CT-PPS Motherboard registers library

Generated by Doxygen 1.8.10

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 TDCStatus::ErrorType Struct Reference

Type of error encountered by the HPTDC.

`#include <TDCStatus.h>`

Public Member Functions

- bool ParityError () const

  Error related on the parity of any register/buffer.
- bool MeasurementError () const

  Error related to the Vernier or Coarse measurement.
- bool GlobalError () const

  Has any error occured?

Public Attributes

- bool **Vernier**
- bool **Coarse**
- bool **ChannelSelect**
- bool **L1BufferParity**
- bool **TriggerFIFOParity**
- bool **TriggerMatchingState**
- bool **ReadoutFIFOParity**
- bool **ReadoutState**
- bool **SetupParity**
- bool **ControlParity**
- bool **JTAGInstruction**

3.1.1 Detailed Description

Type of error encountered by the HPTDC.

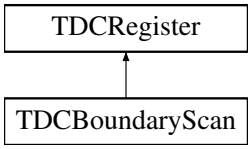
The documentation for this struct was generated from the following file:

- include/TDCStatus.h

3.2 TDCBoundaryScan Class Reference

`#include <TDCBoundaryScan.h>`

Inheritance diagram for TDCBoundaryScan:



Public Member Functions

- **TDCBoundaryScan** (const TDCBoundaryScan &bs)
- bool **IsTokenOut** () const
- bool **IsStrobeOut** () const
- bool **IsSerialOut** () const
- bool **IsTest** () const
- bool **IsError** () const
- bool **IsDataReady** () const
- bool **IsParallelEnabled** () const
- bool **HasParallelDataOut** (unsigned short channel_id) const
- bool **IsEncodedControl** () const
- bool **IsTrigger** () const
- bool **HasTrigger** () const
- bool **HasEventReset** () const
- bool **HasBunchReset** () const
- bool **IsGettingData** () const
- bool **IsSerialBypassIn** () const
- bool **IsSerialIn** () const
- bool **IsTokenBypassIn** () const
- bool **IsTokenIn** () const
- bool **IsReset** () const
- bool **HasAuxiliaryClock** () const
- bool **HasClock** () const
- bool **HasHit** (unsigned short channel_id) const
- void SetConstantValues ()

  Set all hardcoded values to this register.
- void Dump () const

  Printout all useful values of this status register into an output stream.

Additional Inherited Members

3.2.1 Detailed Description

Author

Laurent Forthomme `laurent.forthomme@cern.ch`

Date

24 Apr 2015
May 2016

The documentation for this class was generated from the following file:

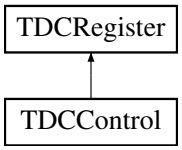
- include/TDCBoundaryScan.h

3.3 TDCControl Class Reference

Control word to be sent to the HPTDC chip.

`#include <TDCControl.h>`

Inheritance diagram for TDCControl:



Public Types

- enum **EnablePattern** { **OutputEnabled** =0x5, **OutputDisabled** =0x4 }

Public Member Functions

- **TDCControl** (const TDCControl &c)
- **TDCControl** (const std::vector< uint8_t > &words)
- void **SetEnablePattern** (const EnablePattern &ep=OutputEnabled)
- EnablePattern **GetEnablePattern** () const
- void **SetGlobalReset** (const bool gr=true)
- bool **GetGlobalReset** () const
- void **SetDLLReset** (const bool dr=true)
- bool **GetDLLReset** () const
- void **SetPLLReset** (const bool pr=true)
- bool **GetPLLReset** () const
- void **EnableChannel** (unsigned int id)
- void **EnableAllChannels** ()
- void **DisableChannel** (unsigned int id)
- void **DisableAllChannels** ()
- void Dump (int verb=1, std::ostream &os=std::cout) const

  Printout all useful values of this control register into an output stream.
- void SetConstantValues ()

Additional Inherited Members

3.3.1 Detailed Description

Control word to be sent to the HPTDC chip.

