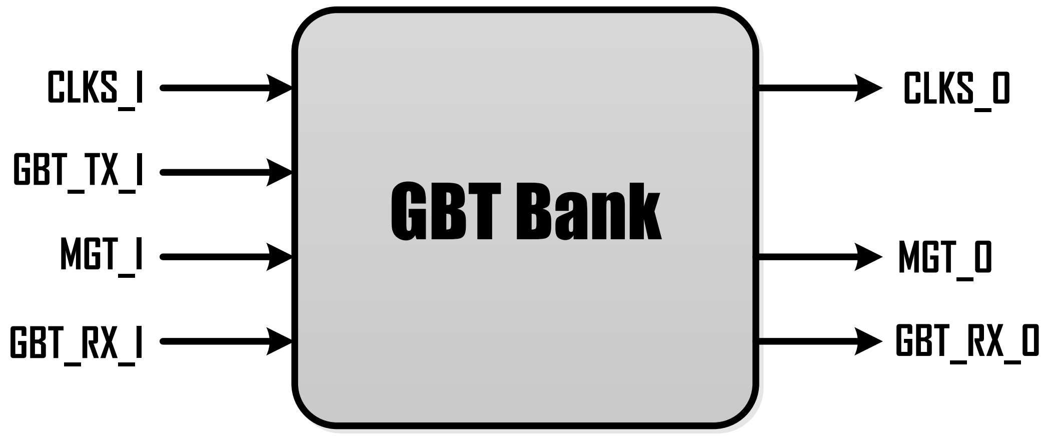
GBT-FPGA

One unified core for multiple users



PH-ESE-BE

PH-ESE-BE Students/Fellows Seminar
(05/02/2014)

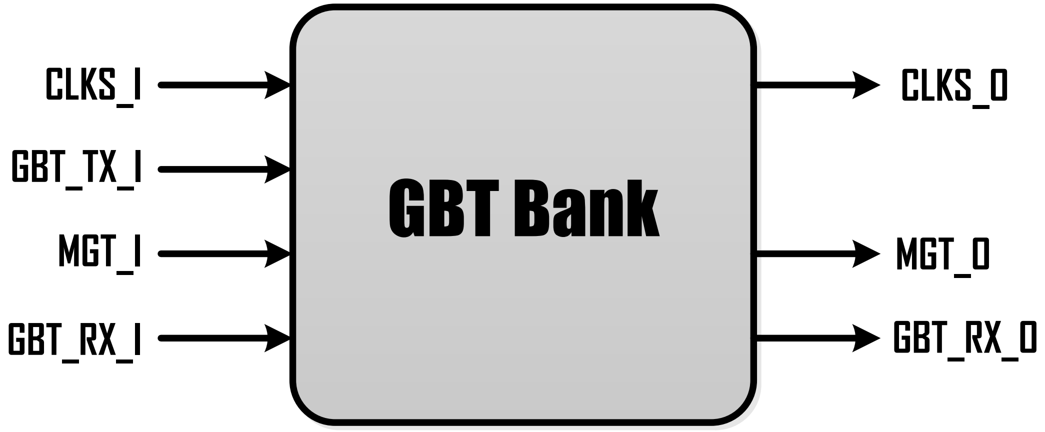
Manoel Barros Marin

GBT-FPGA

One unified core for multiple users

Outline:

- **The context: “multiple needs”**
- **Our approach: “one unified core”**
- **The latency optimization nightmare**
- **Status & Outlook**



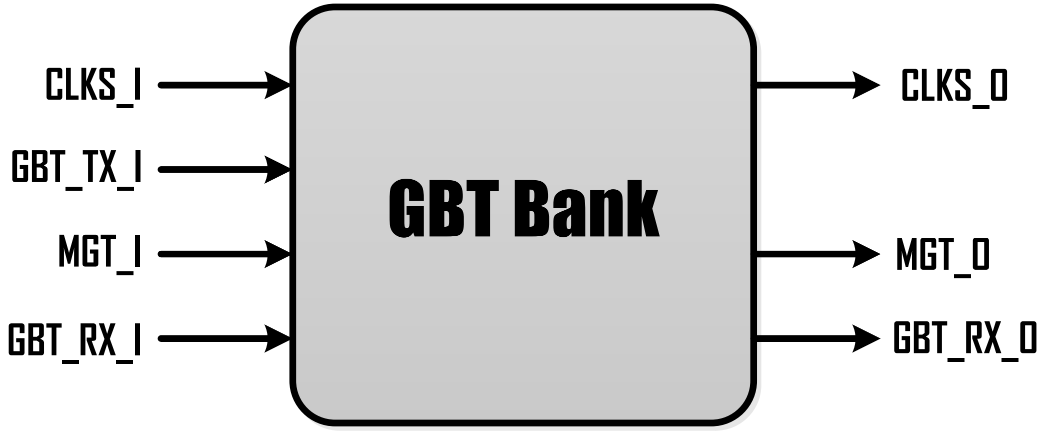
PH-ESE-BE

GBT-FPGA

One unified core for multiple users

Outline:

- **The context: “multiple needs”**
- Our approach: “one unified core”
- The latency optimization nightmare
- Status & Outlook



PH-ESE-BE

The context: "multiple needs" (1 of 5)

PH-ESE-BE

Multiple Categories of Users

- **Multiple profiles**
  - Beginners
  - Experts
- **All LHC experiments**
  - CMS
  - ATLAS
  - ALICE
  - LHCb
- **Accelerators**
  - LHC (Beams Instrumentation)
  - CLIC

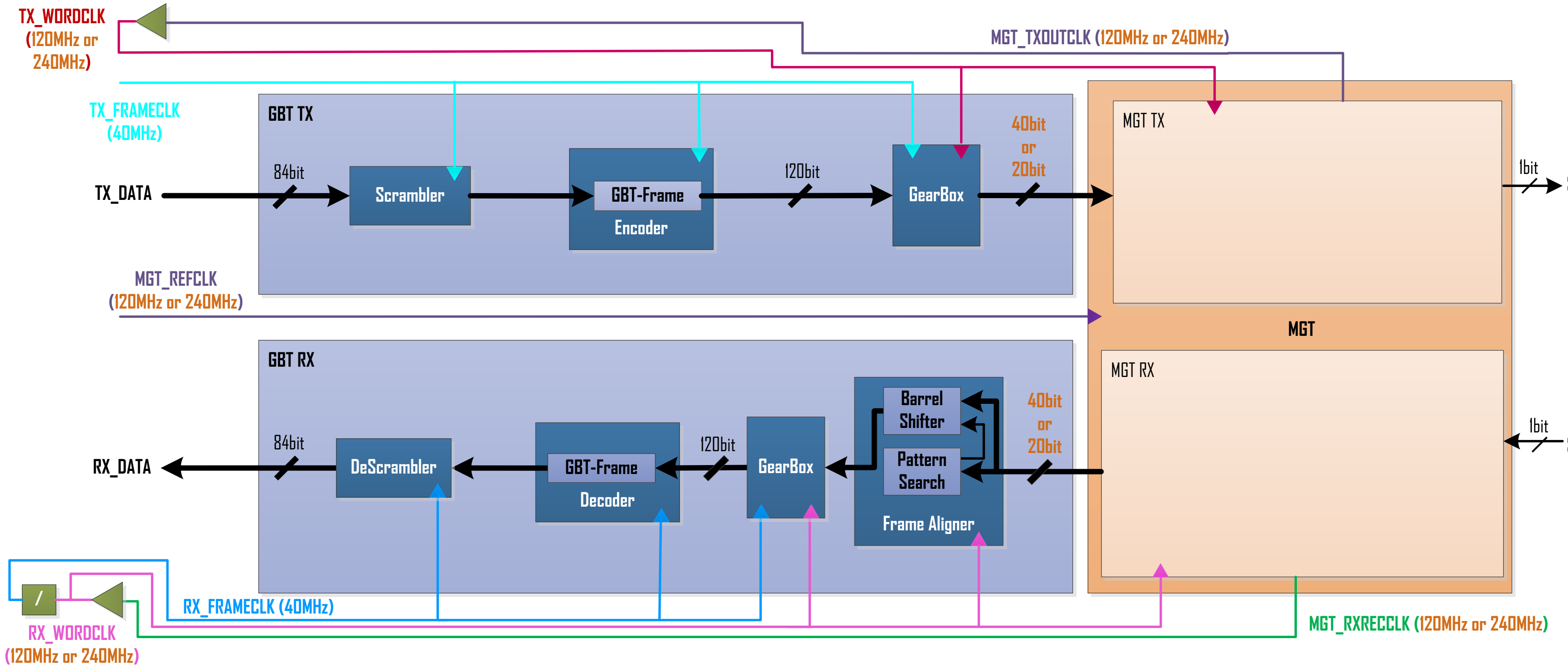
83 registered users so far…

The context: “multiple needs” (2 of 5)

PH-ESE-BE

Multiple Platforms

- **Xilinx:** Virtex 5, Virtex 6, Kintex 7, Virtex 7…
- **Altera:** Stratix V, Cyclone V, ….
- **Microsemi?:** SmartFusion2, Igloo2 (Rad-Hard FPGAs)

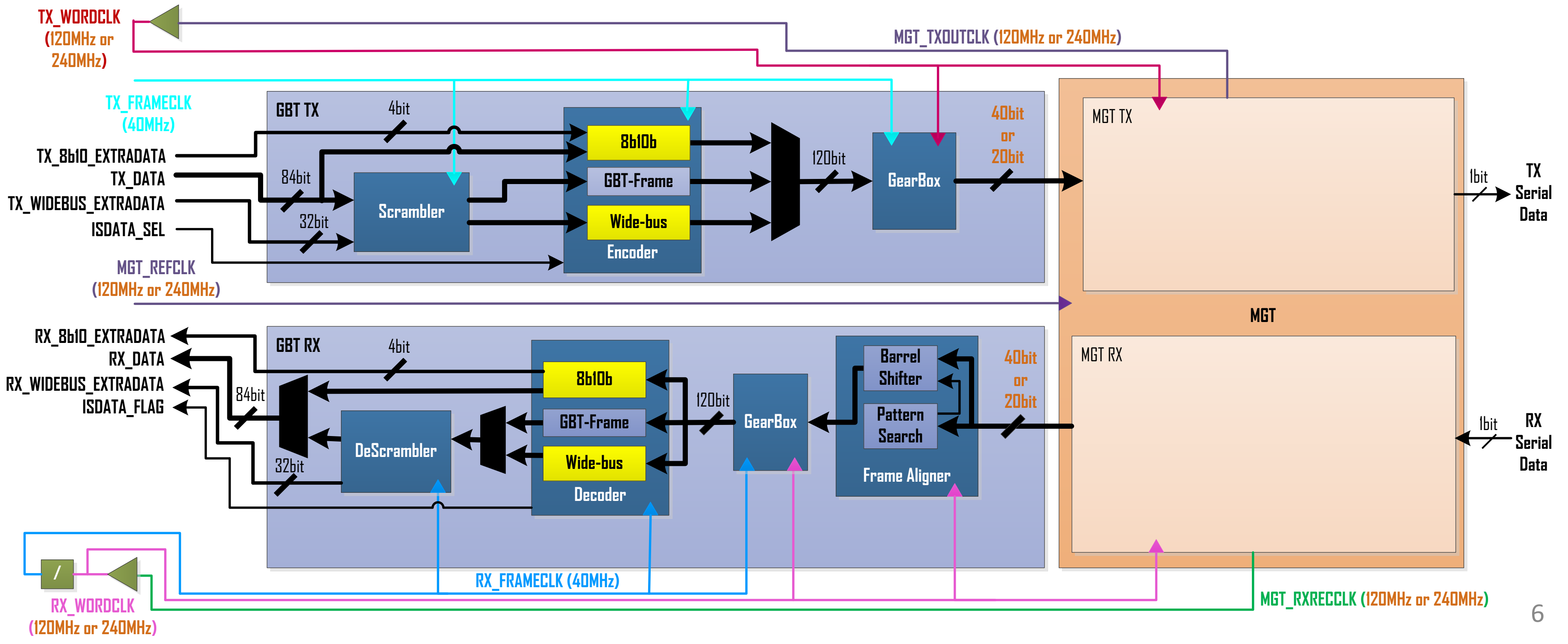


The context: "multiple needs" (3 of 5)

PH-ESE-BE

Multiple Configurations

- Encoding

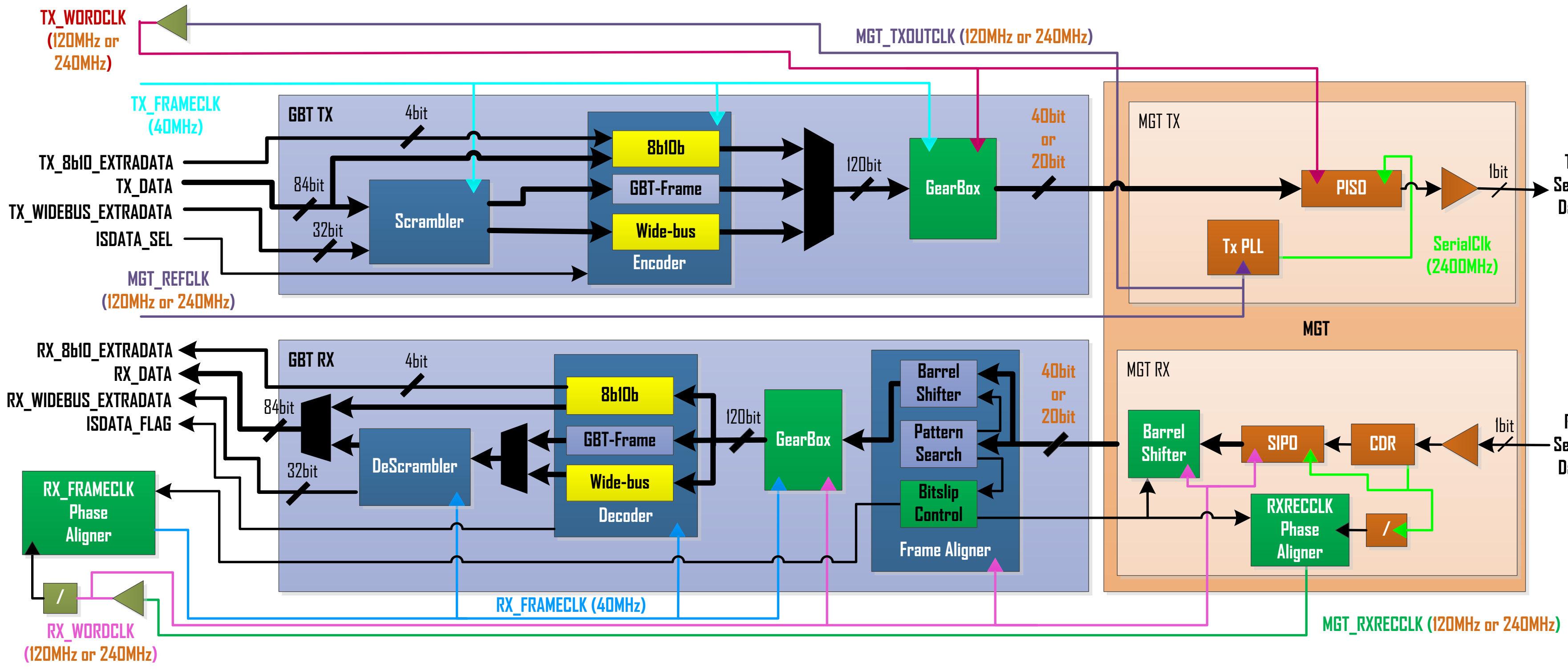


The context: “multiple needs” (4 of 5)

