 1 select bit + 7 data bits

CMD data: 8-bit cmd_addr + 16-bit RdReg cmd

Keywords pattern : SOF CMD/HIT, EOF CMD/HIT

IDLE pattern : K28_1 IDLE

TXU -- transmission unit

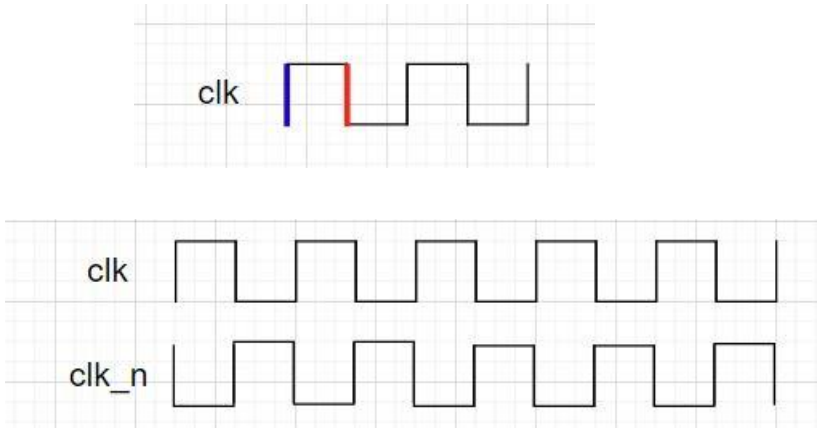


Figure 10 : A pair of clocks with same freq and opposite phase

- Serializer : clk 160 MHz, but a data_rate of 320Mb/s is required
- Input – 10-bit parallel data,
- Baud – 32*10Mbps – 5 clk cycles for 1 data
- Output – serial data stream,
- Baud – 1*160Mbps – using 2 edges as trigger signals, increase to 320Mbps

Solution :

1. generate inverse clock – clk_n, for using both 2 rising edges
2. Separate 10-bit data into 2 groups – odd (1,3,5,7,9)& even (0,2,4,6,8)
3. 'clk' for outputing even group, 'clk_n' for outputing odd group
4. Combine the 2 outputs to achieve 320Mbps

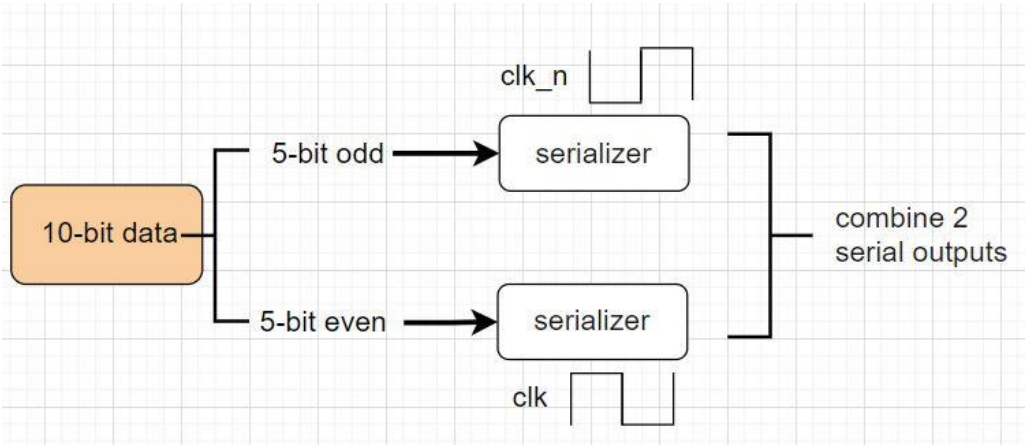


Figure 11 : conception of serializer

difficulty:

1. Ensure 'clk' & 'clk_n' to be fully synchronized – tolerance 20% clk cyle
2. How to combine the 2 outputs without 320MHz clk – use both 2-level of clk for output – simulation passed