Object handling the control word provided by/to the HPTDC chip

Author

Laurent Forthomme laurent.forthomme@cern.ch
Lara Lloret lara@cern.ch

Date

24 Apr 2015

3.3.2 Member Function Documentation

3.3.2.1 void TDCControl::SetConstantValues ( ) [virtual]

Ensure that the critical constant values are properly set in the register word

Implements TDCRegister.

The documentation for this class was generated from the following file:

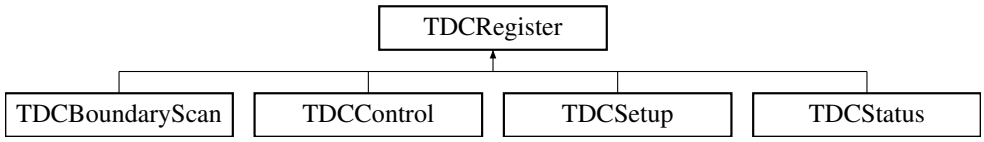
- include/TDCControl.h

3.4 TDCRegister Class Reference

General register object to interact with a HPTDC chip.

`#include <TDCRegister.h>`

Inheritance diagram for TDCRegister:



Public Types

- typedef uint16_t bit
  LSB index.
- typedef uint32_t word_t
  Unit of the TDC register word to be successfully contained on any machine.

Public Member Functions

- TDCRegister (const unsigned int size)
  Initialise an empty register.
- TDCRegister (const unsigned int size, const TDCRegister &r)
  Initialise and fill a register.
- TDCRegister (const unsigned int size, const std::vector< uint8_t > &words, bool reversed=false)
  Initialise and fill a register.
- virtual ~TDCRegister ()
  Destroy the register and its content.
- TDCRegister & operator= (const TDCRegister &r)
  Assign values from another register to this one.

- void SetWord (const unsigned int i, const word_t word)

  Set one bit(s) subset in the register word.
- word_t GetWord (const unsigned int i) const

  Retrieve one subset from the register word.
- word_t * GetWords () const

  Retrieve the whole array of sub-words composing this register.
- std::vector< uint8_t > GetBytesVector () const

  Retrieve a vector of 8-bit words composing this register.
- uint8_t GetNumWords () const

  Number of words in the register.
- void DumpRegister (unsigned short verb=1, std::ostream &os=std::cout, const bit max_bits=-1) const

  Printout all useful information handled by the register.
- virtual void SetConstantValues ()=0

Protected Member Functions

- void SetBits (uint16_t lsb, uint16_t word, uint8_t size)

  Set bits in the register word.
- uint16_t GetBits (uint16_t lsb, uint8_t size) const

  Extract bits from the register word.
- void Clear ()

  Set all bits in this register to '0'.

Protected Attributes

- word_t * fWord

  Pointer to this register's word.
- unsigned int fNumWords
- unsigned int fWordSize

  Number of bits in this register.

3.4.1 Detailed Description

General register object to interact with a HPTDC chip.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

24 Apr 2015

3.4.2 Member Function Documentation

3.4.2.1 uint16_t TDCRegister::GetBits (uint16_t *lsb*, uint8_t *size*) const [protected]

Extract bits from the register word.

Extract a fixed amount of bits from the full register word

Parameters

in	*lsb*	Least significant bit of the word to retrieve
in	*size*	Size of the word to retrieve

3.4.2.2 uint8_t TDCRegister::GetNumWords ( ) const [inline]

Number of words in the register.

Return the number of words making up the full register word.

3.4.2.3 void TDCRegister::SetBits (uint16_t *lsb*, uint16_t *word*, uint8_t *size*) [protected]

Set bits in the register word.

Set a fixed amount of bits in the full register word

Parameters

in	*lsb*	Least significant bit of the word to set
in	*word*	Word to set
in	*size*	Size of the word to set

3.4.2.4 virtual void TDCRegister::SetConstantValues ( ) [pure virtual]

Ensure that the critical constant values are properly set in the register word

Implemented in TDCSetup, TDCBoundaryScan, TDCControl, and TDCStatus.

