VecTHOR's Retargeting Framework

Generated by Doxygen 1.8.20

Chapter 1

README

1.0.1 IMPORTANT

Use macro #define BUFFER_CTR_SIZE 12 for partcompressed and buffercompressed

For all other tests: //#define BUFFER_CTR_SIZE 12

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 vecthor Namespace Reference

Classes

- class CompressedEmitter
- class Compressor
- class CompressorCore
- class CompressorStats
- class Config
- class Decompressor
- class DecompressorStats
- class DynDecompressor
- class Emitter
- class EmitterStats
- class FormalDecompressor
- class FormalDecompressorStats
- class HardwareEmitter
- class LegacyEmitter
- class P2SBuffer
- class Plotter
- class Stats
- class TDRGen
- class TDRReader
- class Validator
- class VecTHOR

Typedefs

- using BitVec = std::vector< u_int8_t >
- using BitVecCPtr = std::shared_ptr< const BitVec >
- using BitVecItr = BitVec::iterator
- using BitVecCItr = BitVec::const_iterator
- using Replacement = std::tuple< CDW, BitVecCItr, BitVecCItr, short >
- using ReplacementPtr = std::shared_ptr< Replacement >
- using CDWMap = std::map< BitVecCItr, std::map< BitVecCItr, ReplacementPtr > >
- using CDWMapItem = std::pair< BitVecCItr, std::map< BitVecCItr, ReplacementPtr > >

- using Edge = std::tuple< short, short, ReplacementPtr >
- using Edges = std::vector< Edge >
- using Route = std::vector< ReplacementPtr >
- using FrequencyContainer = std::map< std::string, unsigned int >
- using FrequencyData = std::vector< std::pair< unsigned int, std::string > >
- using Signals = std::vector< std::tuple< VALUE, VALUE > >
- using CompressorPtr = std::unique_ptr< Compressor >
- using DecompressorPtr = std::unique_ptr< Decompressor >
- using CompressedEmitterPtr = std::unique_ptr< CompressedEmitter >
- using ValidatorPtr = std::unique_ptr< Validator >

Enumerations

- enum CFG : u_int8_t {
  CFG::UNSUPPORTED, CFG::MERGING, CFG::DYNAMIC, CFG::HEUR_INNER_FREQ,
  CFG::HEUR_OUTER_FREQ, CFG::HEUR_WEIGHT, CFG::HEUR_PERMUTE, CFG::SAT,
  CFG::SAT_SEC, CFG::SAT_CONFL, CFG::SAT_RESTART, CFG::MAX_CDWS,
  CFG::PART_SIZE, CFG::EXT_CDWS, CFG::VERBOSE, CFG::DEBUG,
  CFG::STATS, CFG::BENCHMARK, CFG::PLOT, CFG::HEX,
  CFG::P2S_BUFFER, CFG::USE_EXT_FILE, CFG::USE_CONF_FILE, CFG::GEN_LEGACY,
  CFG::GEN_COMPRESSED, CFG::GEN_GOLDEN, CFG::HW_EMIT, CFG::ALLOW_X,
  CFG::VALIDATE }
- enum FILE : u_int8_t {
  FILE::UNSUPPORTED, FILE::LEGACY_PREFIX, FILE::LEGACY_SUFFIX, FILE::DYNCOMPRESSED_INFIX,
  FILE::DYNCOMPRESSED_PRELOAD, FILE::COMPRESSED_PREFIX, FILE::COMPRESSED_SUFFIX,
  FILE::COMPRESSED_FILE,
  FILE::CONFIG_FILE, FILE::RESYNC_FILE, FILE::EXT_FILE, FILE::LEGACY_FILE,
  FILE::GOLDEN_FILE, FILE::VALIDATION_FILE }
- enum VALUE : uint8_t {
  VALUE::LOW = 0, VALUE::HIGH = 1, VALUE::NOP = 2, VALUE::INIT = 3,
  VALUE::UNSUPPORTED = 4 }
- enum CDW {
  CDW::NONE, CDW::XXX, CDW::HXX, CDW::LXX,
  CDW::LLX, CDW::LHX, CDW::HLX, CDW::HHX,
  CDW::LLL, CDW::LLH, CDW::LHL, CDW::LHH,
  CDW::HLL, CDW::HLH, CDW::HHL, CDW::HHH }
- enum REPL_FIELD : uint8_t { REPL_FIELD::CDW, REPL_FIELD::START, REPL_FIELD::END,
  REPL_FIELD::WEIGHT }

Functions

- unsigned int countBitVecX (BitVecCItr start, BitVecCItr end)
- std::string serializeBitVec (BitVecCItr start, BitVecCItr end)
- void writeBitVec (const BitVec &bit_vec, std::ostream *stream=&std::cout, bool header=true)
- void writeBitVec (BitVecCItr start, BitVecCItr end, std::ostream *stream=&std::cout)
- template<class T1 , class T2 >
  std::map< T2, T1 > swapPairs (std::map< T1, T2 > m)

6.1.1 Typedef Documentation

6.1.1.1 BitVec

```
using vecthor::BitVec = typedef std::vector<u_int8_t>
```

6.1.1.2 BitVecCItr

```
using vecthor::BitVecCItr = typedef BitVec::const_iterator
```

6.1.1.3 BitVecCPtr

```
using vecthor::BitVecCPtr = typedef std::shared_ptr<const BitVec>
```

6.1.1.4 BitVecItr

```
using vecthor::BitVecItr = typedef BitVec::iterator
```

6.1.1.5 CDWMap

```
using vecthor::CDWMap = typedef std::map<BitVecCItr, std::map<BitVecCItr, ReplacementPtr> >
```

6.1.1.6 CDWMapItem

```
using vecthor::CDWMapItem = typedef std::pair<BitVecCItr, std::map<BitVecCItr, ReplacementPtr>
>
```

6.1.1.7 CompressedEmitterPtr

```
using vecthor::CompressedEmitterPtr = typedef std::unique_ptr<CompressedEmitter>
```

6.1.1.8 CompressorPtr

```
using vecthor::CompressorPtr = typedef std::unique_ptr<Compressor>
```

6.1.1.9 DecompressorPtr

```
using vecthor::DecompressorPtr = typedef std::unique_ptr<Decompressor>
```

6.1.1.10 Edge

```
using vecthor::Edge = typedef std::tuple<short, short, ReplacementPtr>
```

6.1.1.11 Edges

```
using vecthor::Edges = typedef std::vector<Edge>
```

6.1.1.12 FrequencyContainer

```
using vecthor::FrequencyContainer = typedef std::map<std::string, unsigned int>
```

6.1.1.13 FrequencyData

```
using vecthor::FrequencyData = typedef std::vector<std::pair<unsigned int, std::string> >
```

6.1.1.14 Replacement

```
using vecthor::Replacement = typedef std::tuple<CDW, BitVecCItr, BitVecCItr, short>
```

6.1.1.15 ReplacementPtr

```
using vecthor::ReplacementPtr = typedef std::shared_ptr<Replacement>
```

6.1.1.16 Route

```
using vecthor::Route = typedef std::vector<ReplacementPtr>
```

6.1.1.17 Signals

```
using vecthor::Signals = typedef std::vector<std::tuple<VALUE, VALUE> >
```

6.1.1.18 ValidatorPtr

```
using vecthor::ValidatorPtr = typedef std::unique_ptr<Validator>
```

6.1.2 Enumeration Type Documentation

6.1.2.1 CDW

```
enum vecthor::CDW  [strong]
```

Enumerator

NONE	
XXX	
HXX	
LXX	
LLX	
LHX	
HLX	
HHX	
LLL	
LLH	
LHL	
LHH	
HLL	
HLH	
HHL	
HHH	

6.1.2.2 CFG

enum vecthor::CFG : u_int8_t [strong]

Enumerator

UNSUPPORTED	
MERGING	
DYNAMIC	
HEUR_INNER_FREQ	
HEUR_OUTER_FREQ	
HEUR_WEIGHT	
HEUR_PERMUTE	
SAT	
SAT_SEC	
SAT_CONFL	
SAT_RESTART	
MAX_CDWS	
PART_SIZE	
EXT_CDWS	
VERBOSE	
DEBUG	
STATS	
BENCHMARK	
PLOT	
HEX	
P2S_BUFFER	
USE_EXT_FILE	
USE_CONF_FILE	
GEN_LEGACY	
GEN_COMPRESSED	
GEN_GOLDEN	
HW_EMIT	
ALLOW_X	
VALIDATE	