OBELIX_V1 : Digital Data Path

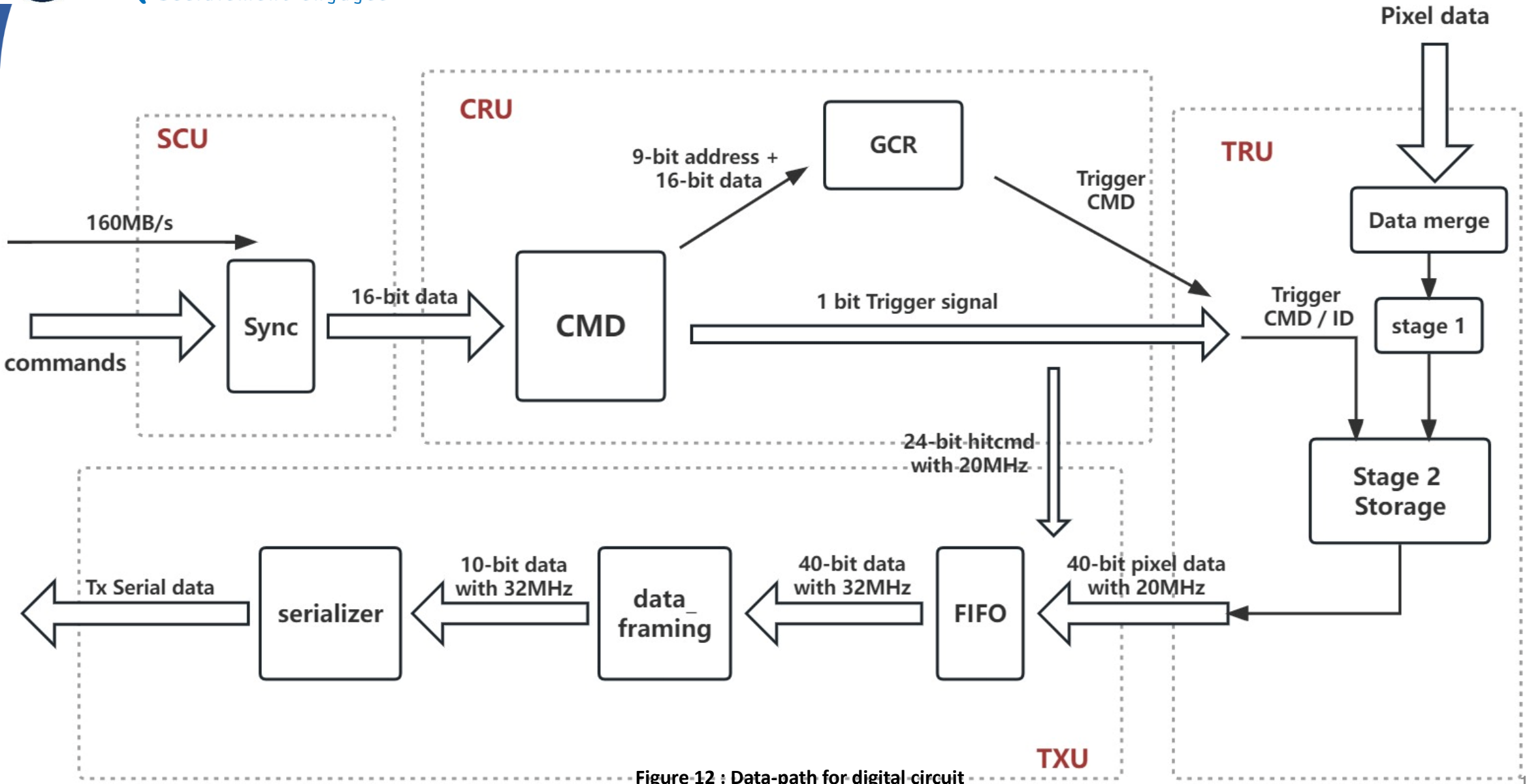


Figure 12 : Data-path for digital circuit

OBELIX_V1 : Digital Clock Path