PH-ESE-BE

Multiple Configurations

- Standard (STD) OR Latency Optimized (LATOPT)

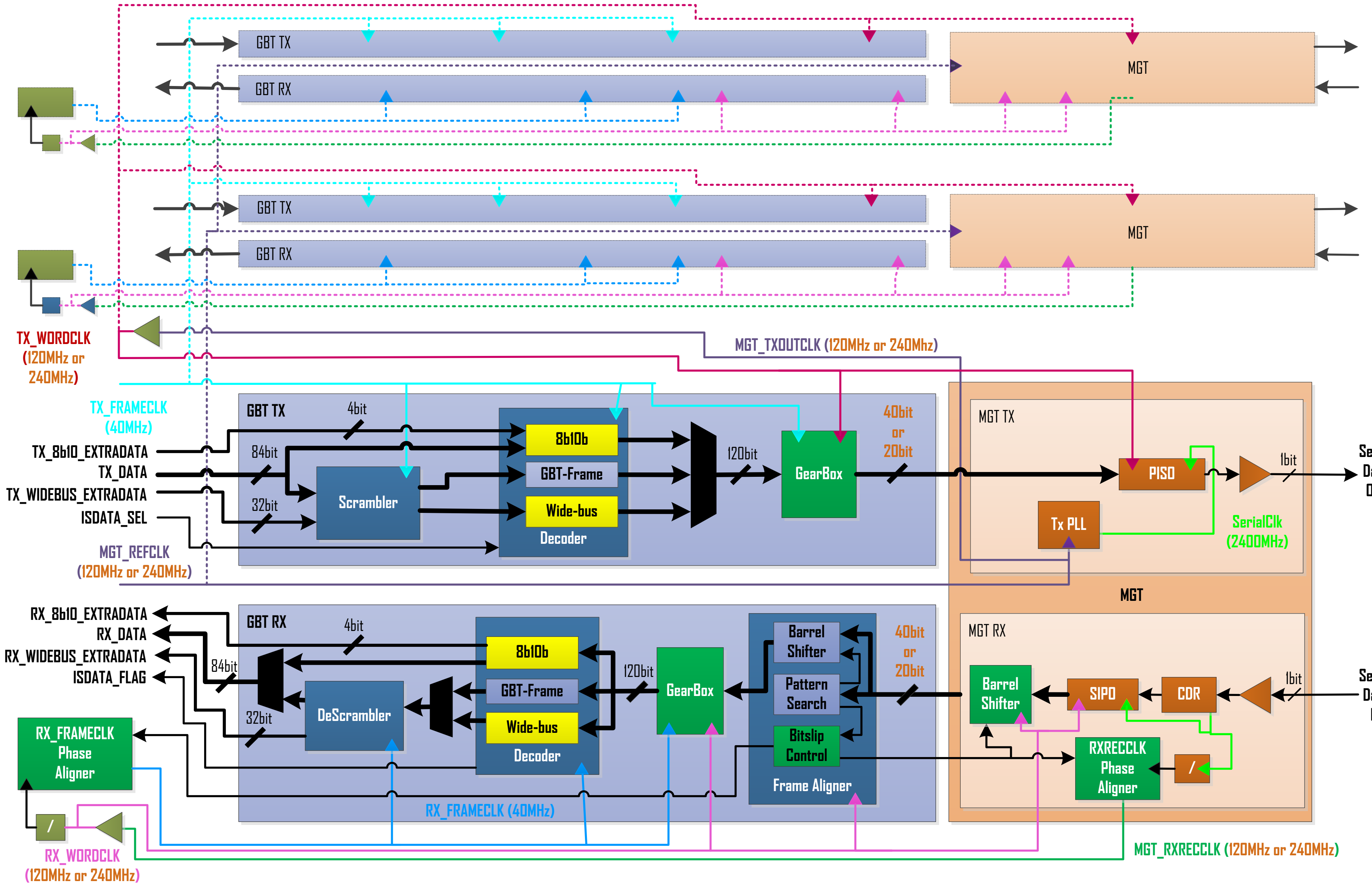


The context: "multiple needs" (5 of 5)

PH-ESE-BE

Multiple Configurations

- Number of links

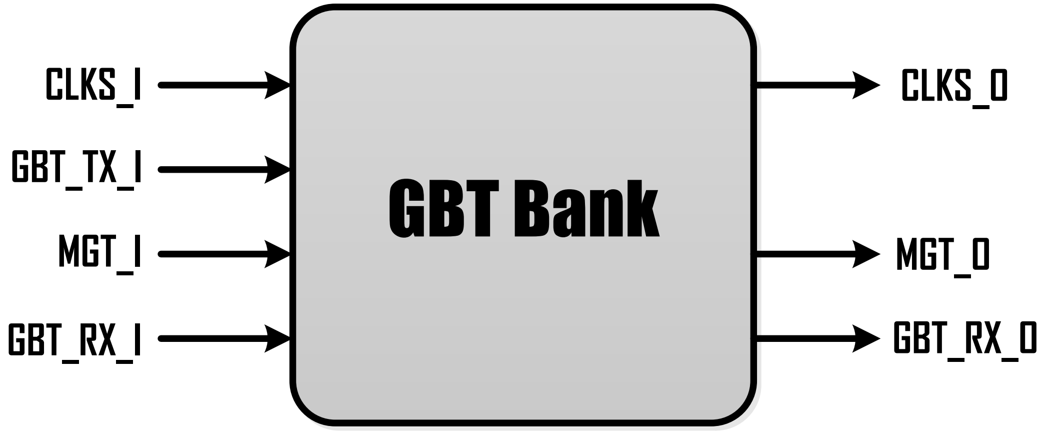


GBT-FPGA

One unified core for multiple users

Outline:

- The context: "multiple needs"
- **Our approach: "one unified core"**
- The latency optimization nightmare
- Status & Outlook



PH-ESE-BE

Our approach: “one unified core” (1 of 7)

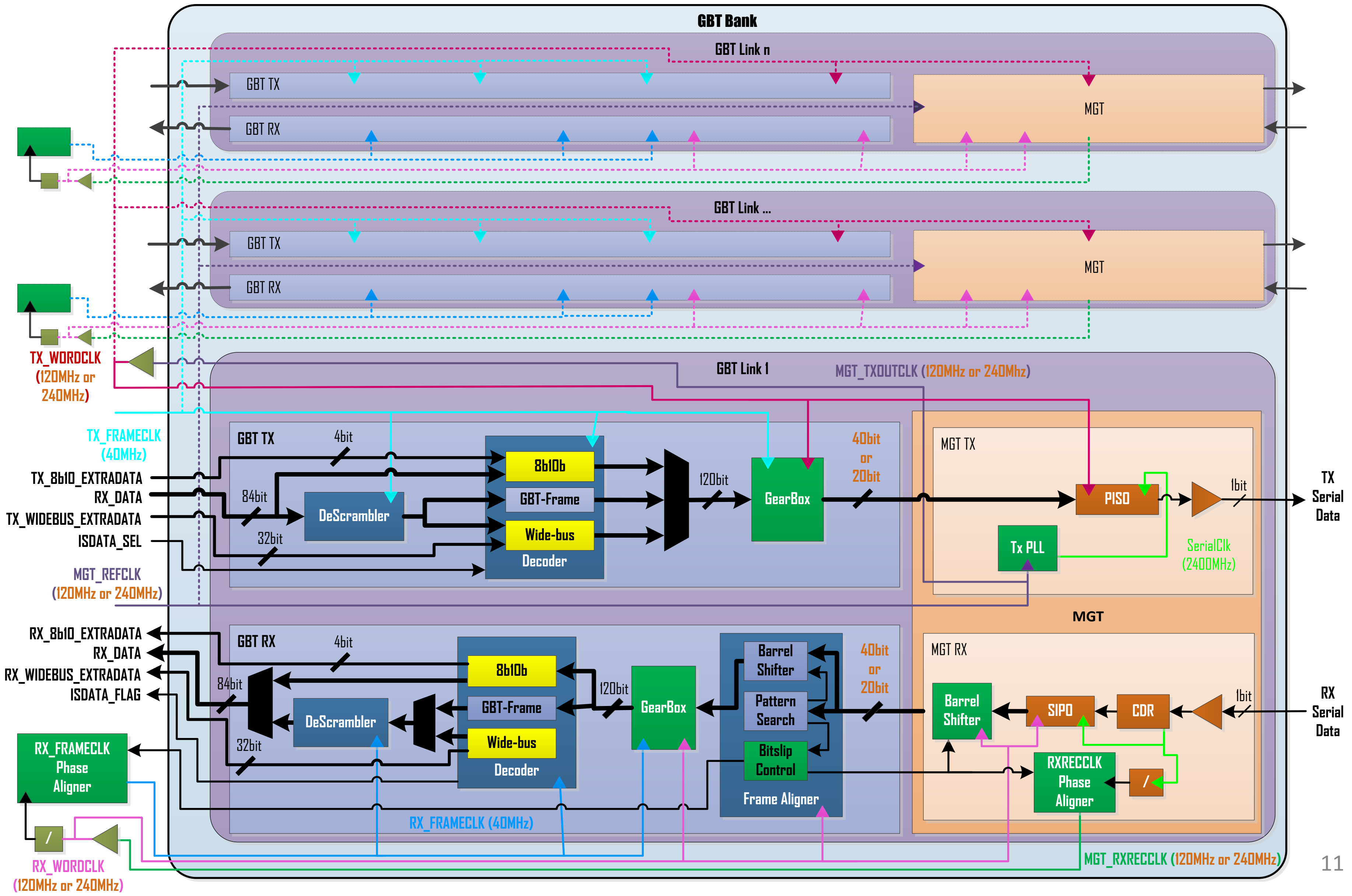
Easy to use & Easy to support

This is a big **CHALLENGE**

Our approach: "one unified core" (2 of 7)

PH-ESE-BE

GBT Bank

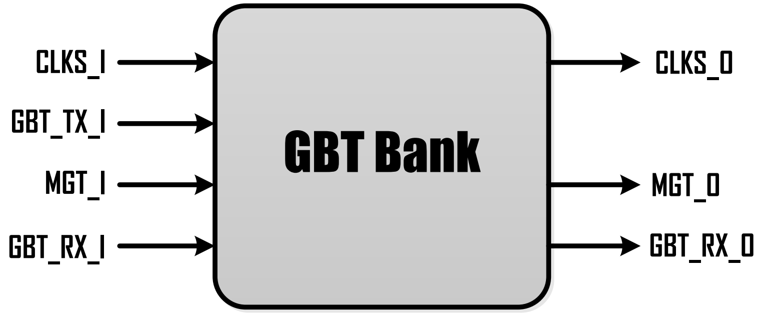


Our approach: "one unified core" (2 of 7)

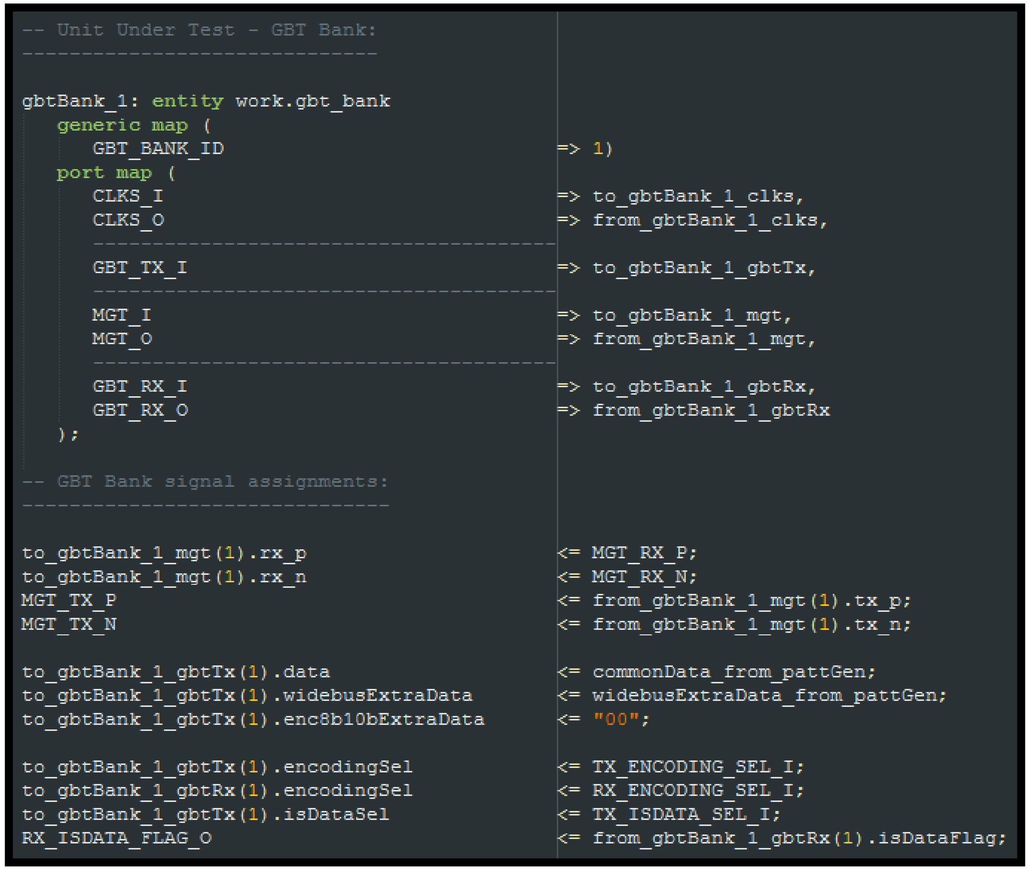
PH-ESE-BE

GBT Bank

- Main module of the GBT-FPGA
- One unified core for multiple users, platforms & configurations