6.1.2.3 FILE

enum vecthor::FILE : u_int8_t [strong]

Enumerator

UNSUPPORTED	
LEGACY_PREFIX	
LEGACY_SUFFIX	
DYNCOMPRESSED_INFIX	
DYNCOMPRESSED_PRELOAD	
COMPRESSED_PREFIX	

Enumerator

COMPRESSED_SUFFIX	
COMPRESSED_FILE	
CONFIG_FILE	
RESYNC_FILE	
EXT_FILE	
LEGACY_FILE	
GOLDEN_FILE	
VALIDATION_FILE	

6.1.2.4 REPL_FIELD

```
enum vecthor::REPL_FIELD : uint8_t [strong]
```

Enumerator

CDW	
START	
END	
WEIGHT	

6.1.2.5 VALUE

```
enum vecthor::VALUE : uint8_t [strong]
```

Enumerator

LOW	
HIGH	
NOP	
INIT	
UNSUPPORTED	

6.1.3 Function Documentation

6.1.3.1 countBitVecX()

```
unsigned int vecthor::countBitVecX (
            BitVecCItr start,
            BitVecCItr end )
```

6.1.3.2 serializeBitVec()

```
std::string vecthor::serializeBitVec (
            BitVecCItr start,
            BitVecCItr end )
```

6.1.3.3 swapPairs()

```
template<class T1 , class T2 >
std::map<T2, T1> vecthor::swapPairs (
            std::map< T1, T2 > m )
```

6.1.3.4 writeBitVec() [1/2]

```
void vecthor::writeBitVec (
            BitVecCItr start,
            BitVecCItr end,
            std::ostream * stream = &std::cout )
```

6.1.3.5 writeBitVec() [2/2]

```
void vecthor::writeBitVec (
            const BitVec & bit_vec,
            std::ostream * stream = &std::cout,
            bool header = true )
```

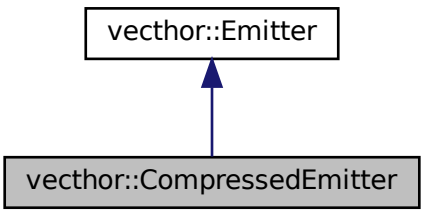
Chapter 7

Class Documentation

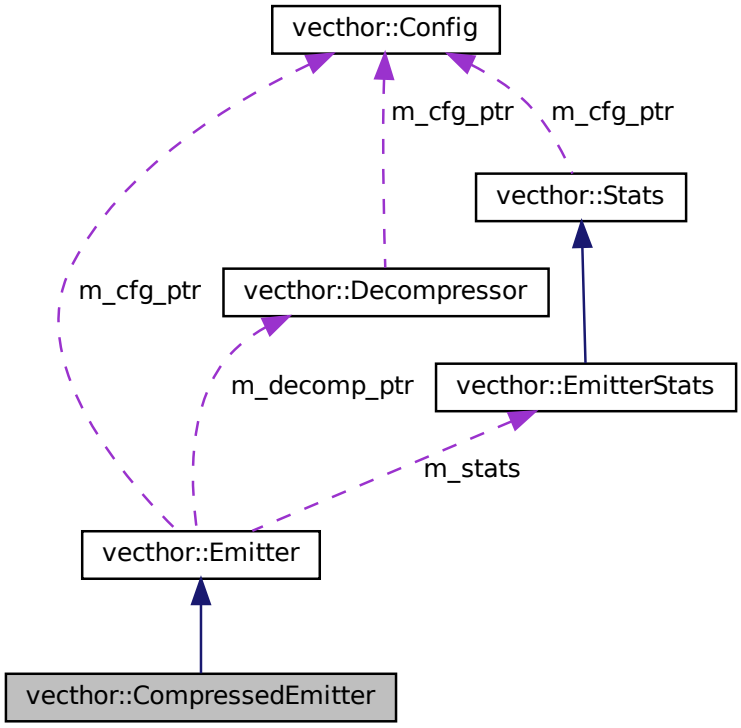
7.1 vecthor::CompressedEmitter Class Reference

```
#include <Emitter.h>
```

Inheritance diagram for vecthor::CompressedEmitter:



Collaboration diagram for vecthor::CompressedEmitter:



Public Member Functions

- CompressedEmitter (const Config *config, const Decompressor *decomp_ptr)
- ~CompressedEmitter ()
- void init ()
- void finalize ()
- void operator() (const Route &route, unsigned int delay=0)
- void writeResyncFile (const P2SBuffer::DataCollector &data_collector, unsigned int delay)

Private Member Functions

- void addSignalValue (const char &a, const char &b)
- void writeJTAG (const Route &route)
- void writePreload (unsigned int delay)
- void writeComprInstr ()
- void writePreloadInstr ()

Private Attributes

- std::ofstream m_compr_file
- Signals m_signals

Additional Inherited Members

7.1.1 Constructor & Destructor Documentation

7.1.1.1 CompressedEmitter()

```
vecthor::CompressedEmitter::CompressedEmitter (
            const Config * config,
            const Decompressor * decomp_ptr )
```

7.1.1.2 ~CompressedEmitter()

```
vecthor::CompressedEmitter::~CompressedEmitter ( )
```

7.1.2 Member Function Documentation

7.1.2.1 addSignalValue()

```
void vecthor::CompressedEmitter::addSignalValue (
            const char & a,
            const char & b )  [private]
```

7.1.2.2 finalize()

```
void vecthor::CompressedEmitter::finalize ( )
```

7.1.2.3 init()

```
void vecthor::CompressedEmitter::init ( )
```

7.1.2.4 operator()()

```
void vecthor::CompressedEmitter::operator() (
            const Route & route,
            unsigned int delay = 0 )
```

7.1.2.5 writeComprInstr()

```
void vecthor::CompressedEmitter::writeComprInstr ( )  [private]
```

7.1.2.6 writeJTAG()

```
void vecthor::CompressedEmitter::writeJTAG (
            const Route & route )  [private]
```

7.1.2.7 writePreload()

```
void vecthor::CompressedEmitter::writePreload (
            unsigned int delay )  [private]
```

7.1.2.8 writePreloadInstr()

```
void vecthor::CompressedEmitter::writePreloadInstr ( )  [private]
```

7.1.2.9 writeResyncFile()

```
void vecthor::CompressedEmitter::writeResyncFile (
            const P2SBuffer::DataCollector & data_collector,
            unsigned int delay )
```

7.1.3 Member Data Documentation

7.1.3.1 m_compr_file

```
std::ofstream vecthor::CompressedEmitter::m_compr_file  [private]
```

7.1.3.2 m_signals

```
Signals vecthor::CompressedEmitter::m_signals  [private]
```

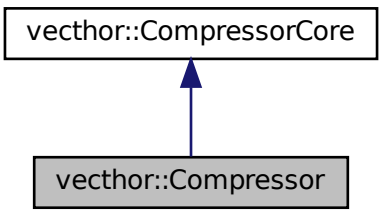
The documentation for this class was generated from the following file:

- src/Emitter.h

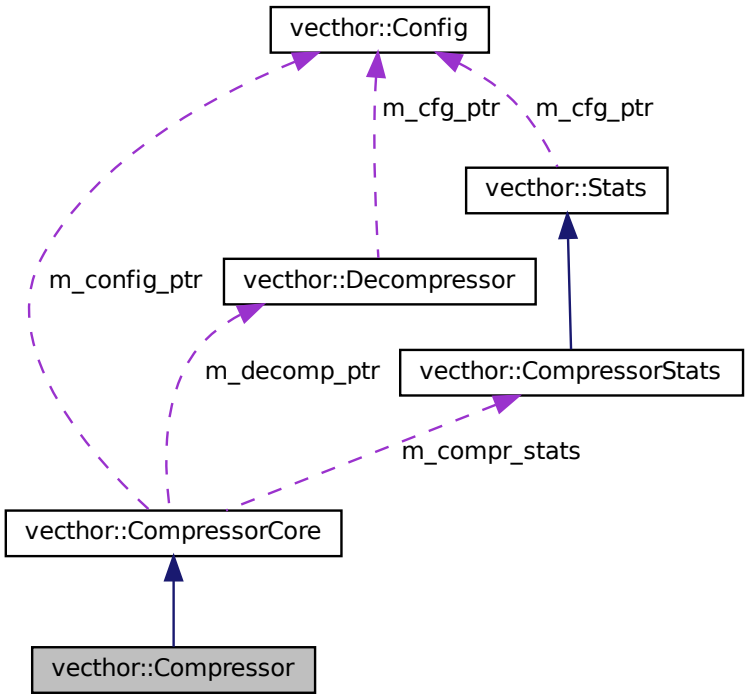
7.2 vecthor::Compressor Class Reference

```
#include <Compressor.h>
```