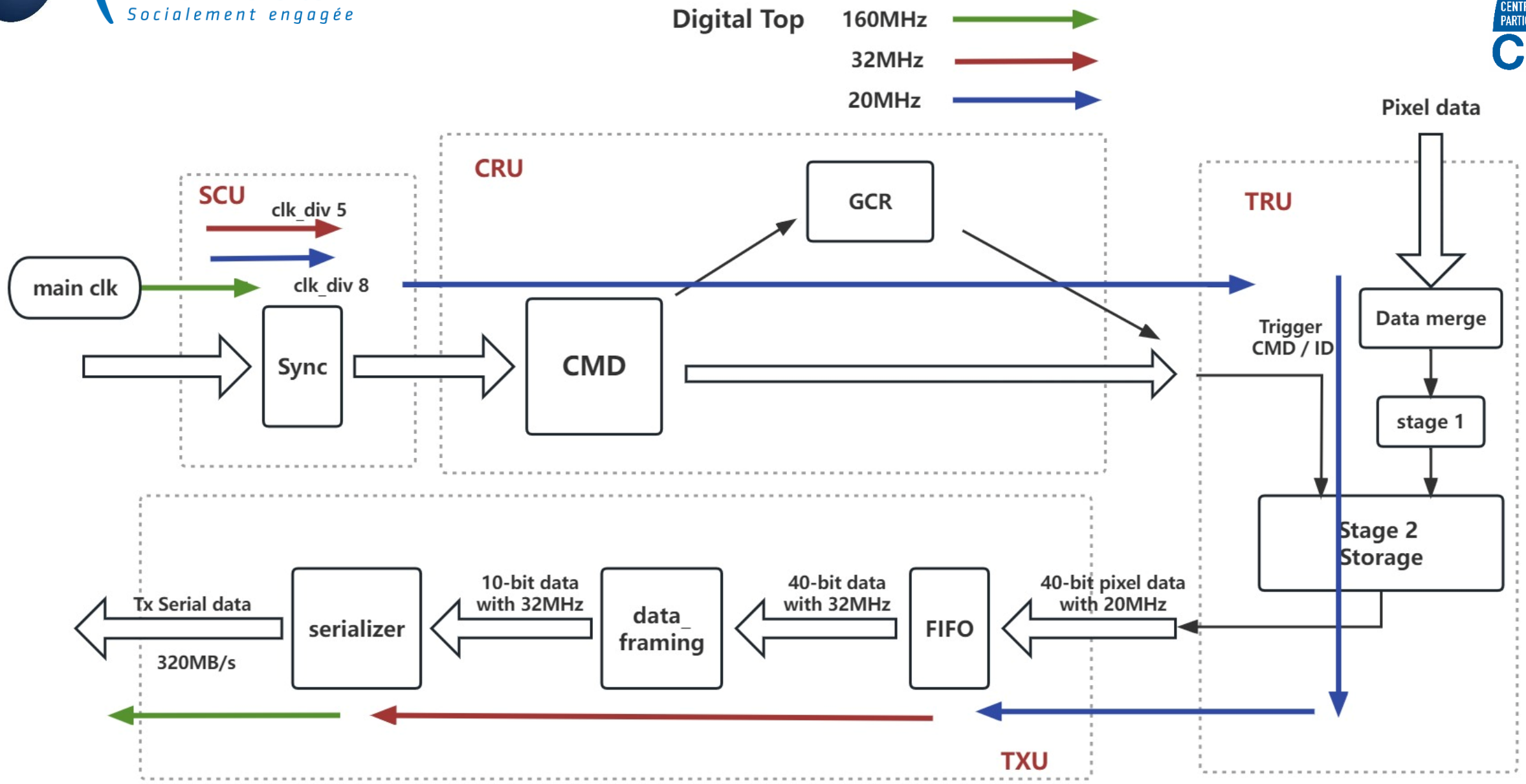


Figure 13 : Clock-path for digital circuit

Conclusions

Current progress and what to do next?