GBT Bank instantiation (VHDL)



```
-- Unit Under Test - GBT Bank:
-------------------------------

gbtBank_1: entity work.gbt_bank
   generic map (
      GBT_BANK_ID                            => 1)
   port map (
      CLKS_I                                 => to_gbtBank_1_clks,
      CLKS_O                                 => from_gbtBank_1_clks,
      ----------------------------------------
      GBT_TX_I                               => to_gbtBank_1_gbtTx,
      ----------------------------------------
      MGT_I                                  => to_gbtBank_1_mgt,
      MGT_O                                  => from_gbtBank_1_mgt,
      ----------------------------------------
      GBT_RX_I                               => to_gbtBank_1_gbtRx,
      GBT_RX_O                               => from_gbtBank_1_gbtRx
   );

-- GBT Bank signal assignments:
--------------------------------

to_gbtBank_1_mgt(1).rx_p                     <= MGT_RX_P;
to_gbtBank_1_mgt(1).rx_n                     <= MGT_RX_N;
MGT_TX_P                                     <= from_gbtBank_1_mgt(1).tx_p;
MGT_TX_N                                     <= from_gbtBank_1_mgt(1).tx_n;

to_gbtBank_1_gbtTx(1).data                   <= commonData_from_pattGen;
to_gbtBank_1_gbtTx(1).widebusExtraData       <= widebusExtraData_from_pattGen;
to_gbtBank_1_gbtTx(1).enc8b10bExtraData      <= "00";

to_gbtBank_1_gbtTx(1).encodingSel            <= TX_ENCODING_SEL_I;
to_gbtBank_1_gbtRx(1).encodingSel            <= RX_ENCODING_SEL_I;
to_gbtBank_1_gbtTx(1).isDataSel              <= TX_ISDATA_SEL_I;
RX_ISDATA_FLAG_O                             <= from_gbtBank_1_gbtRx(1).isDataFlag;
```

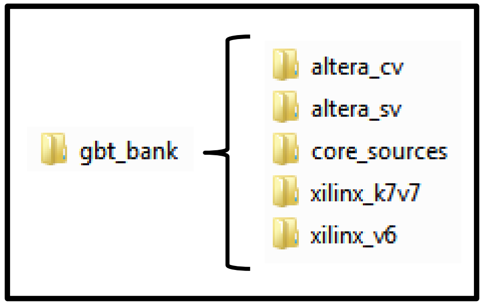
Our approach: "one unified core" (3 of 7)

PH-ESE-BE

HDL Code

- **Consistent**
  - Only one HDL language (VHDL)
  - Use of templates
- **Common code as much as possible (Core Sources)**
  - Minimizes the number of files
  - Small number of vendor specific files:
    - One dedicated package per vendor
    - Dedicated IP Cores from vendor (RAM, MGT)
  - Facilitates User Support & Maintenance
- **Structure & Hierarchy as flat as possible**
  - Avoids too long directory structure issues
  - Facilitates navigation through folders

Common (Core) & Vendor specific sources

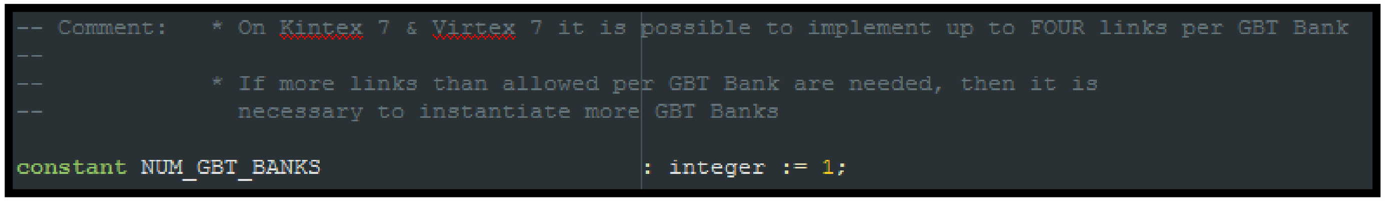


Our approach: “one unified core” (3 of 6)

PH-ESE-BE

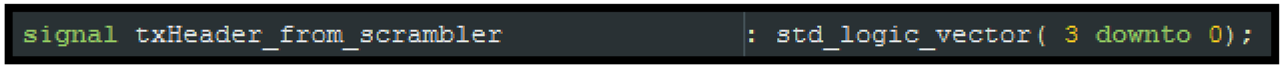
HDL Code

- **Auto-explained**
  - Multiple in-code comments



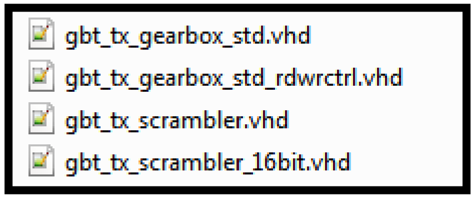
```
-- Comment:   * On Kintex 7 & Virtex 7 it is possible to implement up to FOUR links per GBT Bank
--
--            * If more links than allowed per GBT Bank are needed, then it is
--              necessary to instantiate more GBT Banks

constant NUM_GBT_BANKS                             : integer := 1;
```

  - In-code elements (signals, modules, etc.) names



```
signal txHeader_from_scrambler                     : std_logic_vector( 3 downto 0);
```

  - File names



gbt_tx_gearbox_std.vhd
gbt_tx_gearbox_std_rdwrctrl.vhd
gbt_tx_scrambler.vhd
gbt_tx_scrambler_16bit.vhd

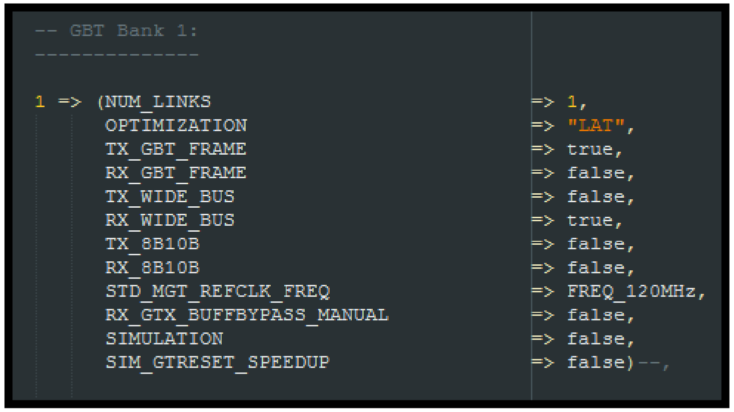
Our approach: "one unified core" (3 of 7)

PH-ESE-BE

HDL Code

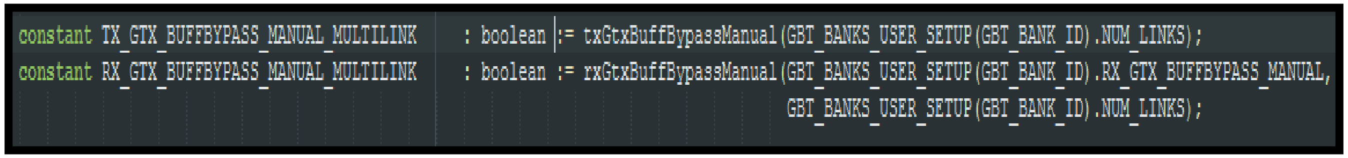
- **Multi-configurable**
  - All features set at implementation time:
    - Encodings
    - Number of GBT Links
    - Etc.
  - Only one file to modify by the user (GBT Bank User Configuration File)
  - Achieved through:
    - Well though design
    - Advanced VHDL coding style:
      - *Packages*
      - *Custom types*
      - *Generics*
      - *Records*
      - *Generates*
      - *Functions*
      - *…*

Extract from GBT User Configuration File



```
-- GBT Bank 1:
--------------

1 => (NUM_LINKS                                   => 1,
      OPTIMIZATION                                => "LAT",
      TX_GBT_FRAME                                => true,
      RX_GBT_FRAME                                => false,
      TX_WIDE_BUS                                 => false,
      RX_WIDE_BUS                                 => true,
      TX_8B10B                                    => false,
      RX_8B10B                                    => false,
      STD_MGT_REFCLK_FREQ                         => FREQ_120MHz,
      RX_GTX_BUFFBYPASS_MANUAL                    => false,
      SIMULATION                                  => false,
      SIM_GTRESET_SPEEDUP                         => false)--,
```

Example of VHDL coding style



```
constant TX_GTX_BUFFBYPASS_MANUAL_MULTILINK    : boolean := txGtxBuffBypassManual(GBT_BANKS_USER_SETUP(GBT_BANK_ID).NUM_LINKS);
constant RX_GTX_BUFFBYPASS_MANUAL_MULTILINK    : boolean := rxGtxBuffBypassManual(GBT_BANKS_USER_SETUP(GBT_BANK_ID).RX_GTX_BUFFBYPASS_MANUAL,
                                                                                   GBT_BANKS_USER_SETUP(GBT_BANK_ID).NUM_LINKS);
```

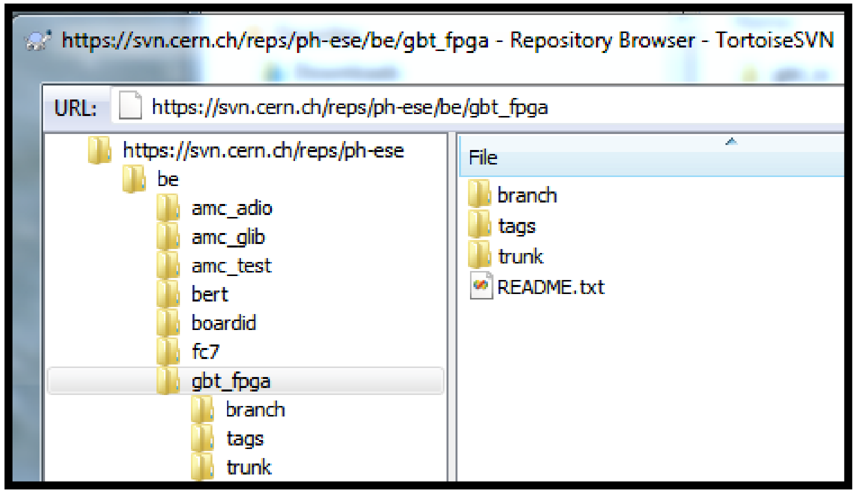
Our approach: "one unified core" (4 of 7)

PH-ESE-BE

Download, Installation & Update

- **SVN repository**
  - Single point of access
  - Facilitates Download, Update & User Support

TortoiseSVN



https://svn.cern.ch/reps/ph-ese/be/gbt_fpga - Repository Browser - TortoiseSVN

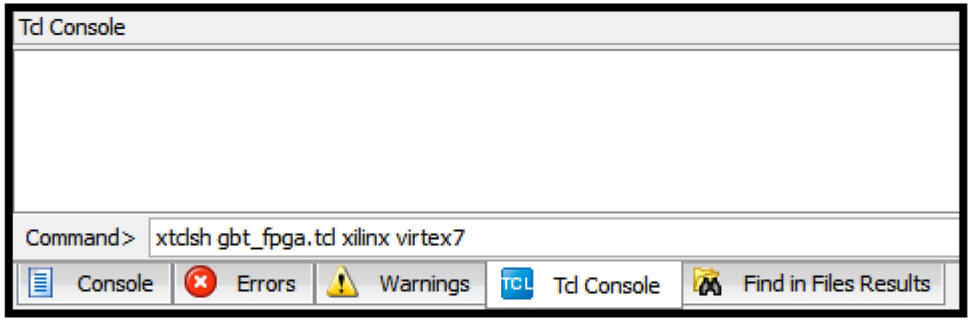
URL: https://svn.cern.ch/reps/ph-ese/be/gbt_fpga

```
https://svn.cern.ch/reps/ph-ese
  be
    amc_adio
    amc_glib
    amc_test
    bert
    boardid
    fc7
    gbt_fpga
      branch
      tags
      trunk
```

File
branch
tags
trunk
README.txt

- **TCL scripts**
  - Facilitates addition of GBT Bank into HDL projects:
    - ISE project (Xilinx)
    - Quartus II project (Altera)

Example of TCL script invocation



Tcl Console

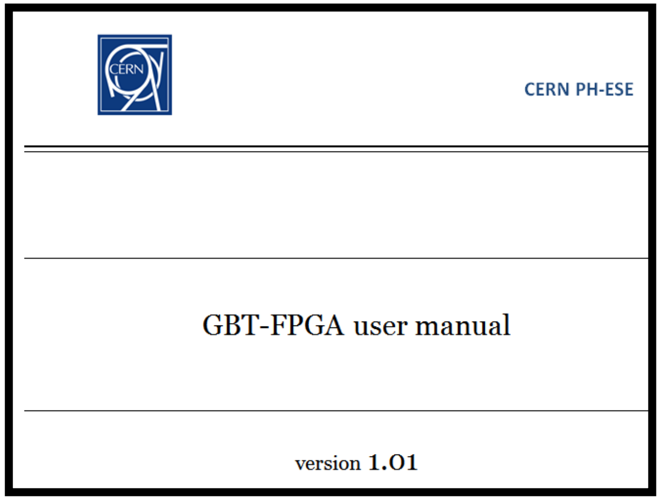
```
Command> xtclsh gbt_fpga.tcl xilinx virtex7
```

Our approach: "one unified core" (5 of 7)

PH-ESE-BE

Documentation & User Support

- **Documentation**
  - Version for main FPGA devices & Evaluation kits

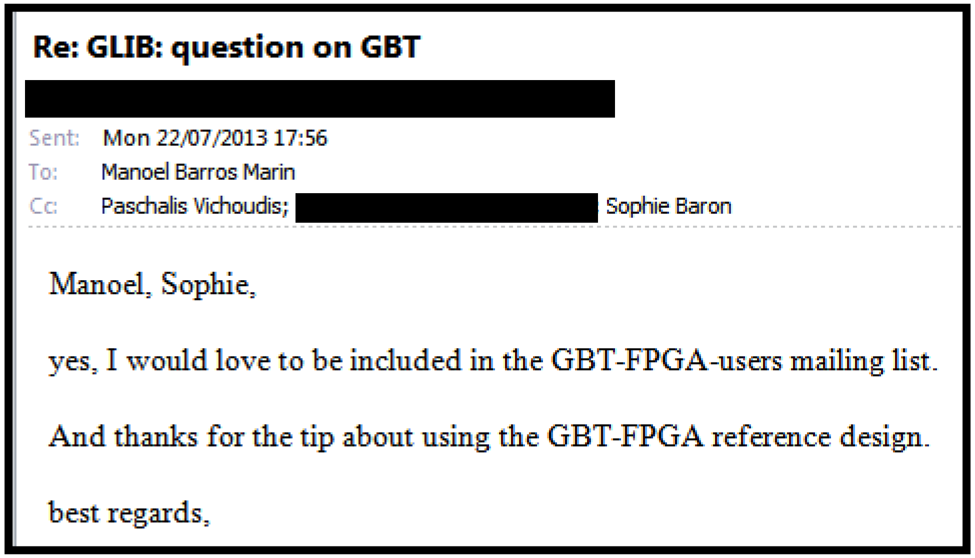


CERN PH-ESE

GBT-FPGA user manual

version 1.01

- **User Support:**
  - GBT-FPGA email list (83 users so far…)
  - Contact with the GBT-FPGA team



Re: GLIB: question on GBT

[redacted]

Sent: Mon 22/07/2013 17:56

To: Manoel Barros Marin

Cc: Paschalis Vichoudis; [redacted] Sophie Baron

Manoel, Sophie,

yes, I would love to be included in the GBT-FPGA-users mailing list.