Inheritance diagram for vecthor::Compressor:



Collaboration diagram for vecthor::Compressor:



Public Member Functions

- Compressor (const Config *config, const Decompressor *decompr)
- void greedy (Route &route)
- Route formal (Route &repl_vec)

Private Member Functions

- void addToRoute (Route &route, ReplacementPtr repl)
- int calculateCDWByte (BitVecCItr start, BitVecCItr end, CDWMap &cdw_map)
- ReplacementPtr classifyCDW (BitVecCItr start, BitVecCItr end)
- float determineCoverage () const
- void dumpCDWmap (const CDWMap &cdw_map, bool force=false) const
- void dumpStart (const Edges &data, bool force=false) const
- void dumpRoute (const Route &route, bool force=false) const
- void dumpReplacement (const ReplacementPtr repl, bool force=false) const
- Edges determineStart (CDWMap &cdw_map)
- void determineRoute (Edges &init, Route &route)
- void fillGap (Route &route, BitVecItr l_start, BitVecItr l_end, const CDWMap &cdw_map)
- void finalizeRoute (Route &route)
- int getCoveragePos (BitVecItr &cov_pos_it) const
- void mergeRepl (ReplacementPtr fst, ReplacementPtr snd)
- void mergeRoute (Route &route)

- bool isCovered (BitVecCItr cov_start_it, BitVecCItr cov_end_it) const
- bool isSingleBit (BitVecCItr &l_start, BitVecCItr &l_end) const
- void postProcRoute (Route &route, const CDWMap &cdw_map)
- void printCDWUsage () const
- void setCovered (BitVecItr &l_start, BitVecItr &l_end)
- void sortRoute (Route &route)
- void addToCoveredRoute (Route &route, ReplacementPtr repl)

Additional Inherited Members

7.2.1 Constructor & Destructor Documentation

7.2.1.1 Compressor()

```
vecthor::Compressor::Compressor (
            const Config * config,
            const Decompressor * decompr )  [inline], [explicit]
```

7.2.2 Member Function Documentation

7.2.2.1 addToCoveredRoute()

```
void vecthor::Compressor::addToCoveredRoute (
            Route & route,
            ReplacementPtr repl )  [private]
```

7.2.2.2 addToRoute()

```
void vecthor::Compressor::addToRoute (
            Route & route,
            ReplacementPtr repl )  [private]
```

7.2.2.3 calculateCDWByte()

```
int vecthor::Compressor::calculateCDWByte (
            BitVecCItr start,
            BitVecCItr end,
            CDWMap & cdw_map )  [private]
```

7.2.2.4 classifyCDW()

```
ReplacementPtr vecthor::Compressor::classifyCDW (
            BitVecCItr start,
            BitVecCItr end )  [private]
```

7.2.2.5 determineCoverage()

```
float vecthor::Compressor::determineCoverage ( ) const  [private]
```

7.2.2.6 determineRoute()

```
void vecthor::Compressor::determineRoute (
            Edges & init,
            Route & route )  [private]
```

7.2.2.7 determineStart()

```
Edges vecthor::Compressor::determineStart (
            CDWMap & cdw_map )  [private]
```

7.2.2.8 dumpCDWmap()

```
void vecthor::Compressor::dumpCDWmap (
            const CDWMap & cdw_map,
            bool force = false ) const  [private]
```

7.2.2.9 dumpReplacement()

```
void vecthor::Compressor::dumpReplacement (
            const ReplacementPtr repl,
            bool force = false ) const  [private]
```

7.2.2.10 dumpRoute()

```
void vecthor::Compressor::dumpRoute (
            const Route & route,
            bool force = false ) const  [private]
```

7.2.2.11 dumpStart()

```
void vecthor::Compressor::dumpStart (
            const Edges & data,
            bool force = false ) const  [private]
```

7.2.2.12 fillGap()

```
void vecthor::Compressor::fillGap (
            Route & route,
            BitVecItr l_start,
            BitVecItr l_end,
            const CDWMap & cdw_map )  [private]
```

7.2.2.13 finalizeRoute()

```
void vecthor::Compressor::finalizeRoute (
            Route & route )  [private]
```

7.2.2.14 formal()

```
Route vecthor::Compressor::formal (
            Route & repl_vec )
```

7.2.2.15 getCoveragePos()

```
int vecthor::Compressor::getCoveragePos (
            BitVecItr & cov_pos_it ) const  [private]
```

7.2.2.16 greedy()

```
void vecthor::Compressor::greedy (
            Route & route )
```

7.2.2.17 isCovered()

```
bool vecthor::Compressor::isCovered (
            BitVecCItr cov_start_it,
            BitVecCItr cov_end_it ) const  [private]
```

7.2.2.18 isSingleBit()

```
bool vecthor::Compressor::isSingleBit (
            BitVecCItr & l_start,
            BitVecCItr & l_end ) const  [private]
```

7.2.2.19 mergeRepl()

```
void vecthor::Compressor::mergeRepl (
            ReplacementPtr fst,
            ReplacementPtr snd )  [private]
```

7.2.2.20 mergeRoute()

```
void vecthor::Compressor::mergeRoute (
            Route & route )  [private]
```

7.2.2.21 postProcRoute()

```
void vecthor::Compressor::postProcRoute (
            Route & route,
            const CDWMap & cdw_map )  [private]
```

7.2.2.22 printCDWUsage()

```
void vecthor::Compressor::printCDWUsage ( ) const  [private]
```

7.2.2.23 setCovered()

```
void vecthor::Compressor::setCovered (
            BitVecItr & l_start,
            BitVecItr & l_end )  [private]
```

7.2.2.24 sortRoute()

```
void vecthor::Compressor::sortRoute (
            Route & route )  [private]
```

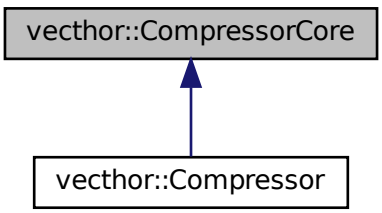
The documentation for this class was generated from the following file:

- src/Compressor.h

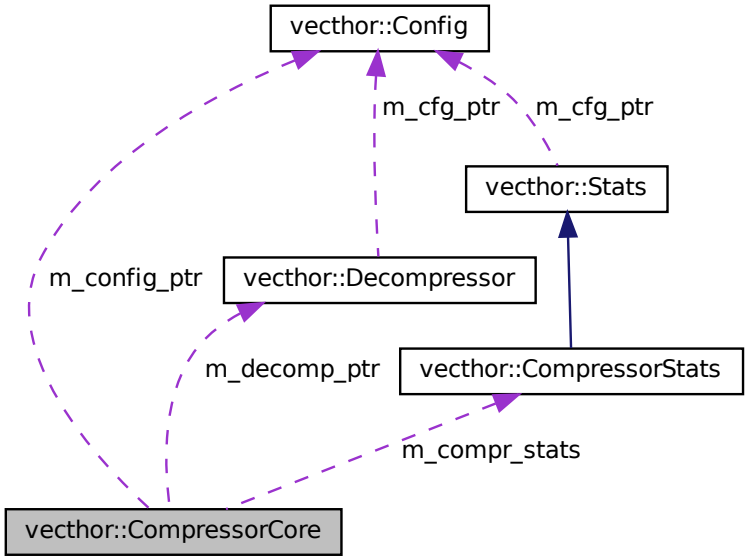
7.3 vecthor::CompressorCore Class Reference

```
#include <CompressorCore.h>
```

Inheritance diagram for vecthor::CompressorCore:



Collaboration diagram for vecthor::CompressorCore:



Public Member Functions

- CompressorCore (const Config ∗config, const Decompressor ∗decomp)
- void printStats (const std::string &title, std::ostream &out=std::cout) const
- CompressorStats & getStats ()
- void prepare (BitVecCItr &bv_begin, BitVecCItr &bv_end)
- void reset ()

Protected Member Functions

- bool isCompleteRoute () const
- bool isDebug () const
- bool isVerbose () const

Protected Attributes

- BitVecCItr m_bit_vec_begin
- BitVecCItr m_bit_vec_end
- const Config ∗ m_config_ptr
- const Decompressor ∗ m_decomp_ptr
- CompressorStats m_compr_stats
- BitVec ∗ m_cover_ptr