1. Finish the design --- all the RTL design
   -- Register configurations;
   -- Top level Integration

2. Back-end flows for digital top -- write scripts, almost automatically done by the EDA tool
   -- synthesis global
   -- Placement & Routage
   -- Static Timing Analysis
   -- Design Rule Checking / Layout Versus Schematic
   -- Logic Equivalence Checking
   -- Verification & Post simulation

Digital design – an iterative process

Back up slides

ASIC Digital Design Flow Method

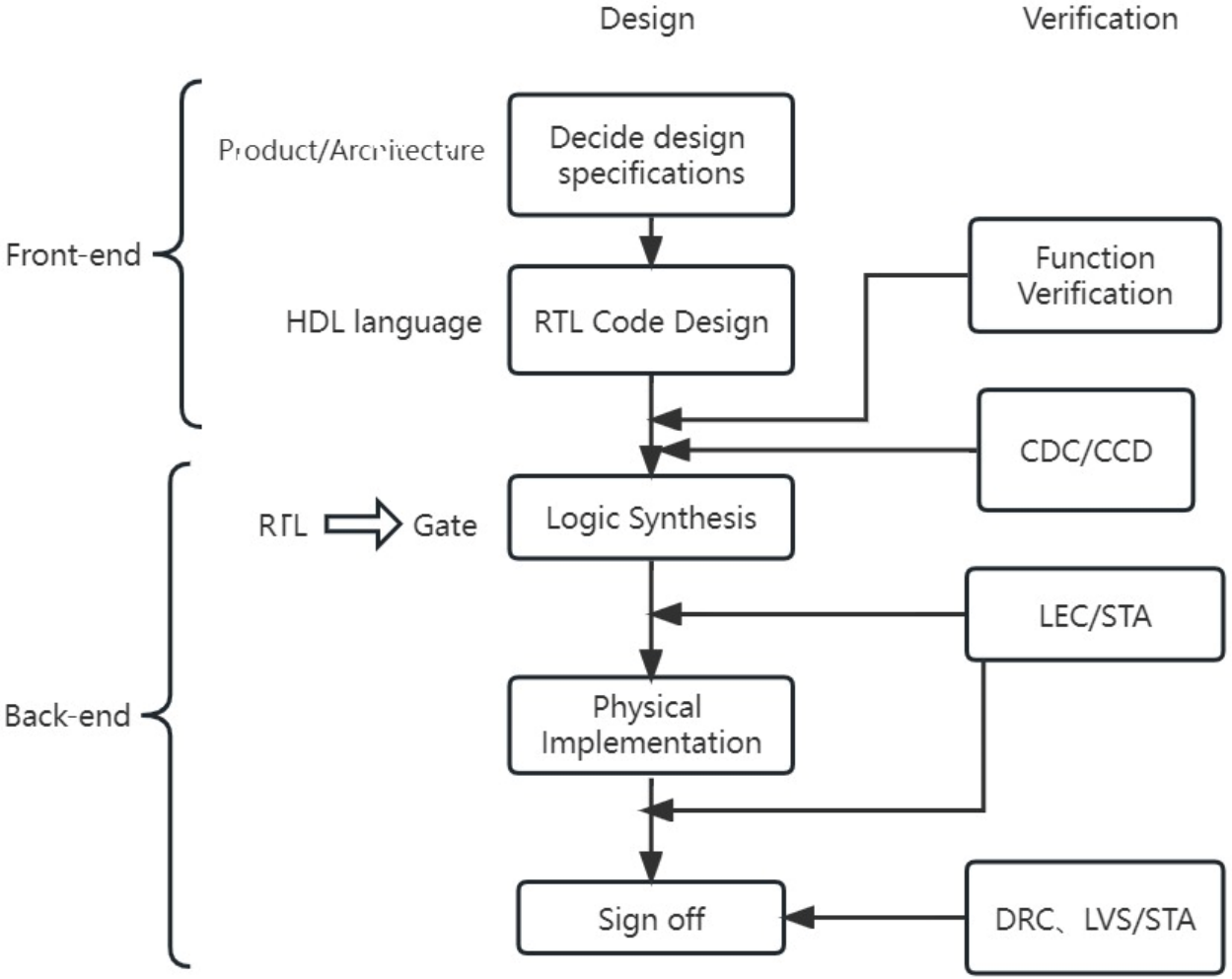


Figure 14 : General design flow for ASIC design

- General ASIC design flow
- **Ensure design specifications**
- **RTL Code Design** : SystemVerilog, Verilog, VHDL, etc
- **Logic synthesis**: Convert logic circuits to gate-level circuits, Genus
- **Placement & Routage:** Convert ideal circuits into real production circuit designs, Innovus
- **Signoff**