And thanks for the tip about using the GBT-FPGA reference design.

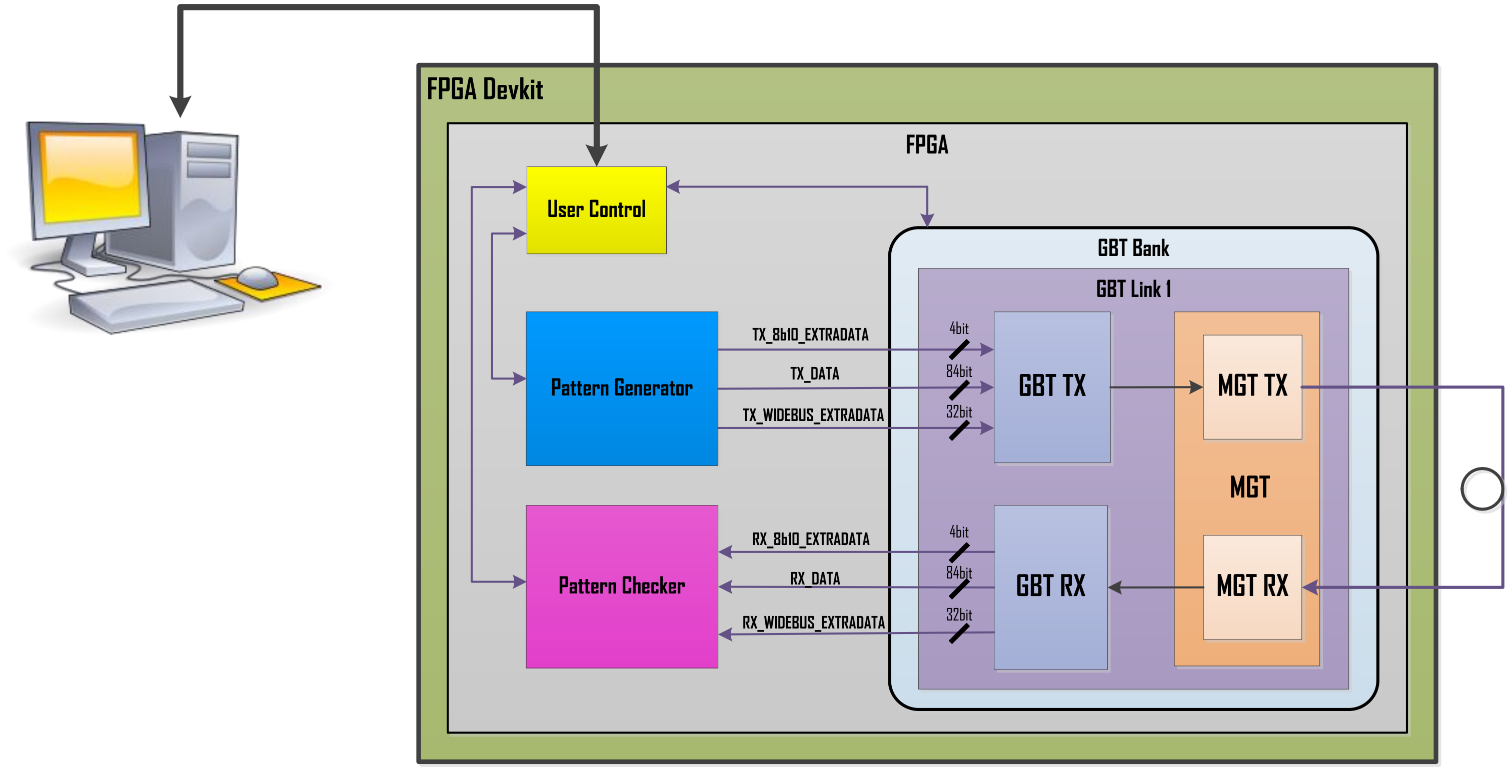
best regards,

Our approach: "one unified core" (6 of 7)

PH-ESE-BE

Example Designs

- Version for main evaluation kits



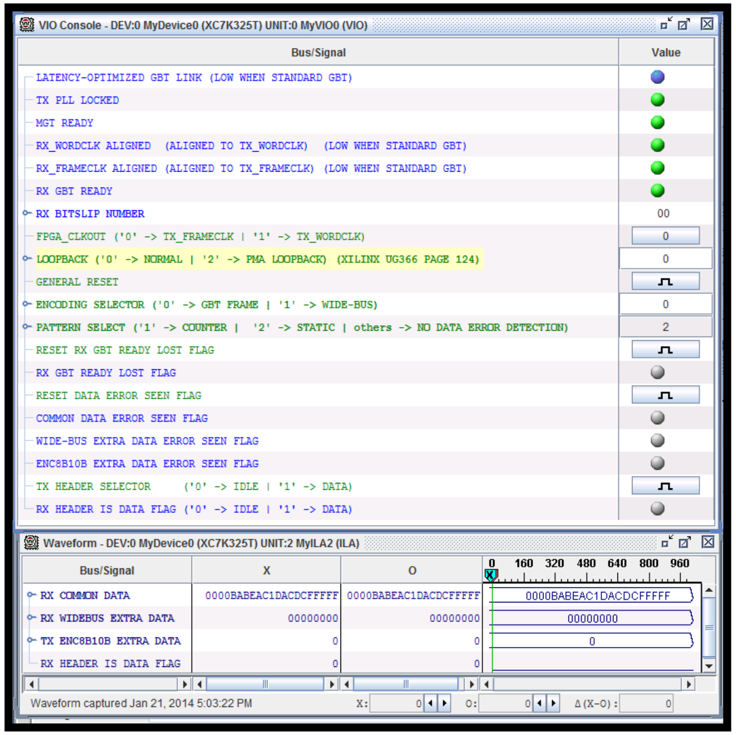
Our approach: "one unified core" (6 of 7)

PH-ESE-BE

Example Designs

- Version for main evaluation kits
- Simplified user control

Xilinx: ChipScope (VIO & ILA)



VIO Console - DEV:0 MyDevice0 (XC7K325T) UNIT:0 MyVIO0 (VIO)

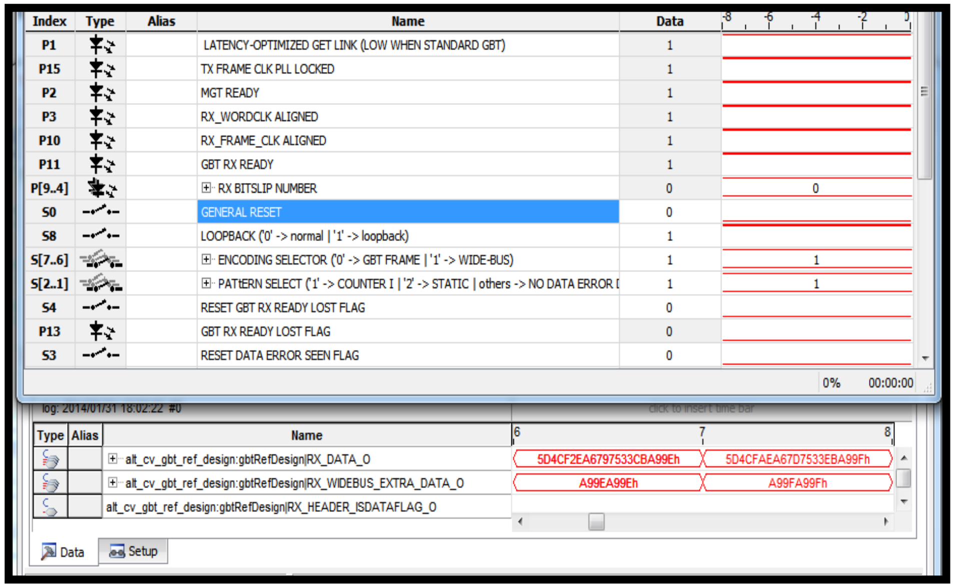
Bus/Signal	Value
LATENCY-OPTIMIZED GBT LINK (LOW WHEN STANDARD GBT)	●
TX PLL LOCKED	●
MGT READY	●
RX_WORDCLK ALIGNED (ALIGNED TO TX_WORDCLK) (LOW WHEN STANDARD GBT)	●
RX_FRAMECLK ALIGNED (ALIGNED TO TX_FRAMECLK) (LOW WHEN STANDARD GBT)	●
RX GBT READY	●
RX BITSLIP NUMBER	00
FPGA_CLKOUT ('0' -> TX_FRAMECLK | '1' -> TX_WORDCLK)	0
LOOPBACK ('0' -> NORMAL | '2' -> PMA LOOPBACK) (XILINX UG366 PAGE 124)	0
GENERAL RESET	⎍
ENCODING SELECTOR ('0' -> GBT FRAME | '1' -> WIDE-BUS)	0
PATTERN SELECT ('1' -> COUNTER | '2' -> STATIC | others -> NO DATA ERROR DETECTION)	2
RESET RX GBT READY LOST FLAG	⎍
RX GBT READY LOST FLAG	●
RESET DATA ERROR SEEN FLAG	⎍
COMMON DATA ERROR SEEN FLAG	●
WIDE-BUS EXTRA DATA ERROR SEEN FLAG	●
ENC8B10B EXTRA DATA ERROR SEEN FLAG	●
TX HEADER SELECTOR ('0' -> IDLE | '1' -> DATA)	⎍
RX HEADER IS DATA FLAG ('0' -> IDLE | '1' -> DATA)	●

Waveform - DEV:0 MyDevice0 (XC7K325T) UNIT:2 MyILA2 (ILA)

Bus/Signal	X	O	0 160 320 480 640 800 960
RX COMMON DATA	0000BABEAC1DACDCFFFFF	0000BABEAC1DACDCFFFFF	0000BABEAC1DACDCFFFFF
RX WIDEBUS EXTRA DATA	00000000	00000000	00000000
TX ENC8B10B EXTRA DATA	0	0	0
RX HEADER IS DATA FLAG	0	0	

Waveform captured Jan 21, 2014 5:03:22 PM X: 0 O: 0 Δ (X-O) : 0

Altera: SignalTap II & In-System Sources And Probes



Index	Type	Alias	Name	Data
P1			LATENCY-OPTIMIZED GET LINK (LOW WHEN STANDARD GBT)	1
P15			TX FRAME CLK PLL LOCKED	1
P2			MGT READY	1
P3			RX_WORDCLK ALIGNED	1
P10			RX_FRAME_CLK ALIGNED	1
P11			GBT RX READY	1
P[9..4]			RX BITSLIP NUMBER	0
S0			GENERAL RESET	0
S8			LOOPBACK ('0' -> normal | '1' -> loopback)	1
S[7..6]			ENCODING SELECTOR ('0' -> GBT FRAME | '1' -> WIDE-BUS)	1
S[2..1]			PATtERN SELECT ('1' -> COUNTER I | '2' -> STATIC | others -> NO DATA ERROR [	1
S4			RESET GBT RX READY LOST FLAG	0
P13			GBT RX READY LOST FLAG	0
S3			RESET DATA ERROR SEEN FLAG	0

0% 00:00:00

log: 2014/01/31 18:02:22 #0

Type	Alias	Name	6 – 7	7 – 8
		alt_cv_gbt_ref_design:gbtRefDesign|RX_DATA_O	5D4CF2EA6797533CBA99Eh	5D4CFAEA67D7533EBA99Fh
		alt_cv_gbt_ref_design:gbtRefDesign|RX_WIDEBUS_EXTRA_DATA_O	A99EA99Eh	A99FA99Fh
		alt_cv_gbt_ref_design:gbtRefDesign|RX_HEADER_ISDATAFLAG_O		

Data Setup

Our approach: "one unified core" (7 of 7)

PH-ESE-BE

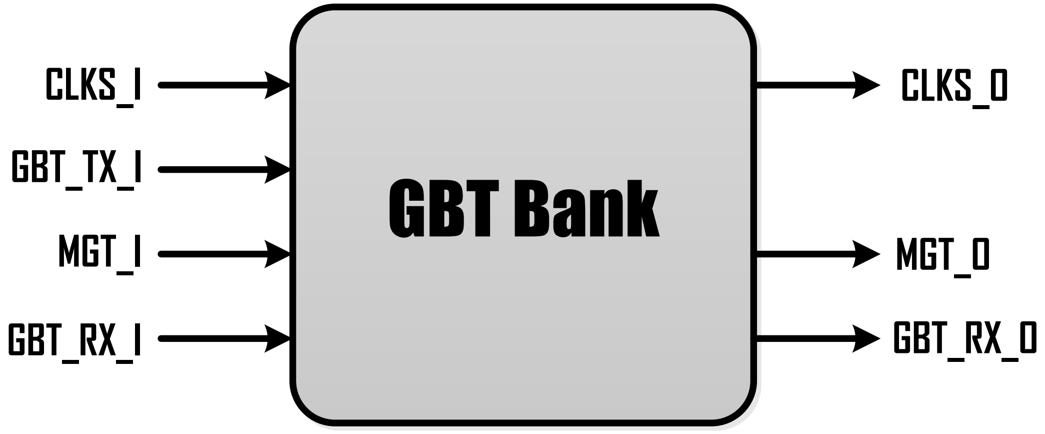
Demo

GBT-FPGA

One unified core for multiple users

Outline:

- The context: “multiple needs”
- Our approach: “one unified core”
- **The latency optimization nightmare**
- Status & Outlook



PH-ESE-BE

The latency optimization nightmare

(1 of 7)

PH-ESE-BE

Latency requirements

- **Standard (STD)**
  - Data Readout (DAQ)
- **Low and Deterministic latency (LATOPT)**
  - Front-End Control (FEC)
  - Time, Trigger & Control (TTC)