3.4.3 Member Data Documentation

3.4.3.1 unsigned int TDCRegister::fNumWords [protected]

Number of words to fit the *fWordSize* bits of this register to this object

The documentation for this class was generated from the following file:

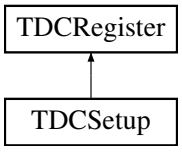
- include/TDCRegister.h

3.5 TDCSetup Class Reference

Setup word to be sent to the HPTDC chip.

`#include <TDCSetup.h>`

Inheritance diagram for TDCSetup:



Public Types

- enum **EdgeResolution** {
  E_100ps =0, **E_200ps**, **E_400ps**, **E_800ps**,
  E_1p6ns, **E_3p12ns**, **E_6p25ns**, **E_12p5ns** }
- enum **DeadTime** { **DT_5ns** =0, **DT_10ns**, **DT_30ns**, **DT_100ns** }
- enum **WidthResolution** {
  W_100ps =0, **W_200ps**, **W_400ps**, **W_800ps**,
  W_1p6ns, **W_3p2ns**, **W_6p25ns**, **W_12p5ns**,
  W_25ns, **W_50ns**, **W_100ns**, **W_200ns**,
  W_400ns, **W_800ns** }
- enum **EnabledError** {
  VernierError =0x1, **CoarseError** =0x2, **ChannelSelectError** =0x4, **L1BufferParityError** =0x8,
  TriggerFIFOParityError =0x10, **TriggerMatchingError** =0x20, **ReadoutFIFOParityError** =0x40,
  ReadoutStateError =0x80,
  SetupParityError =0x100, **ControlParityError** =0x200, **JTAGInstructionParityError** =0x400 }
- enum **DLLSpeedMode** { **DLL_40MHz** =0x0, **DLL_160MHz** =0x1, **DLL_320MHz** =0x2, **DLL_Illegal** =0x3 }
- enum **SerialClockSource** { **Serial_pll_clock_80** =0x0, **Serial_pll_clock_160** =0x1, **Serial_pll_clock_40** =0x2, **Serial_aux_clock** =0x3 }
- enum **IOClockSource** { **IO_clock_40** =0x0, **IO_pll_clock_80** =0x1, **IO_pll_clock_160** =0x2, **IO_aux_clock** =0x3 }
- enum **CoreClockSource** { **Core_clock_40** =0x0, **Core_pll_clock_80** =0x1, **Core_pll_clock_160** =0x2, **Core_aux_clock** =0x3 }
- enum **DLLClockSource** {
  DLL_clock_40 =0x0, **DLL_pll_clock_40** =0x1, **DLL_pll_clock_160** =0x2, **DLL_pll_clock_320** =0x3,
  DLL_aux_clock =0x4 }
- enum **ReadoutSpeed** { **RO_Fixed** =0x0, **RO_pll_80Mbits_s** =0x1 }
- enum **SerialStrobeType** { **SS_NoStrobe** =0x0, **SS_DSStrobe** =0x1, **SS_LeadingTrailingStrobe** =0x2, **SS_LeadingEdge** =0x3 }
- enum **ReadoutSingleCycleSpeed** {
  RSC_40Mbits_s =0x0, **RSC_20Mbits_s** =0x1, **RSC_10Mbits_s** =0x2, **RSC_5Mbits_s** =0x3,
  RSC_2p5Mbits_s =0x4, **RSC_1p25Mbits_s** =0x5, **RSC_625kbits_s** =0x6, **RSC_312p5kbits_s** =0x7 }

Public Member Functions

- **TDCSetup** (const TDCSetup &c)
- **TDCSetup** (const std::vector< uint8_t > &v, bool reverse=false)
- void SetEnableErrorMark (const bool em)

  Mark events with error if global error signal is set.
- bool **GetEnableErrorMark** () const
- void SetEnableErrorBypass (const bool eb)

  Bypass TDC chip if global error signal is set.
- bool **GetEnableErrorBypass** () const
- void SetEnableError (const uint16_t &err)

  Enable internal error types for generation of global error signals.
- uint16_t **GetEnableError** () const
- void SetEnableSerial (const bool es)

  Enable of serial read-out (otherwise parallel read-out)
- bool **GetEnableSerial** () const
- void SetEnableJTAGReadout (const bool jr)

  Enable of read-out via JTAG.
- bool **GetEnableJTAGReadout** () const
- void SetReadoutFIFOSize (int rfs)

  Effective size of readout FIFO.