- General ASIC verification flow
- **Logical verification**: simulation, xcelium / questa
- **CCD/CDC**: constraint check for syn , clock domain check for rtl
- **Static Timing Analysis** : check timing violation, tempus
- **LEC check**: verify design consistency, conformal
- **DRC, LVS**: design rule check & layout versus schematic, Innovus
- **Verification:** post-simulation, verification, UVM

OBELIX_V1 : Digital Clock Path

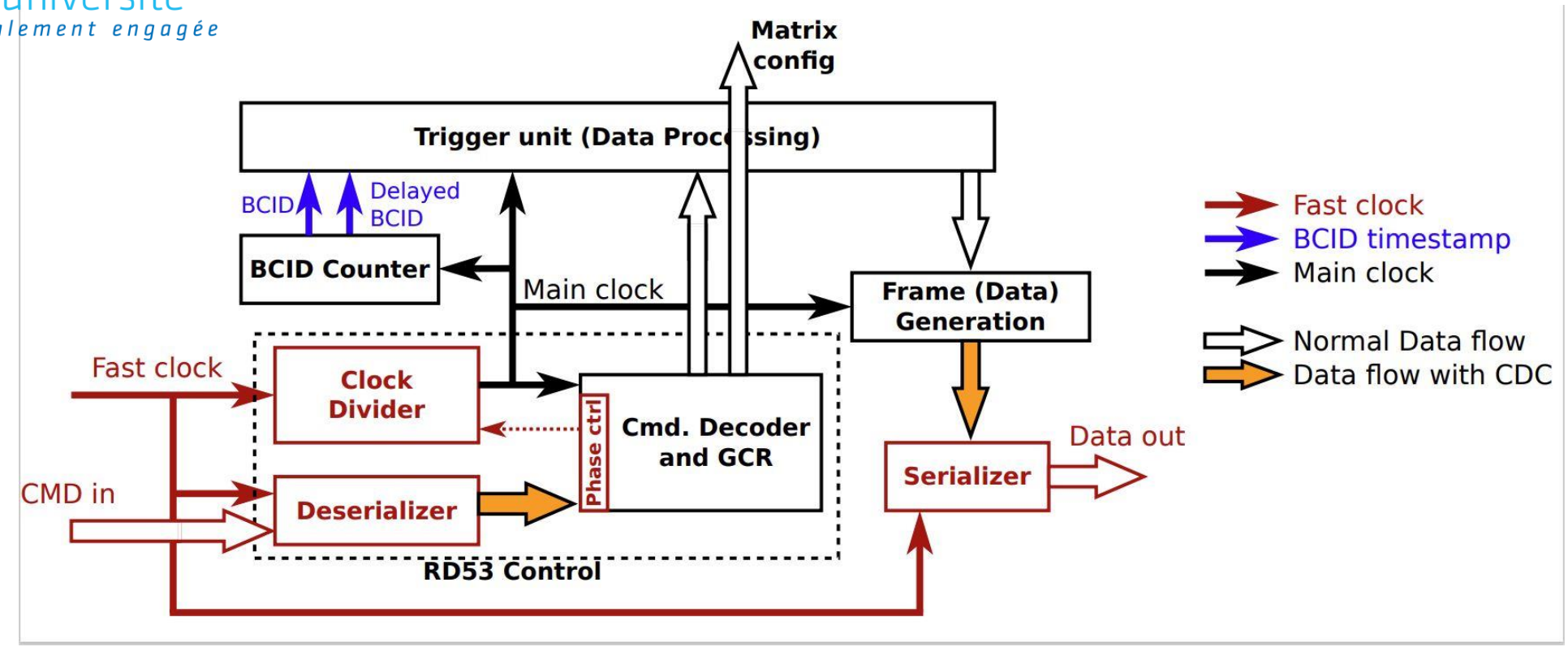


Figure 15 : Clock domain of Obelix digital top

Clock Signal	Ideal	Alignment RF Clk
Main Clock	160MHz	169.7MHz
Divider 8 Clock	20 MHz	21.2 MHz
BCID Clock	20 MHz	21.2 MHz
TXU Clock	32MHz	33.9MHz

Table 1 : CLK frequency for degital design and RF clk

Thanks for your attention