The latency optimization nightmare

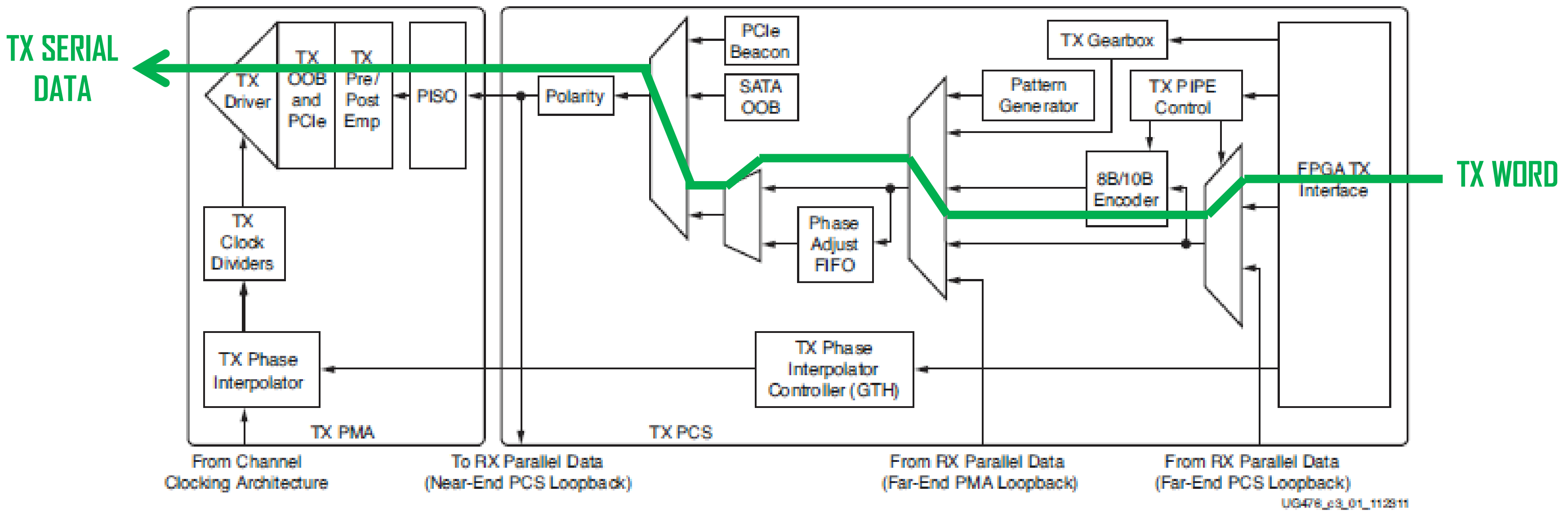
(1 of 7)

PH-ESE-BE

Low latency

- How do we achieve LOW latency?
  - Avoiding unnecessary components in critical paths (e.g. unused internal blocks of MGT)

Example of component bypassing in GTX transceiver



The latency optimization nightmare

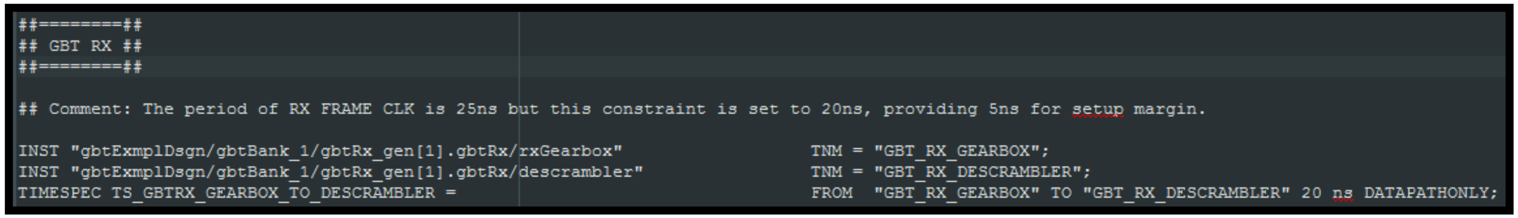
(1 of 7)

PH-ESE-BE

Low latency

- **How do we achieve LOW latency?**
  - Avoiding unnecessary components in critical paths (e.g. unused internal blocks of MGT)
  - Using as less registers as possible
    - Need of constraints for achieving timing closure:
      - *Timing constraints for critical paths*

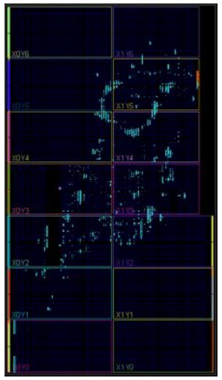
Example of timing constraint



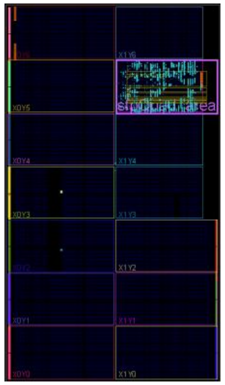
```
##=========##
## GBT RX ##
##=========##

## Comment: The period of RX FRAME CLK is 25ns but this constraint is set to 20ns, providing 5ns for setup margin.

INST "gbtExmplDsgn/gbtBank_1/gbtRx_gen[1].gbtRx/rxGearbox"                TNM = "GBT_RX_GEARBOX";
INST "gbtExmplDsgn/gbtBank_1/gbtRx_gen[1].gbtRx/descrambler"              TNM = "GBT_RX_DESCRAMBLER";
TIMESPEC TS_GBTRX_GEARBOX_TO_DESCRAMBLER =                                FROM  "GBT_RX_GEARBOX" TO "GBT_RX_DESCRAMBLER" 20 ns DATAPATHONLY;
```

      - *Floorplaning constrains to concentrate the logic*

Without logic floorplanning



With logic floorplanning



The latency optimization nightmare

(2 of 7)

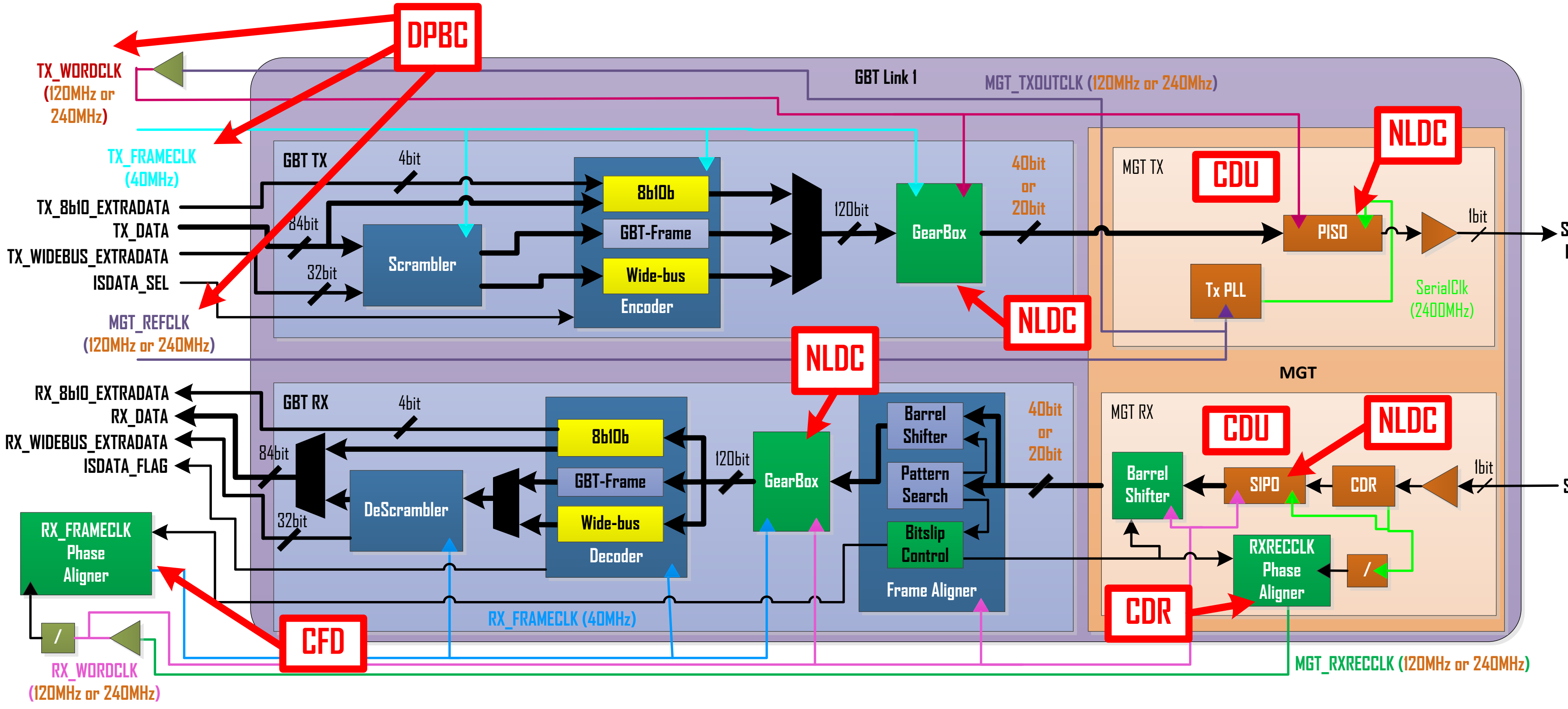
PH-ESE-BE

Deterministic Latency

- How do we achieve DETERMINISTIC latency?
  - Properly managing potential uncertainty points:

Non Latency Deterministic Component (NLDC) Deterministic Phase Between Clocks (DPBC)
Clock Domain Unification (CDU) Clock & Data Recovery (CDR) Clock Frequency Division (CFD)

- Where do we have those potential uncertainty points in a typical GBT Bank-based system?

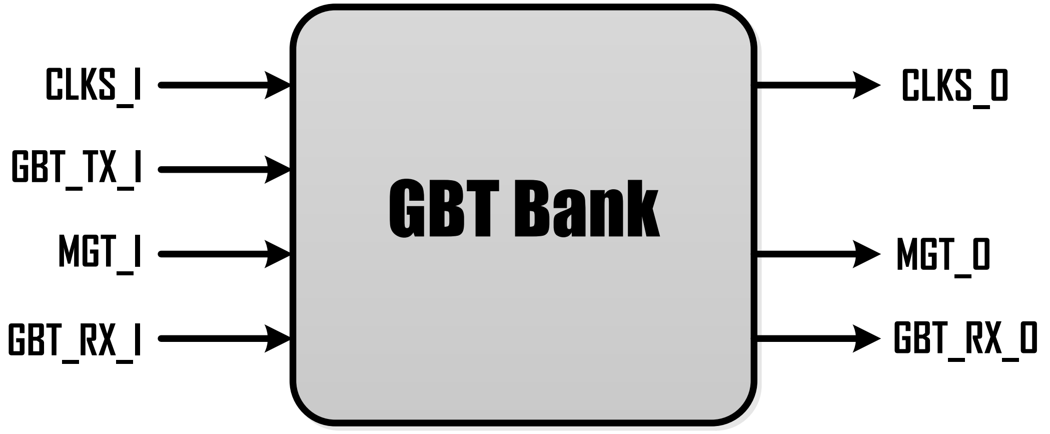


GBT-FPGA

One unified core for multiple users

Outline:

- The context: “multiple needs”
- Our approach: “one unified core”
- The latency optimization nightmare
- **Status & Outlook**



PH-ESE-BE

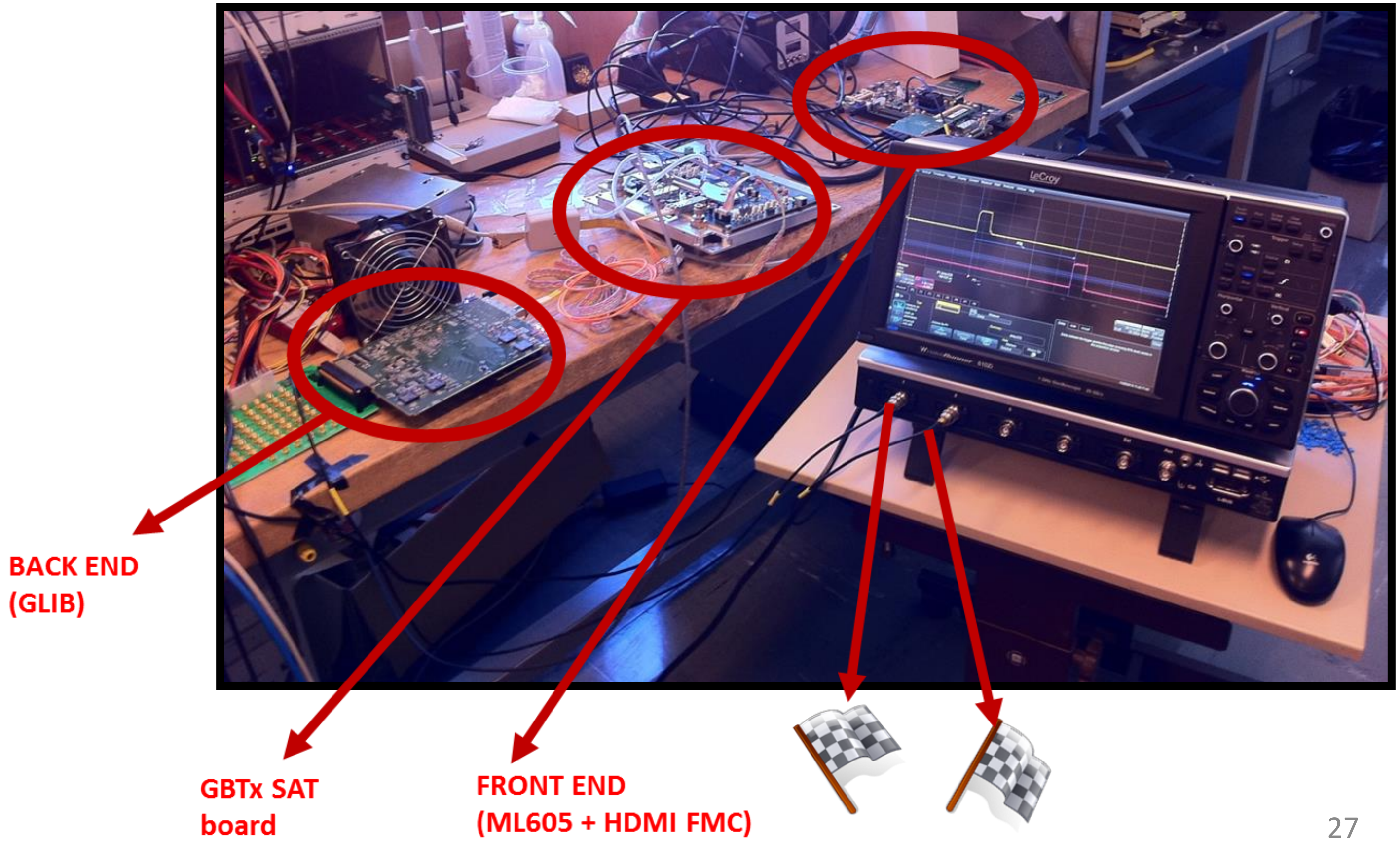
Status & Outlook

PH-ESE-BE

Status

- **Figures**
  - Latency of the LATOPT GBT Link (Virtex 6):
    - **GBT Link TX:** 29.2ns (GBT TX) + 18.7ns (GTX TX) = **47.9ns**
    - **GBT Link RX:** 54.2ns (GBT RX) + 28.2ns (GTX RX) = **82.4ns**

GBT Link latency measurement setup



Status & Outlook

PH-ESE-BE

Status

- Figures
  - Resources utilization of one GBT Bank instantiating one GBT Link:

Xilinx (Kintex7: XC7K325T)		
Resources	**STD (%)**	**LATOPT (%)**
LUT	2658 (1.30)	2776 (1.36)
FD_LD	817 (0.20)	969 (0.24)
BMEM	10 (1.12)	0 (0.00)
GTX	1 (6.25)	1 (6.25)

Altera (Cyclone V: 5CGTFD9E5F35C7N)		
Resources	**STD (%)**	**LATOPT (%)**
ALM	1674 (1.47)	1827 (1.61)
Register	1100 (0.24)	1475 (0.32)
Mem (M10K)	10 (0.81)	2 (0.16)
GT	1 (8.33)	1 (8.33)

Issues usually come from Clocking Resources!!!