7.3.1 Constructor & Destructor Documentation

7.3.1.1 CompressorCore()

```
vecthor::CompressorCore::CompressorCore (
            const Config * config,
            const Decompressor * decomp )  [inline], [explicit]
```

7.3.2 Member Function Documentation

7.3.2.1 getStats()

```
CompressorStats& vecthor::CompressorCore::getStats ( )  [inline]
```

7.3.2.2 isCompleteRoute()

```
bool vecthor::CompressorCore::isCompleteRoute ( ) const  [protected]
```

7.3.2.3 isDebug()

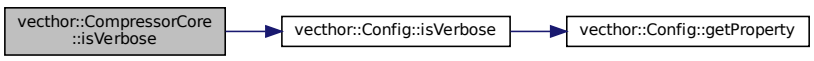
```
bool vecthor::CompressorCore::isDebug ( ) const  [inline], [protected]
```

Here is the call graph for this function:

7.3.2.4 isVerbose()

```
bool vecthor::CompressorCore::isVerbose ( ) const  [inline], [protected]
```

Here is the call graph for this function:



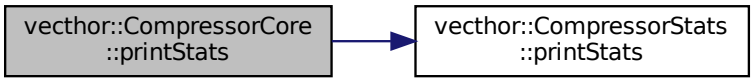
7.3.2.5 prepare()

```
void vecthor::CompressorCore::prepare (
            BitVecCItr & bv_begin,
            BitVecCItr & bv_end )
```

7.3.2.6 printStats()

```
void vecthor::CompressorCore::printStats (
            const std::string & title,
            std::ostream & out = std::cout ) const  [inline]
```

Here is the call graph for this function:



7.3.2.7 reset()

```
void vecthor::CompressorCore::reset ( )
```

7.3.3 Member Data Documentation

7.3.3.1 m_bit_vec_begin

```
BitVecCItr vecthor::CompressorCore::m_bit_vec_begin  [protected]
```

7.3.3.2 m_bit_vec_end

```
BitVecCItr vecthor::CompressorCore::m_bit_vec_end  [protected]
```

7.3.3.3 m_compr_stats

```
CompressorStats vecthor::CompressorCore::m_compr_stats  [protected]
```

7.3.3.4 m_config_ptr

```
const Config* vecthor::CompressorCore::m_config_ptr  [protected]
```

7.3.3.5 m_cover_ptr

```
BitVec* vecthor::CompressorCore::m_cover_ptr  [protected]
```

7.3.3.6 m_decomp_ptr

```
const Decompressor* vecthor::CompressorCore::m_decomp_ptr  [protected]
```

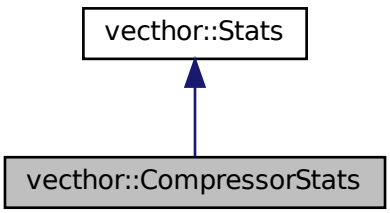
The documentation for this class was generated from the following file:

- src/CompressorCore.h

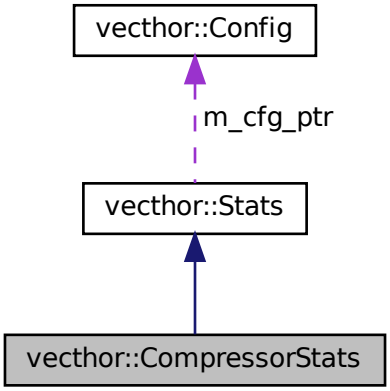
7.4 vecthor::CompressorStats Class Reference

```
#include <Stats.h>
```

Inheritance diagram for vecthor::CompressorStats:



Collaboration diagram for vecthor::CompressorStats:



Public Member Functions

- CompressorStats (const Config *cfg_ptr)
- void printStats (const std::string &title="", std::ostream &out=std::cout) const
- void collectBenchmarkData () override
- void clear ()
- void printBenchmarkData (const std::string &title="") const override

Public Attributes

- unsigned int m_num_sbf = 0
- unsigned int m_num_benefit = 0
- unsigned int m_num_bit = 0
- unsigned int m_num_replacements = 0
- unsigned int m_num_s1_repls = 0
- unsigned int m_num_s2_repls = 0
- unsigned int m_num_cdw_repetition = 0
- unsigned int m_num_red_repetition = 0
- unsigned int m_num_overall_bit = 0
- unsigned int m_num_overall_compressed_bit = 0
- unsigned int m_num_overall_mc_overhead_bit = 0
- std::map< CDW, int > m_counter_cdws = std::map<CDW, int>()
- std::map< unsigned, long long > m_counter_cdws_length

Private Member Functions

- int getComprBit () const
- void printCDWUsage (std::ostream &out) const

Additional Inherited Members

7.4.1 Constructor & Destructor Documentation

7.4.1.1 CompressorStats()

```
vecthor::CompressorStats::CompressorStats (
            const Config * cfg_ptr )  [inline]
```

7.4.2 Member Function Documentation

7.4.2.1 clear()

```
void vecthor::CompressorStats::clear ( )
```

7.4.2.2 collectBenchmarkData()

```
void vecthor::CompressorStats::collectBenchmarkData ( )  [override], [virtual]
```

Implements vecthor::Stats.

7.4.2.3 getComprBit()

```
int vecthor::CompressorStats::getComprBit ( ) const  [inline], [private]
```

7.4.2.4 printBenchmarkData()

```
void vecthor::CompressorStats::printBenchmarkData (
            const std::string & title = "" ) const  [override], [virtual]
```

Implements vecthor::Stats.

7.4.2.5 printCDWUsage()

```
void vecthor::CompressorStats::printCDWUsage (
            std::ostream & out ) const  [private]
```

7.4.2.6 printStats()

```
void vecthor::CompressorStats::printStats (
            const std::string & title = "",
            std::ostream & out = std::cout ) const  [virtual]
```

Implements vecthor::Stats.

7.4.3 Member Data Documentation

7.4.3.1 m_counter_cdws

```
std::map<CDW, int> vecthor::CompressorStats::m_counter_cdws = std::map<CDW, int>()
```

7.4.3.2 m_counter_cdws_length

```
std::map<unsigned, long long> vecthor::CompressorStats::m_counter_cdws_length
```

Initial value:

```
=
      std::map<unsigned, long long>()
```

7.4.3.3 m_num_benefit

```
unsigned int vecthor::CompressorStats::m_num_benefit = 0
```

7.4.3.4 m_num_bit

```
unsigned int vecthor::CompressorStats::m_num_bit = 0
```

7.4.3.5 m_num_cdw_repetition

```
unsigned int vecthor::CompressorStats::m_num_cdw_repetition = 0
```

7.4.3.6 m_num_overall_bit

```
unsigned int vecthor::CompressorStats::m_num_overall_bit = 0
```

7.4.3.7 m_num_overall_compressed_bit

```
unsigned int vecthor::CompressorStats::m_num_overall_compressed_bit = 0
```

7.4.3.8 m_num_overall_mc_overhead_bit

```
unsigned int vecthor::CompressorStats::m_num_overall_mc_overhead_bit = 0  [mutable]
```

7.4.3.9 m_num_red_repetition

```
unsigned int vecthor::CompressorStats::m_num_red_repetition = 0
```

7.4.3.10 m_num_replacements

```
unsigned int vecthor::CompressorStats::m_num_replacements = 0
```

7.4.3.11 m_num_s1_repls

```
unsigned int vecthor::CompressorStats::m_num_s1_repls = 0
```

7.4.3.12 m_num_s2_repls

```
unsigned int vecthor::CompressorStats::m_num_s2_repls = 0
```

7.4.3.13 m_num_sbf

```
unsigned int vecthor::CompressorStats::m_num_sbf = 0
```

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

- src/Stats.h

7.5 vecthor::Config Class Reference

```
#include <Config.h>
```

Public Member Functions

- Config ()
- ~Config ()
- bool isDebug () const
- bool isVerbose () const
- CFG getCFGType (std::string &property) const
- FILE getFILEType (std::string &property) const
- OfStrPtr getBenchmarkFile () const
- unsigned int getProperty (CFG cfg) const
- const std::string & getFile (FILE file) const
- void setProperty (CFG cfg, int value=1)
- void setFile (FILE file, std::string &filename)
- bool parseArgs (int argc, char **argv)
- bool parseConfig ()
- void prepare (const std::string &run_name)
- void preloadCDW ()
- void dump ()
- void printIcon ()