- int **GetReadoutFIFOSize** () const
- void SetRejectCountOffset (uint16_t rco)

  Set the offset in reject counter (defines reject latency together with coarse count offset)
- uint16_t GetRejectCountOffset () const

  Extract the offset in reject counter.
- void SetSearchWindow (uint16_t sw)

  Set the search window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)
- uint16_t GetSearchWindow () const

  Extract the search window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)
- void SetMatchWindow (uint16_t mw)

  Set the matching window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)
- uint16_t GetMatchWindow () const

  Extract the matching window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)
- void **SetEdgeResolution** (const EdgeResolution r)
- EdgeResolution **GetEdgeResolution** () const
- void SetMaxEventSize (int sz=-1)

  Set the maximum number of hits per event.
- uint8_t GetMaxEventSize () const

  Extract the maximum number of hits per event.
- void SetRejectFIFOFull (const bool rej=true)

  Reject hits when readout FIFO full.
- bool GetRejectFIFOFull () const

  Are hits rejected when readout FIFO is full?
- void SetEnableReadoutOccupancy (const bool ro=true)

  Enable the readout of buffer occupancies for each event (for debugging purposes)
- bool **GetEnableReadoutOccupancy** () const
- void SetEnableReadoutSeparator (const bool ro=true)

  Enable the readout of separators for each event (for debugging purposes, valid if readout of occupancies is enabled)
- bool **GetEnableReadoutSeparator** () const
- void SetEventCountOffset (uint16_t eco)

  Set offset for the event counter.
- void SetTriggerCountOffset (uint16_t tco)

  Set offset for the trigger time tag counter to set effective trigger latency.
- uint16_t GetTriggerCountOffset () const

  Extract trigger time tag count offset.
- void SetChannelOffset (int channel, uint16_t offset)

  Set the time offset for one single channel.
- uint16_t GetChannelOffset (int channel) const

  Return the offset for one single channel.
- void SetAllChannelsOffset (uint16_t offset)

  Set the time offset for all channels.
- void SetCoarseCountOffset (uint16_t cco)

  Set offset for the coarse time counter.
- uint16_t GetCoarseCountOffset () const

  Extract offset for the coarse time counter.
- void SetDLLAdjustment (int tap, uint8_t adj)

  Set the DLL taps adjustments with a resolution of ∼10 ps.
- uint8_t GetDLLAdjustment (int tap) const

  Set the adjustment of DLL taps.
- void SetAllTapsDLLAdjustment (uint8_t adj)

  Extract the adjustment of DLL taps.

- void SetRCAdjustment (int tap, uint8_t adj)

  Set the adjustment of the RC delay line.
- uint8_t GetRCAdjustment (int tap)

  Extract the adjustment of the RC delay line.
- void SetWidthResolution (const WidthResolution r)

  Set the pulse width resolution when paired measurements are performed.
- WidthResolution GetWidthResolution () const

  Extract the pulse width resolution when paired measurements are performed.
- void SetVernierOffset (const uint8_t vo)

  Set the offset in vernier decoding.
- uint8_t GetVernierOffset () const

  Extract the offset in vernier decoding.
- void SetDeadTime (const DeadTime dt)

  Channel dead time between hits.
- DeadTime **GetDeadTime** () const
- void SetTestInvert (const bool ti=true)

  Automatic inversion of test pattern. Only used during production testing.
- bool **GetTestInvert** () const
- void SetTestMode (const bool tm=true)

  Test mode where hit data are taken from coretest. Only used during production testing.
- bool **GetTestMode** () const
- void SetTrailingMode (const bool trail=true)

  Enable/disable the detection of trailing edges.
- bool GetTrailingMode () const

  Extract the status for the detection of trailing edges.
- void SetLeadingMode (const bool lead=true)

  Enable the detection of leading edges.
- bool GetLeadingMode () const

  Extract the status for the detection of leading edges.
- void SetTriggerMatchingMode (const bool trig=true)