Status & Outlook

PH-ESE-BE

Status

- **GBT-FPGA development kit**
  - Includes:
    - Sources
    - Example Designs
    - Documentation
    - TCL scripts
  - First release: March 2014
  - Available for:

Altera	
FPGA	**FPGA-based Board**
Cyclone V	Cyclone V GT Devkit
Stratix V	AMC40

Xilinx	
FPGA	**FPGA-based Board**
Virtex 6	GLIB, ML605
Kintex 7	KC705
Virtex 7	VC705

  - To do:
    - Documentation & TCL scripts
    - Finalize 8b10b encoding
    - Implement new versions (Artix 7, etc.)
    - Perform different studies (Wide-bus latency, Clock phase drift with temperature, etc.)

Thank you

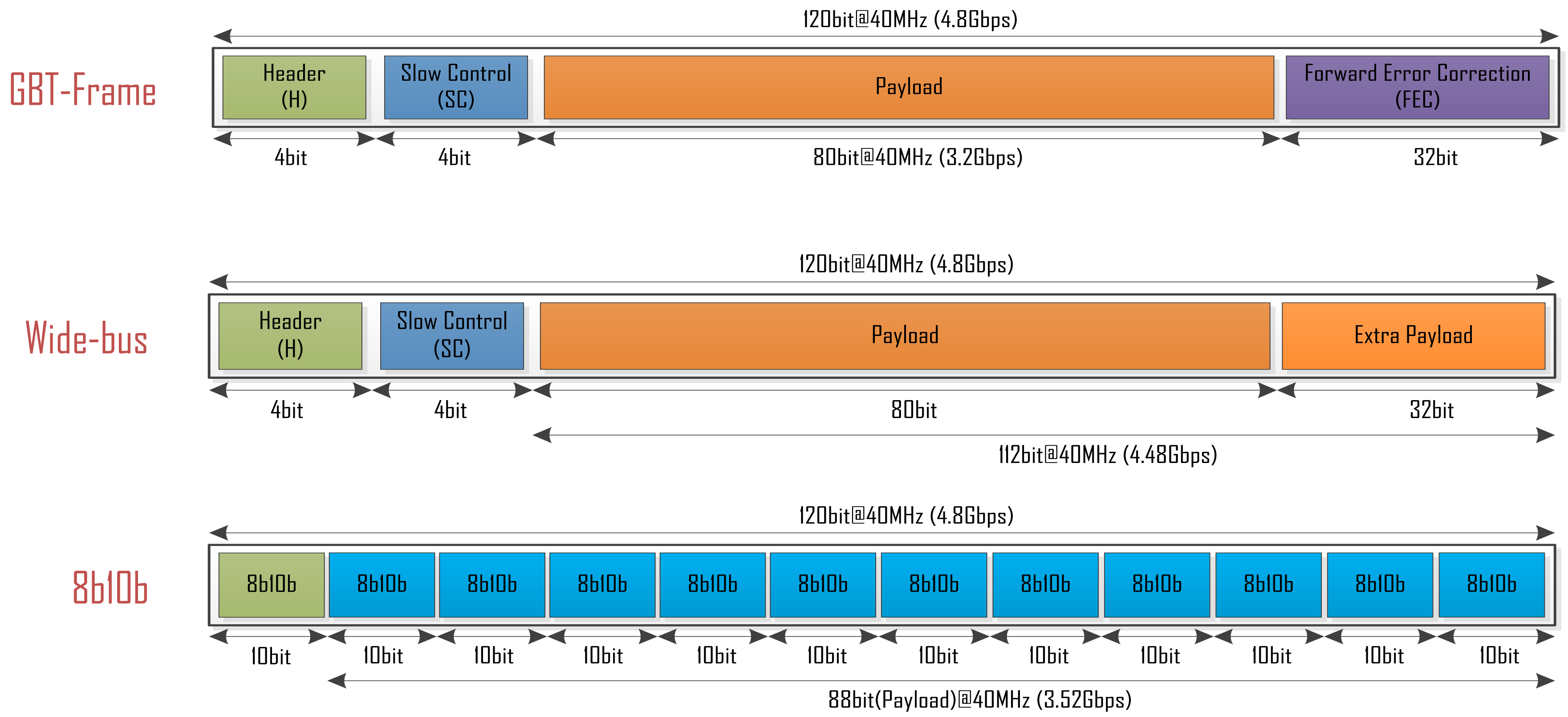
manoel.barros.marin@cern.ch

The context: "multiple needs" (3 of 5)

PH-ESE-BE

Multiple Configurations

- Encoding



Main page

The latency optimization nightmare

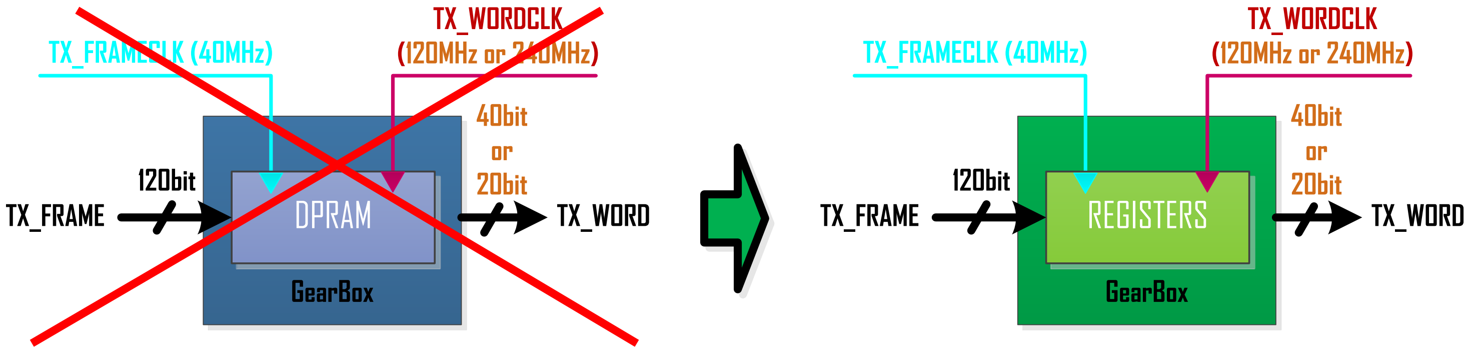
(3 of 7)

PH-ESE-BE

Deterministic Latency

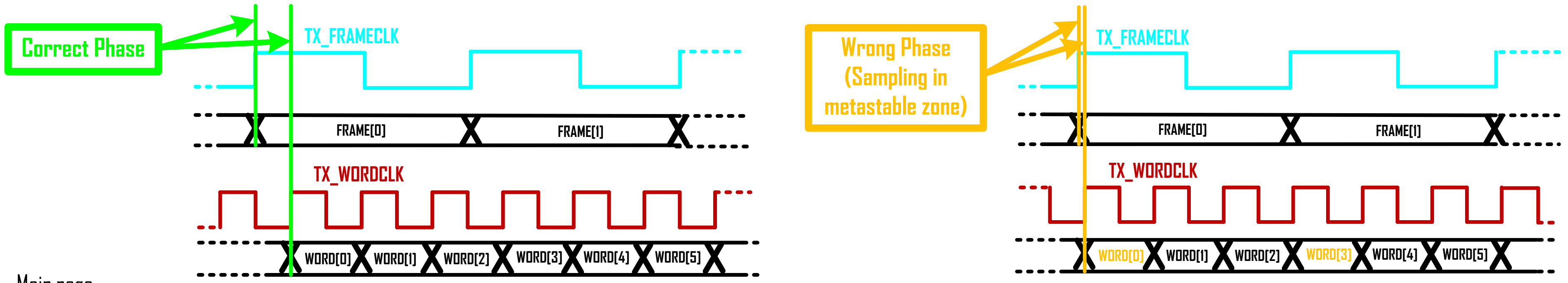
- Non Latency Deterministic Component (NLDC)
  - Registers instead of RAMs (or FIFOs) ensures latency determinism

Example of NLDC (TX GearBox)



  - But registers requires correct phase relationship between clocks

Example of sampling in latency deterministic TX GearBox (with registers)



Main page

The latency optimization nightmare

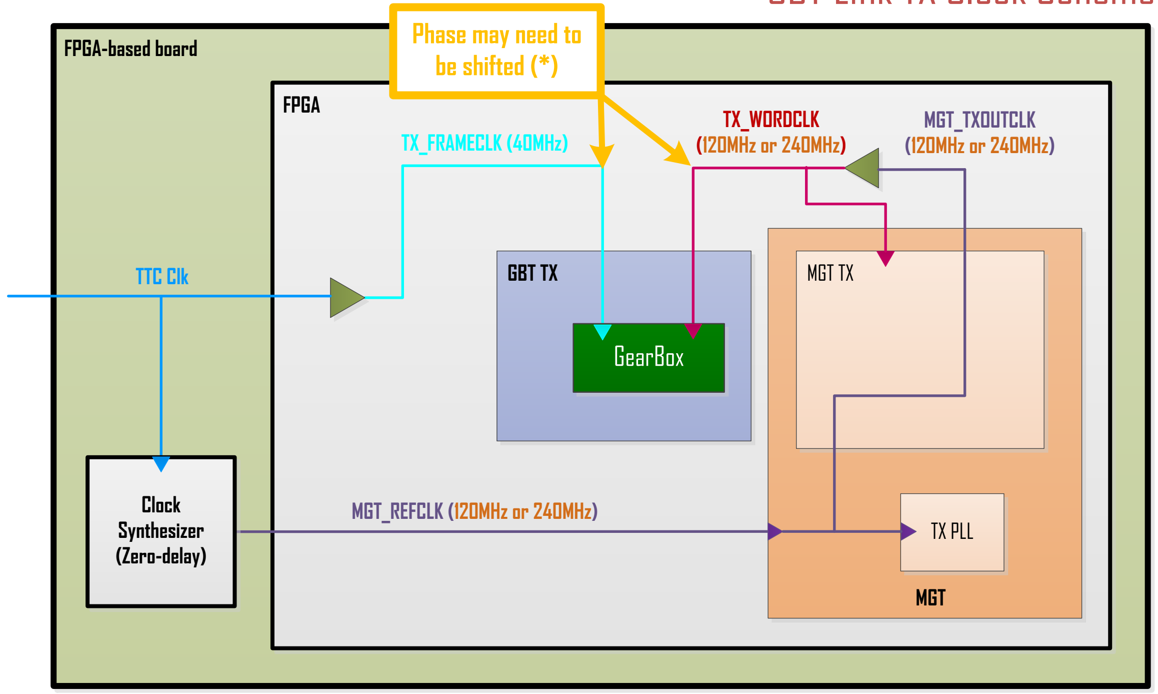
(4 of 7)

PH-ESE-BE

Deterministic Latency

- **Deterministic Phase Between Clocks (DPBC)**
  - All TX clocks must have deterministic phase relationship:
    - TX_FRAMECLK may be directly derived from TTC Clk
    - Use of zero-delay clock synthesizers for derived clock generation (e.g. MGT_REFCLK from TTC Clk)
    - MGT_TXOUTCLK directly derived from MGT_REFCLK
  - Need to find correct sampling point in GBT TX Gearbox (*)

GBT Link TX Clock Scheme


FPGA-based board
Phase may need to be shifted (*)
FPGA
TX_FRAMECLK (40MHz)
TX_WORDCLK (120MHz or 240MHz)
MGT_TXOUTCLK (120MHz or 240MHz)
TTC Clk
GBT TX
GearBox
MGT TX
Clock Synthesizer (Zero-delay)
MGT_REFCLK (120MHz or 240MHz)
TX PLL
MGT