Public Attributes

- unsigned int m_num_rtdr = 16

Private Types

- using OfStrPtr = std::shared_ptr< std::ofstream >
- using CfgStringMap = std::map< CFG, std::string >
- using FileStringMap = std::map< FILE, std::string >

Private Member Functions

- void initialize ()
- std::string CFGtoString (CFG cfg) const

Private Attributes

- std::map< CFG, int > m_cfg_map
- CfgStringMap m_cfg_str
- std::map< FILE, std::string > m_file_map
- FileStringMap m_file_str
- OfStrPtr m_benchmark_filep

7.5.1 Member Typedef Documentation

7.5.1.1 CfgStringMap

```
using vecthor::Config::CfgStringMap = std::map<CFG, std::string>  [private]
```

7.5.1.2 FileStringMap

```
using vecthor::Config::FileStringMap = std::map<FILE, std::string>  [private]
```

7.5.1.3 OfStrPtr

```
using vecthor::Config::OfStrPtr = std::shared_ptr<std::ofstream>  [private]
```

7.5.2 Constructor & Destructor Documentation

7.5.2.1 Config()

```
vecthor::Config::Config ( )
```

7.5.2.2 ~Config()

```
vecthor::Config::~Config ( )
```

7.5.3 Member Function Documentation

7.5.3.1 CFGtoString()

```
std::string vecthor::Config::CFGtoString (
            CFG cfg ) const  [private]
```

7.5.3.2 dump()

```
void vecthor::Config::dump ( )
```

7.5.3.3 getBenchmarkFile()

```
OfStrPtr vecthor::Config::getBenchmarkFile ( ) const
```

7.5.3.4 getCFGType()

```
CFG vecthor::Config::getCFGType (
            std::string & property ) const
```

7.5.3.5 getFile()

```
const std::string& vecthor::Config::getFile (
            FILE file ) const
```

7.5.3.6 getFILEType()

```
FILE vecthor::Config::getFILEType (
            std::string & property ) const
```

7.5.3.7 getProperty()

```
unsigned int vecthor::Config::getProperty (
            CFG cfg ) const
```

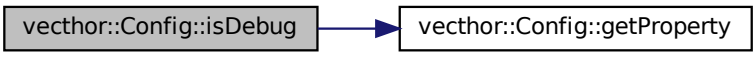
7.5.3.8 initialize()

```
void vecthor::Config::initialize ( )  [private]
```

7.5.3.9 isDebug()

```
bool vecthor::Config::isDebug ( ) const  [inline]
```

Here is the call graph for this function:



7.5.3.10 isVerbose()

```
bool vecthor::Config::isVerbose ( ) const  [inline]
```

Here is the call graph for this function:

7.5.3.11 parseArgs()

```
bool vecthor::Config::parseArgs (
            int argc,
            char ** argv )
```

7.5.3.12 parseConfig()

```
bool vecthor::Config::parseConfig ( )
```

7.5.3.13 preloadCDW()

```
void vecthor::Config::preloadCDW ( )
```

7.5.3.14 prepare()

```
void vecthor::Config::prepare (
            const std::string & run_name )
```

7.5.3.15 printIcon()

```
void vecthor::Config::printIcon ( )
```

7.5.3.16 setFile()

```
void vecthor::Config::setFile (
            FILE file,
            std::string & filename )
```

7.5.3.17 setProperty()

```
void vecthor::Config::setProperty (
            CFG cfg,
            int value = 1 )
```

7.5.4 Member Data Documentation

7.5.4.1 m_benchmark_filep

```
OfStrPtr vecthor::Config::m_benchmark_filep  [private]
```

7.5.4.2 m_cfg_map

```
std::map<CFG, int> vecthor::Config::m_cfg_map  [private]
```

7.5.4.3 m_cfg_str

```
CfgStringMap vecthor::Config::m_cfg_str  [private]
```

7.5.4.4 m_file_map

```
std::map<FILE, std::string> vecthor::Config::m_file_map  [private]
```

7.5.4.5 m_file_str

```
FileStringMap vecthor::Config::m_file_str  [private]
```

7.5.4.6 m_num_rtdr

```
unsigned int vecthor::Config::m_num_rtdr = 16
```

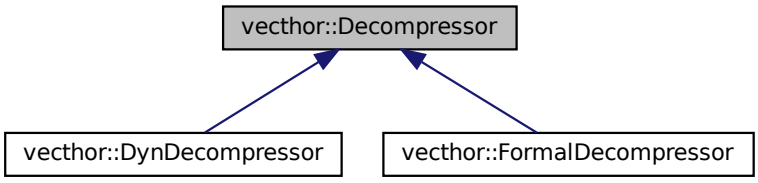
The documentation for this class was generated from the following file:

- src/Config.h

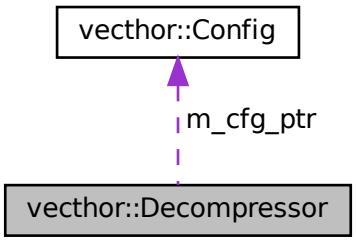
7.6 vecthor::Decompressor Class Reference

`#include <Decompressor.h>`

Inheritance diagram for vecthor::Decompressor:



Collaboration diagram for vecthor::Decompressor:



Public Types

- using CDWStringMap = std::map< CDW, std::string >
- using UDWStringMap = std::map< std::string, CDW >
- using CDWBenefitMap = std::map< CDW, short >

Public Member Functions

- Decompressor (const Config *cfg_ptr)
- const std::vector< CDW > & getTBRs () const
- const std::vector< std::string > & getTBCs () const
- size_t numTBCs () const
- size_t lengthTBCs () const
- CDW getCDW (const boost::dynamic_bitset<> &bit_str) const
- CDW getCDW (const std::string &bit_str) const
- std::string CDWtoEncoding (CDW cdw) const

- std::string CDWtoString (CDW cdw) const
- short getCDWBenefit (CDW cdw)
- u_int8_t getCDWLength (CDW &cdw) const
- virtual void determineCDW (BitVecCItr &, BitVecCItr &)
- virtual void clear ()
- virtual Stats & getStats ()
- void reset ()
- void dumpEntries () const
- void dumpConfiguration () const
- const UDWStringMap extractUDW () const

Static Public Member Functions

- static bool isUDWLength (BitVecCItr &l_start, BitVecCItr &l_end, unsigned int lb=0)
- static bool isEmptyCDW (CDW result)
- static bool isValidCDW (CDW result)
- static bool isStaticCDW (CDW result, bool ext_cdws)

Protected Member Functions

- bool storeDynCDW (std::string &cdw_repl)
- void preloadCDW ()
- bool storeDynCDW (boost::dynamic_bitset<> cdw_repl)

Protected Attributes

- const Config * m_cfg_ptr

Private Attributes

- CDWStringMap m_cdw_map
- UDWStringMap m_udw_map
- CDWBenefitMap m_cdw_benefit
- std::map< const CDW, u_int8_t > m_cdw_weight
- std::vector< CDW > m_tbrs
- std::vector< std::string > m_tbcs

7.6.1 Member Typedef Documentation

7.6.1.1 CDWBenefitMap

```
using vecthor::Decompressor::CDWBenefitMap = std::map<CDW, short>
```

7.6.1.2 CDWStringMap

```
using vecthor::Decompressor::CDWStringMap = std::map<CDW, std::string>
```

7.6.1.3 UDWStringMap

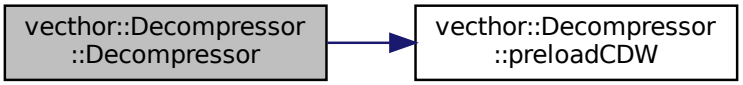
```
using vecthor::Decompressor::UDWStringMap = std::map<std::string, CDW>
```

7.6.2 Constructor & Destructor Documentation

7.6.2.1 Decompressor()

```
vecthor::Decompressor::Decompressor (
            const Config * cfg_ptr )  [inline]
```

Here is the call graph for this function:



7.6.3 Member Function Documentation

7.6.3.1 CDWtoEncoding()

```
std::string vecthor::Decompressor::CDWtoEncoding (
            CDW cdw ) const
```

7.6.3.2 CDWtoString()

```
std::string vecthor::Decompressor::CDWtoString (
            CDW cdw ) const
```

7.6.3.3 clear()

```
virtual void vecthor::Decompressor::clear ( )  [inline], [virtual]
```

Reimplemented in vecthor::FormalDecompressor, and vecthor::DynDecompressor.

7.6.3.4 determineCDW()

```
virtual void vecthor::Decompressor::determineCDW (
            BitVecCItr & ,
            BitVecCItr &  )  [inline], [virtual]
```

Reimplemented in vecthor::FormalDecompressor, and vecthor::DynDecompressor.