  Set the enable status of trigger matching mode.
- bool GetTriggerMatchingMode () const

  Extract the enable status of trigger matching mode.
- void SetEdgesPairing (const bool pair=true)

  Enable the pairing of leading and trailing edges (overrides individual enable of leading/trailing edges)
- bool **GetEdgesPairing** () const
- void SetSetupParity (const bool sp=true)

  Set the parity of setup data (should be an even parity)
- bool GetSetupParity () const

  Extract the parity of setup data (should be an even parity)
- void SetConstantValues ()

  Ensure that the critical constant values are properly set in the setup word.
- uint16_t GetTriggerLatency () const

  Effective trigger latency in number of clock cycles (when no counter roll-over is used)
- void SetTDCId (const uint8_t id=0x0)

  Set the unique identifier of the TDC object on the board.
- uint16_t GetTDCId () const

  Get the unique identifier of the TDC object on the board.
- void Dump (int verb=1, std::ostream &os=std::cout) const

  Printout all useful values of this setup register into an output stream.

Additional Inherited Members

3.5.1 Detailed Description

Setup word to be sent to the HPTDC chip.

Object handling the setup word provided by/to the HPTDC chip

Author

Laurent Forthomme laurent.forthomme@cern.ch
Lara Lloret lara@cern.ch

Date

16 Apr 2015
May 2016

3.5.2 Member Function Documentation

3.5.2.1 bool TDCSetup::GetRejectFIFOFull ( ) const [inline]

Are hits rejected when readout FIFO is full?

Extract whether or not hits are rejected once FIFO is full.

3.5.2.2 void TDCSetup::SetMaxEventSize (int *sz* = -1) [inline]

Set the maximum number of hits per event.

Set the maximum number of hits that can be recorded for each event. It is always rounded to the next power of 2 (in the range 0-128), and if lower than 0 or bigger than 128 then set to unimited.

3.5.2.3 void TDCSetup::SetRejectFIFOFull (const bool *rej* = true) [inline]

Reject hits when readout FIFO full.

Set whether or not hits are rejected once FIFO is full.

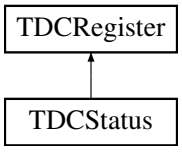
The documentation for this class was generated from the following file:

- include/TDCSetup.h

3.6 TDCStatus Class Reference

`#include <TDCStatus.h>`

Inheritance diagram for TDCStatus:



Classes

- struct ErrorType

  Type of error encountered by the HPTDC.

Public Types

- typedef struct TDCStatus::ErrorType ErrorType

  Type of error encountered by the HPTDC.

Public Member Functions

- TDCStatus ()

  Initialise a status register with all hardcoded values.
- TDCStatus (const std::vector< uint8_t > &words)

  Initialise a status register from a vector of 8-bit words.
- void SetConstantValues ()

  Set the hardcoded values to the register.
- ErrorType Error () const

  Retrieve the list of errors monitored.
- bool HasToken () const

  TDC have read-out token.
- uint16_t FIFOOccupancy () const

  Occupancy of readout FIFO.
- bool FIFOFull () const

  It the readout FIFO full?
- bool FIFOEmpty () const

  It the readout FIFO empty?
- uint32_t L1Occupancy (unsigned short group=-1) const

  Occupancy of L1 buffer in channels of a group (or all groups)
- uint16_t TriggerFIFOOccupancy () const

  Occupancy of trigger FIFO.
- bool TriggerFIFOFull () const

  Is the trigger FIFO full?
- bool TriggerFIFOEmpty () const

  Is the trigger FIFO empty?
- bool DLLLock () const

  Is the DLL in lock state?
- void Dump (int verb=1, std::ostream &os=std::cout) const

  Printout all useful values of this status register into an output stream.

Friends

- std::ostream & operator<< (std::ostream &out, const ErrorType &err)

  Printout the error type(s) into the output stream.

Additional Inherited Members

3.6.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch
Lara Lloret lara@cern.ch

Date

27 Apr 2015
May 2016

The documentation for this class was generated from the following file:

- include/TDCStatus.h

Index