The latency optimization nightmare

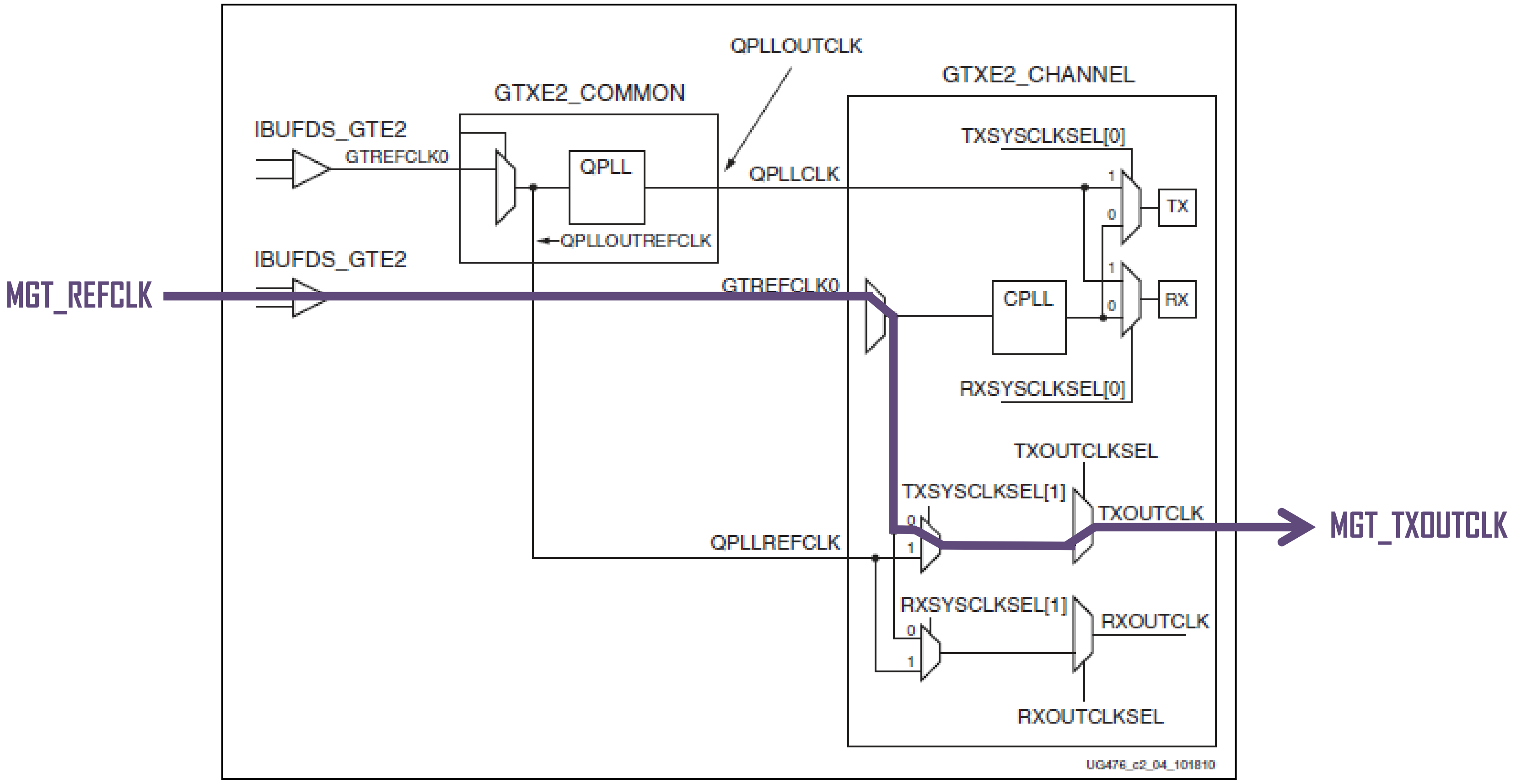
(4 of 7)

PH-ESE-BE

Deterministic Latency

- **Deterministic Phase Between Clocks (DPBC)**
    - MGT_TXOUTCLK from MGT_REFCLK in Xilinx MGT (GTX)

7 series GTX TX Clock Scheme



The latency optimization nightmare

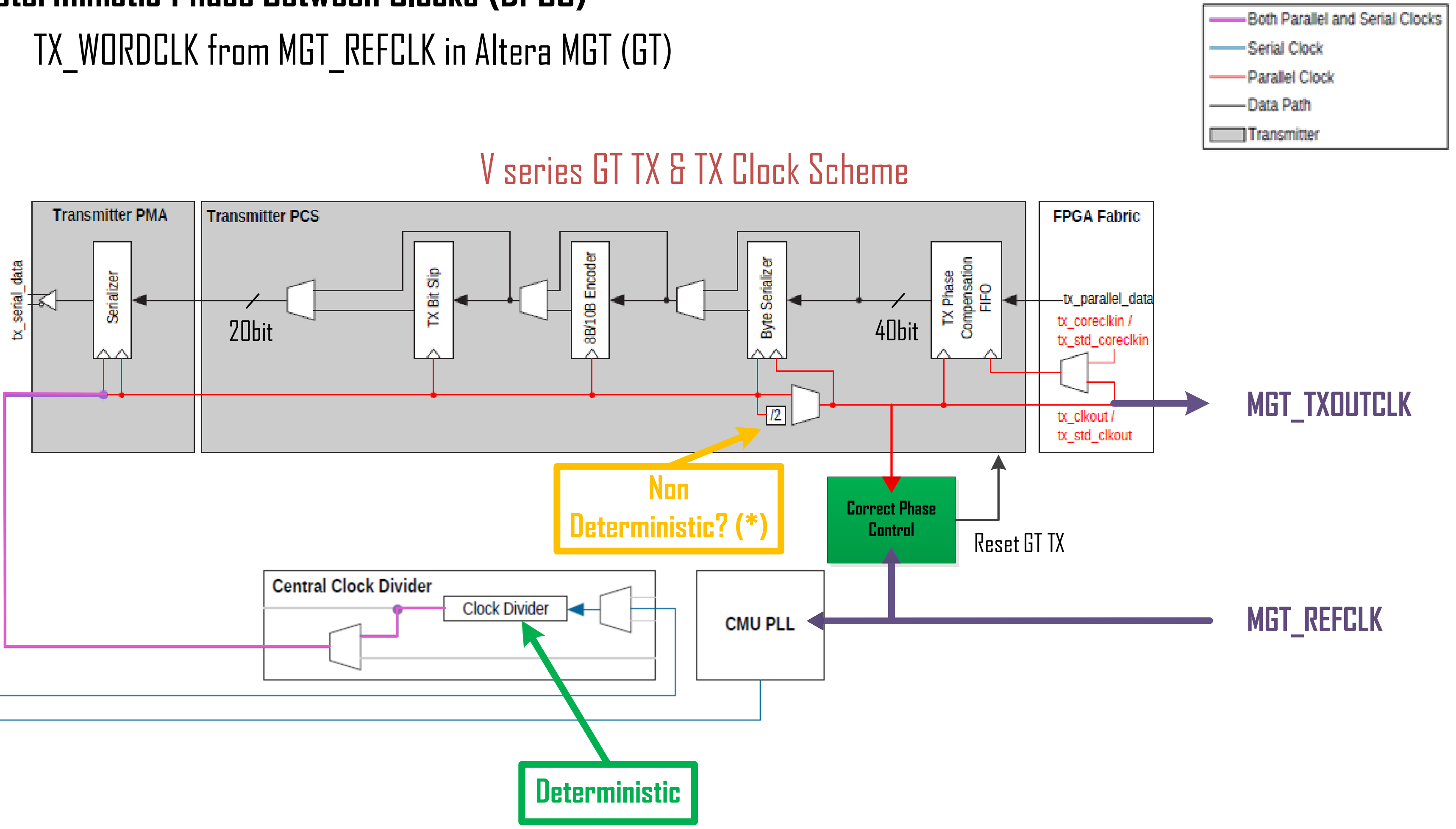
(4 of 7)

PH-ESE-BE

Deterministic Latency

- **Deterministic Phase Between Clocks (DPBC)**
  - TX_WORDCLK from MGT_REFCLK in Altera MGT (GT)

V series GT TX & TX Clock Scheme



(*) We are in contact with Altera to clarify this issue

Main page

The latency optimization nightmare

(5 of 7)

PH-ESE-BE

Latency optimization

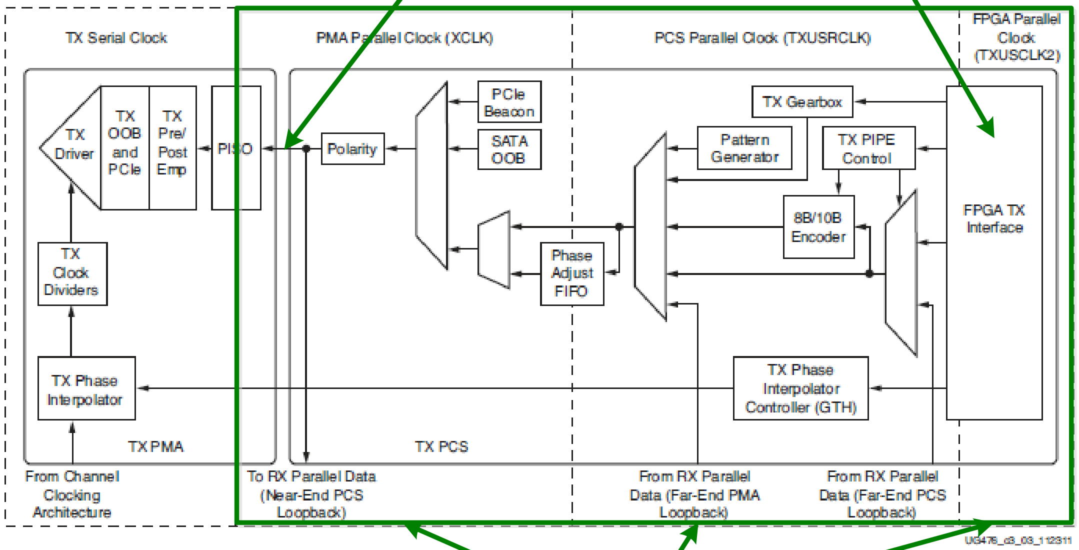
Lane Rate (Gbps) = Bus Width (bit) x Frequency (GHz)

GBT Lane Rate = 4.8 Gbps

- **Clock Domain Unification (CDU)**
  - Possible with Xilinx MGT (GTX)

Example of Xilinx 7 series GTX TX CDU

Internal Bus Width = External Bus Width (40 or 20 bits)


TX Serial Clock
PMA Parallel Clock (XCLK)
PCS Parallel Clock (TXUSRCLK)
FPGA Parallel Clock (TXUSCLK2)
TX Driver
TX OOB and PCIe
TX Pre/ Post Emp
PISO
Polarity
PCIe Beacon
SATA OOB
Phase Adjust FIFO
TX Gearbox
Pattern Generator
TX PIPE Control
8B/10B Encoder
FPGA TX Interface
TX Clock Dividers
TX Phase Interpolator
TX Phase Interpolator Controller (GTH)
TX PMA
TX PCS
From Channel Clocking Architecture
To RX Parallel Data (Near-End PCS Loopback)
From RX Parallel Data (Far-End PMA Loopback)
From RX Parallel Data (Far-End PCS Loopback)
UG476_c3_03_112311

MGT_REFCLK = XCLK = TXUSRCLK = TXUSRCLK2 = (120Mhz or 240MHz)

The latency optimization nightmare

(5 of 7)

PH-ESE-BE

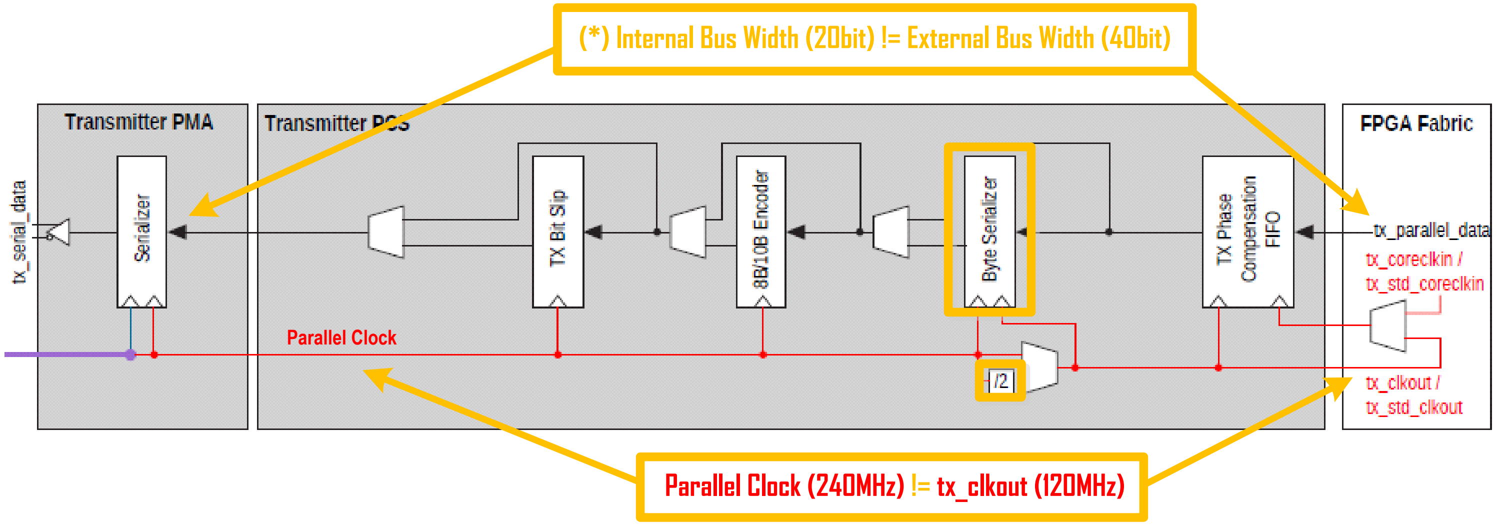
Latency optimization

- **Clock Domain Unification (CDU)**
  - Possible with Xilinx MGT (GTX)
  - NOT possible with Altera MGT (GT) at GBT lane rate

Lane Rate (Gbps) = Bus Width (bit) x Frequency (GHz)

GBT Lane Rate = 4.8 Gbps

Example of Altera V series GT TX CDU



(*) Internal Bus Width (20bit) != External Bus Width (40bit)
Transmitter PMA
Transmitter PCS
FPGA Fabric
tx_serial_data
Serializer
TX Bit Slip
8B/10B Encoder
Byte Serializer
TX Phase Compensation FIFO
tx_parallel_data
tx_coreclkin / tx_std_coreclkin
Parallel Clock
/2
tx_clkout / tx_std_clkout
Parallel Clock (240MHz) != tx_clkout (120MHz)

Main page

The latency optimization nightmare

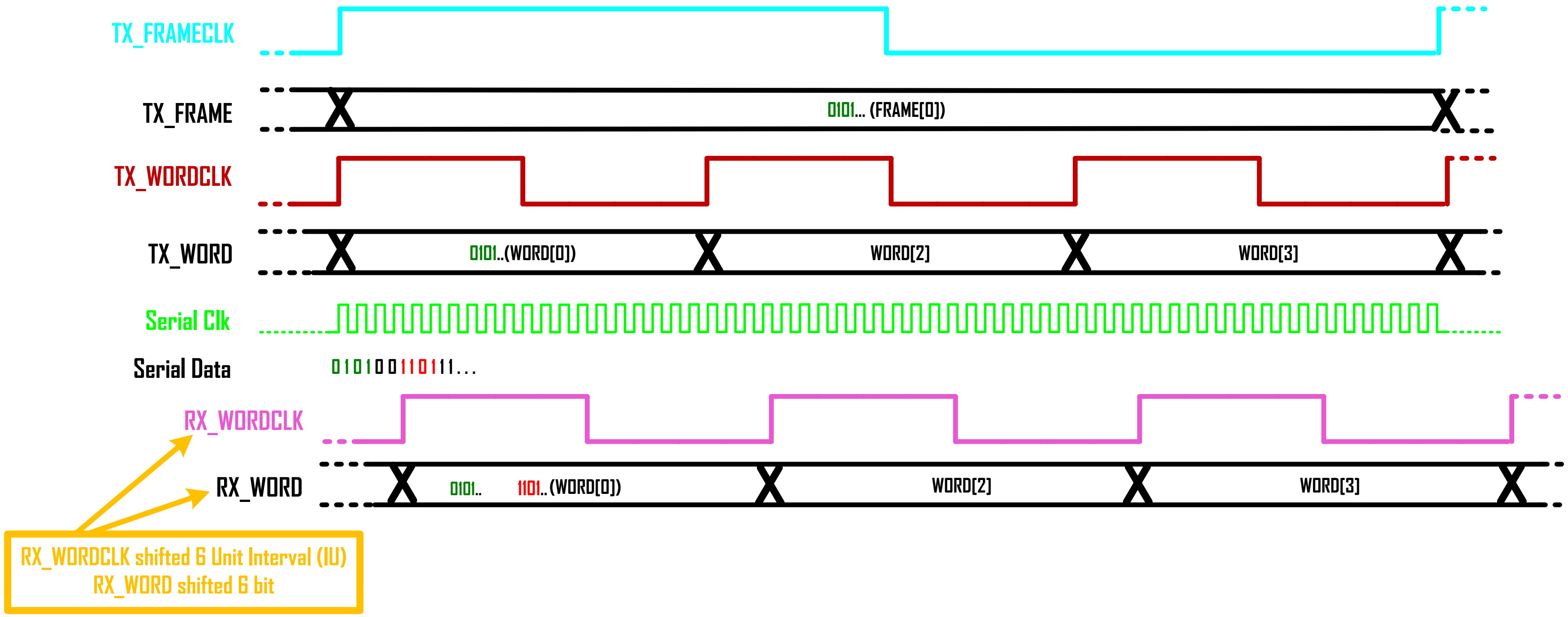
(6 of 7)

PH-ESE-BE

Latency optimization

- Clock & Data Recovery (CDR)

EXAMPLE
- Frame Header = 0101
- Frame Width = 120bit
- Word Width = 40bit
- TX_FRAMECLK = 40MHz
- TX_WORDCLK = 120MHz
- Serial Clk = 2.4GHz (DDR)
- RX_WORDCLK = 120MHz = MGT_RXRECCLK
- Max. bitslips = (2.4GHz(DDR)/120MHz)-1 = [((2.4GHz x 2)/120MHz)]-1 = 39
- Bitslips = 6



The latency optimization nightmare

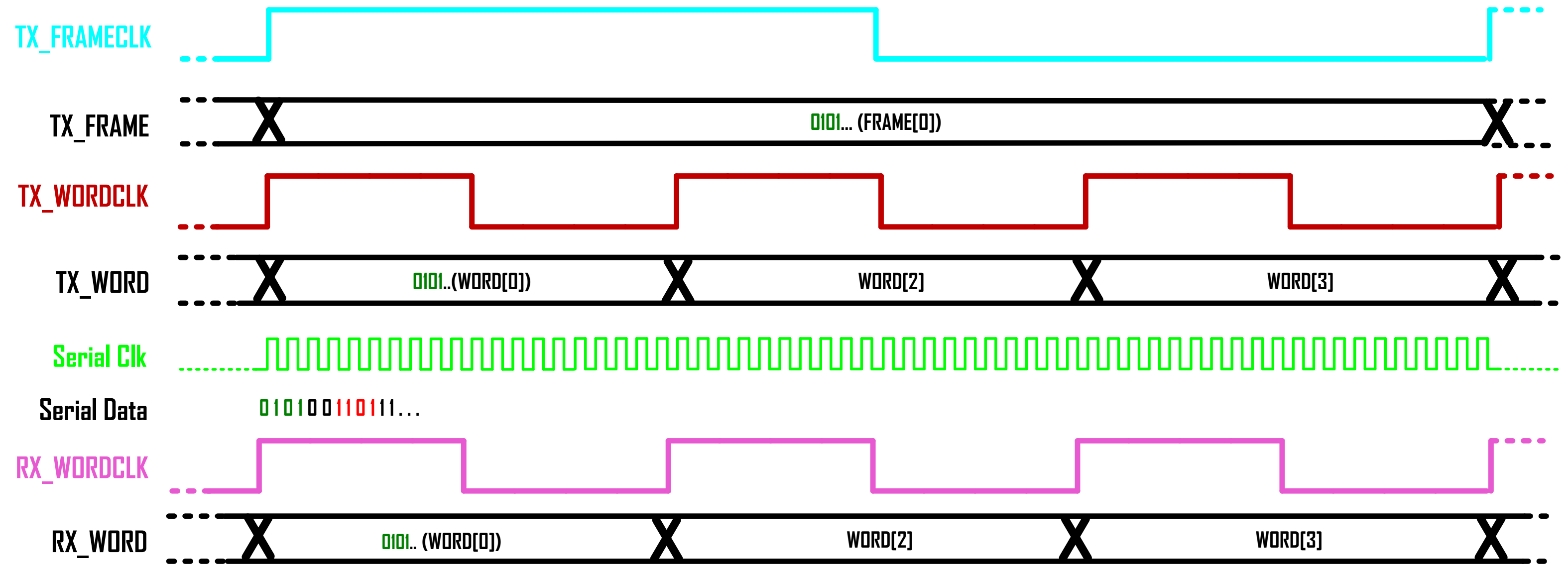
(6 of 7)

PH-ESE-BE

Latency optimization

- Clock & Data Recovery (CDR)

EXAMPLE
- Frame Header = 0101
- Frame Width = 120bit
- Word Width = 40bit
- TX_FRAMECLK = 40MHz
- TX_WORDCLK = 120MHz
- Serial Clk = 2.4GHz (DDR)
- RX_WORDCLK = 120MHz = MGT_RXRECCLK
- Max. bitslips = (2.4GHz(DDR)/120MHz)-1 = [((2.4GHz x 2)/120MHz)]-1 = 39
- Bitslips = 6



Main page

The latency optimization nightmare

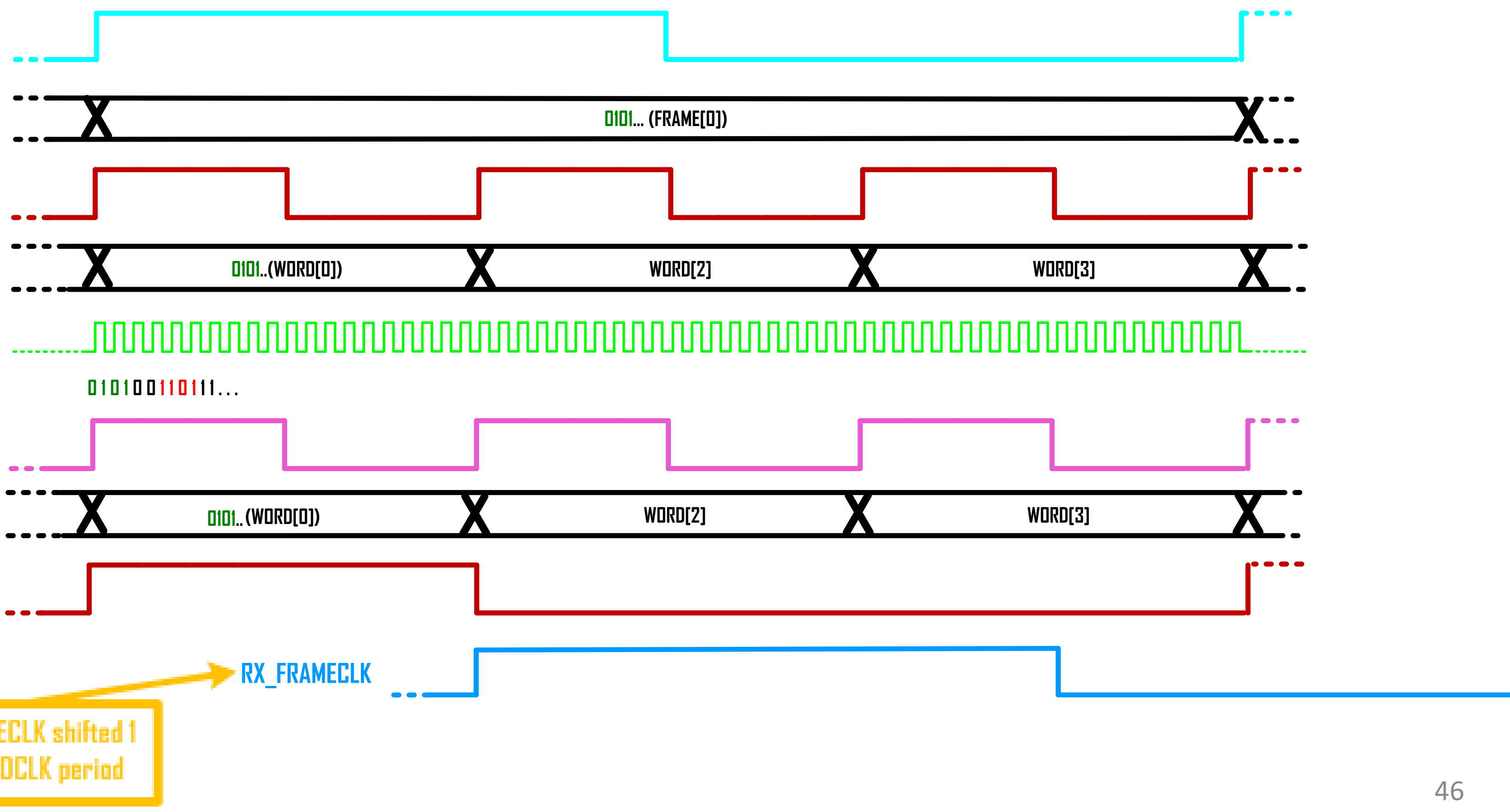
(7 of 7)

PH-ESE-BE

Latency optimization

- Clock Frequency Division (CFD)

EXAMPLE
- Frame Header = 0101
- Frame Width = 120bit
- Word Width = 40bit
- RX_WORDCLK = 120MHz = MGT_RECCLK
- RX_FRAMECLK = 40MHz
- Max. Clk Slips = (120MHz/40MHz)-1 = 2
- Clk Slips = 1



The latency optimization nightmare

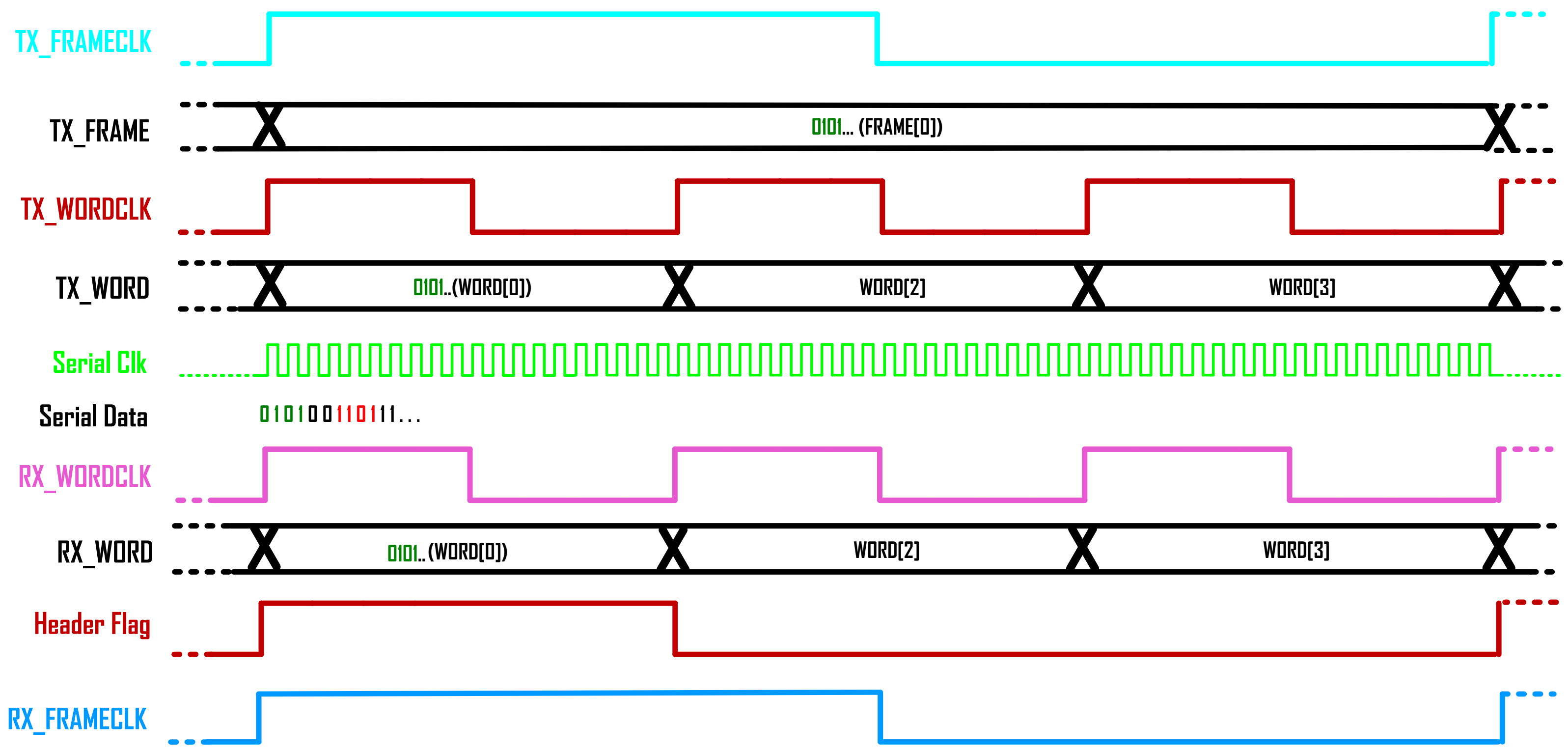
(7 of 7)

PH-ESE-BE

Latency optimization

- Clock Frequency Division (CFD)

EXAMPLE
- Frame Header = 0101
- Frame Width = 120bit
- Word Width = 40bit
- RX_WORDCLK = 120MHz = MGT_RECCLK
- RX_FRAMECLK = 40MHz
- Max. Clk Slips = (120MHz/40MHz)-1 = 2
- Clk Slips = 1



Main page