7.6.3.5 dumpConfiguration()

```
void vecthor::Decompressor::dumpConfiguration ( ) const
```

7.6.3.6 dumpEntries()

```
void vecthor::Decompressor::dumpEntries ( ) const
```

7.6.3.7 extractUDW()

```
const UDWStringMap vecthor::Decompressor::extractUDW ( ) const
```

7.6.3.8 getCDW() [1/2]

```
CDW vecthor::Decompressor::getCDW (
            const boost::dynamic_bitset<> & bit_str ) const
```

7.6.3.9 getCDW() [2/2]

```
CDW vecthor::Decompressor::getCDW (
            const std::string & bit_str ) const
```

7.6.3.10 getCDWBenefit()

```
short vecthor::Decompressor::getCDWBenefit (
            CDW cdw )
```

7.6.3.11 getCDWLength()

```
u_int8_t vecthor::Decompressor::getCDWLength (
            CDW & cdw ) const
```

7.6.3.12 getStats()

```
virtual Stats& vecthor::Decompressor::getStats ( )  [inline], [virtual]
```

Reimplemented in vecthor::FormalDecompressor, and vecthor::DynDecompressor.

7.6.3.13 getTBCs()

```
const std::vector<std::string>& vecthor::Decompressor::getTBCs ( ) const
```

7.6.3.14 getTBRs()

```
const std::vector<CDW>& vecthor::Decompressor::getTBRs ( ) const
```

7.6.3.15 isEmptyCDW()

```
static bool vecthor::Decompressor::isEmptyCDW (
            CDW result )  [static]
```

7.6.3.16 isStaticCDW()

```
static bool vecthor::Decompressor::isStaticCDW (
            CDW result,
            bool ext_cdws )  [static]
```

7.6.3.17 isUDWLength()

```
static bool vecthor::Decompressor::isUDWLength (
            BitVecCItr & l_start,
            BitVecCItr & l_end,
            unsigned int lb = 0 )  [static]
```

7.6.3.18 isValidCDW()

```
static bool vecthor::Decompressor::isValidCDW (
            CDW result )  [static]
```

7.6.3.19 lengthTBCs()

```
size_t vecthor::Decompressor::lengthTBCs ( ) const
```

7.6.3.20 numTBCs()

```
size_t vecthor::Decompressor::numTBCs ( ) const
```

7.6.3.21 preloadCDW()

```
void vecthor::Decompressor::preloadCDW ( )  [protected]
```

7.6.3.22 reset()

```
void vecthor::Decompressor::reset ( )
```

7.6.3.23 storeDynCDW() [1/2]

```
bool vecthor::Decompressor::storeDynCDW (
            boost::dynamic_bitset<> cdw_repl )  [protected]
```

7.6.3.24 storeDynCDW() [2/2]

```
bool vecthor::Decompressor::storeDynCDW (
            std::string & cdw_repl )  [protected]
```

7.6.4 Member Data Documentation

7.6.4.1 m_cdw_benefit

```
CDWBenefitMap vecthor::Decompressor::m_cdw_benefit  [private]
```

7.6.4.2 m_cdw_map

```
CDWStringMap vecthor::Decompressor::m_cdw_map  [private]
```

7.6.4.3 m_cdw_weight

```
std::map<const CDW, u_int8_t> vecthor::Decompressor::m_cdw_weight  [private]
```

7.6.4.4 m_cfg_ptr

```
const Config* vecthor::Decompressor::m_cfg_ptr  [protected]
```

7.6.4.5 m_tbcs

```
std::vector<std::string> vecthor::Decompressor::m_tbcs  [private]
```

7.6.4.6 m_tbrs

```
std::vector<CDW> vecthor::Decompressor::m_tbrs  [private]
```

7.6.4.7 m_udw_map

```
UDWStringMap vecthor::Decompressor::m_udw_map  [private]
```

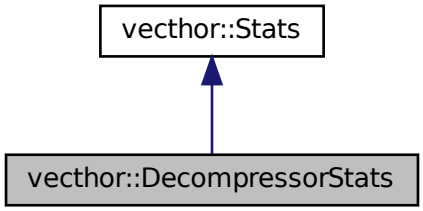
The documentation for this class was generated from the following file:

- src/Decompressor.h

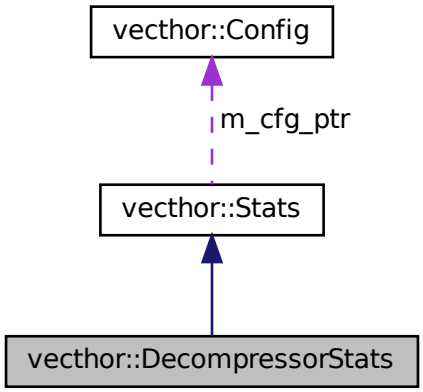
7.7 vecthor::DecompressorStats Class Reference

```
#include <Stats.h>
```

Inheritance diagram for vecthor::DecompressorStats:



Collaboration diagram for vecthor::DecompressorStats:



Public Member Functions

- DecompressorStats (const Config *cfg_ptr)
- void printStats (const std::string &title="", std::ostream &out=std::cout) const
- void collectBenchmarkData () override
- void printBenchmarkData (const std::string &title="") const override

Public Attributes

- unsigned int m_config_bit = 0
- unsigned int m_overall_config_bit = 0

Additional Inherited Members

7.7.1 Constructor & Destructor Documentation

7.7.1.1 DecompressorStats()

```
vecthor::DecompressorStats::DecompressorStats (
            const Config * cfg_ptr )  [inline]
```

7.7.2 Member Function Documentation

7.7.2.1 collectBenchmarkData()

```
void vecthor::DecompressorStats::collectBenchmarkData ( )  [override], [virtual]
```

Implements vecthor::Stats.

7.7.2.2 printBenchmarkData()

```
void vecthor::DecompressorStats::printBenchmarkData (
            const std::string & title = "" ) const  [override], [virtual]
```

Implements vecthor::Stats.

7.7.2.3 printStats()

```
void vecthor::DecompressorStats::printStats (
            const std::string & title = "",
            std::ostream & out = std::cout ) const  [virtual]
```

Implements vecthor::Stats.

7.7.3 Member Data Documentation

7.7.3.1 m_config_bit

```
unsigned int vecthor::DecompressorStats::m_config_bit = 0
```

7.7.3.2 m_overall_config_bit

```
unsigned int vecthor::DecompressorStats::m_overall_config_bit = 0
```

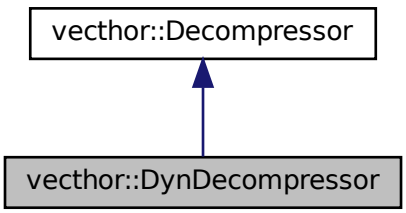
The documentation for this class was generated from the following file:

- src/Stats.h

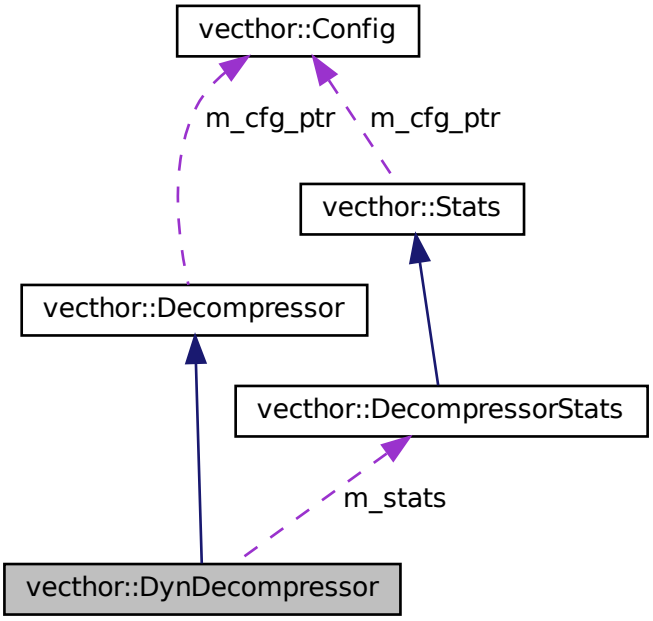
7.8 vecthor::DynDecompressor Class Reference

```
#include <DynDecompressor.h>
```

Inheritance diagram for vecthor::DynDecompressor:



Collaboration diagram for vecthor::DynDecompressor:



Public Member Functions

- DynDecompressor (const Config *cfg_ptr)
- void clear () override
- void determineCDW (BitVecCItr &bv_begin, BitVecCItr &bv_end) override

Private Member Functions

- void assumeCDW (std::string &cdw_repl)
- void preloadConfiguration ()
- void extractData ()
- void plot (std::string plot_name="default_plot")
- void sortFrequencyData ()
- void sortFrequencyData (FrequencyData::iterator begin_it, FrequencyData::iterator end_it)
- unsigned int removeExternals (const std::string &vec_to_str, const FrequencyData::value_type &elem, boost::dynamic_bitset<> &covered)
- void removeExternalsIntersects (const std::string &vec_to_str)
- void removeInternalIntersects (const std::string &vec_to_str)
- unsigned int removeInternals (const std::string &vec_to_str, const FrequencyData::value_type &elem)
- void walk (BitVecCItr bv_start, BitVecCItr bv_end)
- void dumpCoverMap (boost::dynamic_bitset<> &covered) const
- void dumpFreqContainer () const
- bool hasNextInner (unsigned int i) const
- bool hasNextOuter (unsigned int i) const
- bool checkAll (boost::dynamic_bitset<> &covered, std::size_t lb, std::size_t ub)
- void setAll (boost::dynamic_bitset<> &covered, std::size_t lb, std::size_t ub)
- Stats & getStats () override

Private Attributes

- unsigned int m_num_det_cdw
- unsigned int m_lb_freq_inner
- unsigned int m_lb_freq_outer
- unsigned int m_byte_weight
- FrequencyContainer m_freq_container
- FrequencyData m_freq_data
- std::vector< std::string > m_cdw_container
- DecompressorStats m_stats

Additional Inherited Members

7.8.1 Constructor & Destructor Documentation

7.8.1.1 DynDecompressor()

```
vecthor::DynDecompressor::DynDecompressor (
            const Config * cfg_ptr )
```

7.8.2 Member Function Documentation

7.8.2.1 assumeCDW()

```
void vecthor::DynDecompressor::assumeCDW (
            std::string & cdw_repl )  [private]
```

7.8.2.2 checkAll()

```
bool vecthor::DynDecompressor::checkAll (
            boost::dynamic_bitset<> & covered,
            std::size_t lb,
            std::size_t ub )  [inline], [private]
```

7.8.2.3 clear()

```
void vecthor::DynDecompressor::clear ( )  [override], [virtual]
```

Reimplemented from vecthor::Decompressor.

7.8.2.4 determineCDW()

```
void vecthor::DynDecompressor::determineCDW (
            BitVecCItr & bv_begin,
            BitVecCItr & bv_end )  [override], [virtual]
```

Reimplemented from vecthor::Decompressor.

7.8.2.5 dumpCoverMap()

```
void vecthor::DynDecompressor::dumpCoverMap (
            boost::dynamic_bitset<> & covered ) const  [private]
```

7.8.2.6 dumpFreqContainer()

```
void vecthor::DynDecompressor::dumpFreqContainer ( ) const  [private]
```

7.8.2.7 extractData()

```
void vecthor::DynDecompressor::extractData ( )  [private]
```

7.8.2.8 getStats()

```
Stats& vecthor::DynDecompressor::getStats ( )  [inline], [override], [private], [virtual]
```

Reimplemented from vecthor::Decompressor.

7.8.2.9 hasNextInner()

```
bool vecthor::DynDecompressor::hasNextInner (
            unsigned int i ) const  [inline], [private]
```

7.8.2.10 hasNextOuter()

```
bool vecthor::DynDecompressor::hasNextOuter (
            unsigned int i ) const  [inline], [private]
```

7.8.2.11 plot()

```
void vecthor::DynDecompressor::plot (
            std::string plot_name = "default_plot" )  [private]
```

7.8.2.12 preloadConfiguration()

```
void vecthor::DynDecompressor::preloadConfiguration ( )  [private]
```

7.8.2.13 removeExternals()

```
unsigned int vecthor::DynDecompressor::removeExternals (
            const std::string & vec_to_str,
            const FrequencyData::value_type & elem,
            boost::dynamic_bitset<> & covered )  [private]
```

7.8.2.14 removeExternalsIntersects()

```
void vecthor::DynDecompressor::removeExternalsIntersects (
            const std::string & vec_to_str )  [private]
```

7.8.2.15 removeInternalIntersects()

```
void vecthor::DynDecompressor::removeInternalIntersects (
            const std::string & vec_to_str )  [private]
```

7.8.2.16 removeInternals()

```
unsigned int vecthor::DynDecompressor::removeInternals (
            const std::string & vec_to_str,
            const FrequencyData::value_type & elem )  [private]
```

7.8.2.17 setAll()

```
void vecthor::DynDecompressor::setAll (
            boost::dynamic_bitset<> & covered,
            std::size_t lb,
            std::size_t ub )  [inline], [private]
```

7.8.2.18 sortFrequencyData() [1/2]

```
void vecthor::DynDecompressor::sortFrequencyData ( )  [inline], [private]
```

7.8.2.19 sortFrequencyData() [2/2]

```
void vecthor::DynDecompressor::sortFrequencyData (
            FrequencyData::iterator begin_it,
            FrequencyData::iterator end_it )  [inline], [private]
```

7.8.2.20 walk()

```
void vecthor::DynDecompressor::walk (
            BitVecCItr bv_start,
            BitVecCItr bv_end )  [private]
```

7.8.3 Member Data Documentation

7.8.3.1 m_byte_weight

```
unsigned int vecthor::DynDecompressor::m_byte_weight  [private]
```

7.8.3.2 m_cdw_container

```
std::vector<std::string> vecthor::DynDecompressor::m_cdw_container  [private]
```

7.8.3.3 m_freq_container

```
FrequencyContainer vecthor::DynDecompressor::m_freq_container  [private]
```

7.8.3.4 m_freq_data

```
FrequencyData vecthor::DynDecompressor::m_freq_data  [private]
```

7.8.3.5 m_lb_freq_inner

```
unsigned int vecthor::DynDecompressor::m_lb_freq_inner  [private]
```

7.8.3.6 m_lb_freq_outer

```
unsigned int vecthor::DynDecompressor::m_lb_freq_outer  [private]
```

7.8.3.7 m_num_det_cdw

```
unsigned int vecthor::DynDecompressor::m_num_det_cdw  [private]
```

7.8.3.8 m_stats

```
DecompressorStats vecthor::DynDecompressor::m_stats  [private]
```

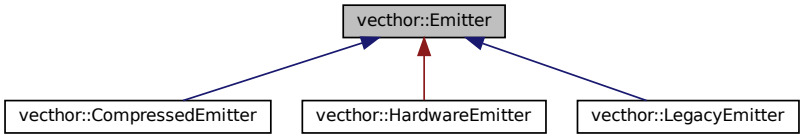
The documentation for this class was generated from the following file:

- src/DynDecompressor.h

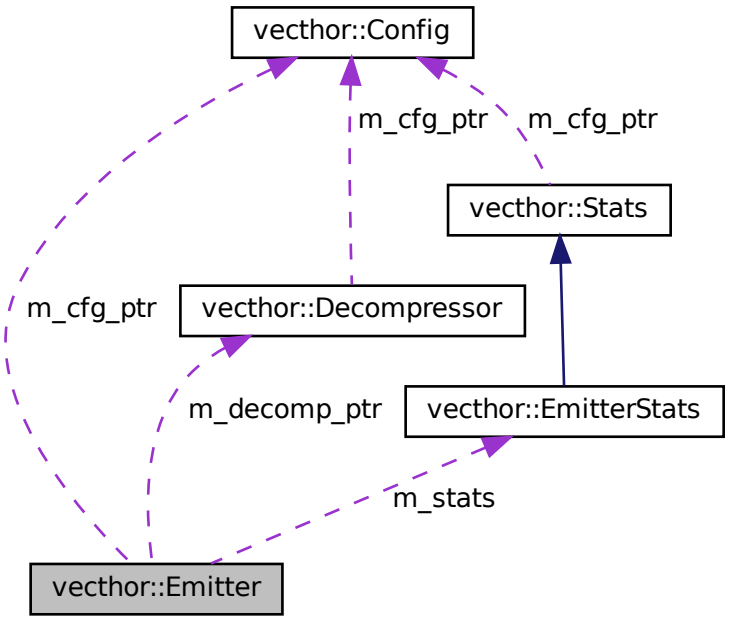
7.9 vecthor::Emitter Class Reference

`#include <Emitter.h>`

Inheritance diagram for vecthor::Emitter:



Collaboration diagram for vecthor::Emitter:



Public Member Functions

- Emitter (const Config *config_ptr, const Decompressor *decomp_ptr=nullptr)
- void writeGoldenFile (const BitVec &bit_vec)
- EmitterStats & getStats ()
- EmitterStats getStats () const

Protected Member Functions

- VALUE getValue (const char &c)
- bool isHigh (VALUE val)
- bool isLow (VALUE val)

Protected Attributes

- const Config * m_cfg_ptr
- const Decompressor * m_decomp_ptr
- EmitterStats m_stats

7.9.1 Constructor & Destructor Documentation

7.9.1.1 Emitter()

```
vecthor::Emitter::Emitter (
            const Config * config_ptr,
            const Decompressor * decomp_ptr = nullptr )  [inline]
```

7.9.2 Member Function Documentation

7.9.2.1 getStats() [1/2]

```
EmitterStats& vecthor::Emitter::getStats ( )  [inline]
```

7.9.2.2 getStats() [2/2]

```
EmitterStats vecthor::Emitter::getStats ( ) const  [inline]
```

7.9.2.3 getValue()

```
VALUE vecthor::Emitter::getValue (
            const char & c )  [protected]
```

7.9.2.4 isHigh()

```
bool vecthor::Emitter::isHigh (
            VALUE val )  [inline], [protected]
```

7.9.2.5 isLow()

```
bool vecthor::Emitter::isLow (
            VALUE val )  [inline], [protected]
```

7.9.2.6 writeGoldenFile()

```
void vecthor::Emitter::writeGoldenFile (
            const BitVec & bit_vec )
```

7.9.3 Member Data Documentation

7.9.3.1 m_cfg_ptr

```
const Config* vecthor::Emitter::m_cfg_ptr  [protected]
```

7.9.3.2 m_decomp_ptr

```
const Decompressor* vecthor::Emitter::m_decomp_ptr  [protected]
```

7.9.3.3 m_stats

```
EmitterStats vecthor::Emitter::m_stats  [protected]
```

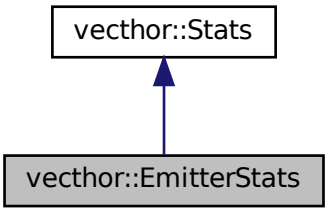
The documentation for this class was generated from the following file:

- src/Emitter.h

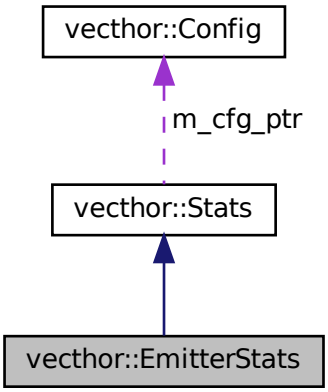
7.10 vecthor::EmitterStats Class Reference

`#include <Stats.h>`

Inheritance diagram for vecthor::EmitterStats:



Collaboration diagram for vecthor::EmitterStats:



Public Member Functions

- EmitterStats (const Config *cfg_ptr)
- void printStats (const std::string &title="", std::ostream &out=std::cout) const
- void collectBenchmarkData () override
- void clear ()
- void printBenchmarkData (const std::string &title="") const override

Public Attributes

- unsigned int m_cycles = 3
- unsigned int m_config_cycles = 3
- unsigned int m_tdi_resets = 0
- unsigned int m_compr_dr = 0
- unsigned int m_compr_exit = 0
- unsigned int m_compre_repeat = 0
- unsigned int m_multi_rep = 0

Additional Inherited Members

7.10.1 Constructor & Destructor Documentation

7.10.1.1 EmitterStats()

```
vecthor::EmitterStats::EmitterStats (
            const Config * cfg_ptr )  [inline]
```

7.10.2 Member Function Documentation

7.10.2.1 clear()

```
void vecthor::EmitterStats::clear ( )
```

7.10.2.2 collectBenchmarkData()

```
void vecthor::EmitterStats::collectBenchmarkData ( )  [override], [virtual]
```

Implements vecthor::Stats.

7.10.2.3 printBenchmarkData()

```
void vecthor::EmitterStats::printBenchmarkData (
            const std::string & title = "" ) const  [override], [virtual]
```

Implements vecthor::Stats.

7.10.2.4 printStats()

```
void vecthor::EmitterStats::printStats (
            const std::string & title = "",
            std::ostream & out = std::cout ) const  [virtual]
```

Implements vecthor::Stats.

7.10.3 Member Data Documentation

7.10.3.1 m_compr_dr

```
unsigned int vecthor::EmitterStats::m_compr_dr = 0
```

7.10.3.2 m_compr_exit

```
unsigned int vecthor::EmitterStats::m_compr_exit = 0
```

7.10.3.3 m_compre_repeat

```
unsigned int vecthor::EmitterStats::m_compre_repeat = 0
```

7.10.3.4 m_config_cycles

```
unsigned int vecthor::EmitterStats::m_config_cycles = 3
```

7.10.3.5 m_cycles

```
unsigned int vecthor::EmitterStats::m_cycles = 3
```

7.10.3.6 m_multi_rep

```
unsigned int vecthor::EmitterStats::m_multi_rep = 0
```

7.10.3.7 m_tdi_resets

```
unsigned int vecthor::EmitterStats::m_tdi_resets = 0
```

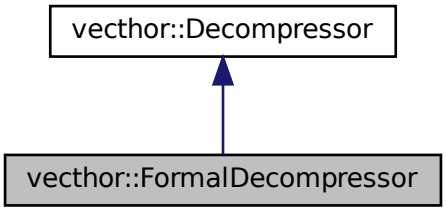
The documentation for this class was generated from the following file:

- src/Stats.h

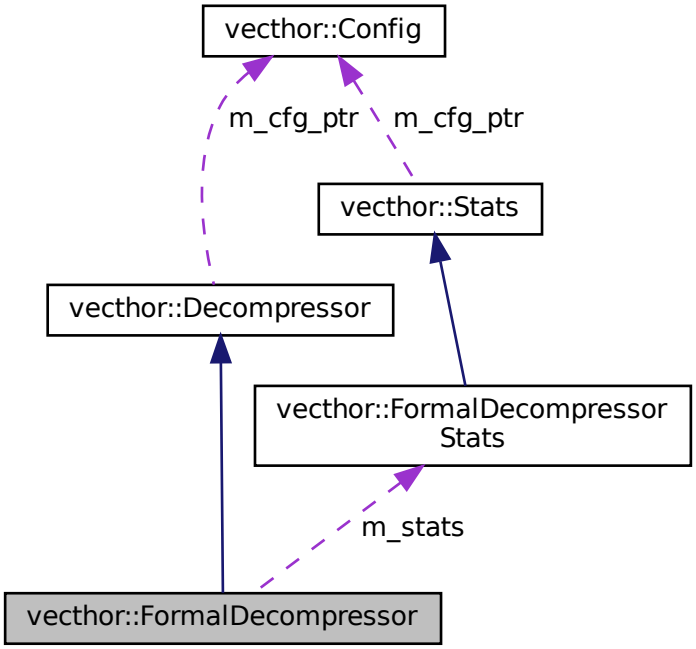
7.11 vecthor::FormalDecompressor Class Reference

```
#include <FormalDecompressor.h>
```

Inheritance diagram for vecthor::FormalDecompressor:



Collaboration diagram for vecthor::FormalDecompressor:



Public Types

- enum MinimizationType {
  MinimizationType::SBI, MinimizationType::SBIMerge, MinimizationType::MergeSBI, MinimizationType::Merge,
  MinimizationType::Codwords }

Public Member Functions

- FormalDecompressor (const Config *cfg_ptr)
- void determineCDW (BitVecCItr &bv_begin, BitVecCItr &bv_end) override
- void processModel (Route &repl_vec)
- void clear () override
- Stats & getStats () override

Private Types

- using VarIndexMap = boost::bimap< Clasp::Var, std::pair< BitVecCItr, BitVecCItr > >
- using VarSBIMap = std::map< unsigned int, Clasp::Var >
- using VarOverlapMap = std::map< unsigned int, std::vector< Clasp::Var > >
- using VarMergeMap = std::map< Clasp::Var, std::pair< Clasp::Var, Clasp::Var > >
- using VarUDWMap = std::map< boost::dynamic_bitset<>, std::tuple< Clasp::Var, Clasp::Var > >
- using VarModelMap = std::map< Clasp::Var, std::pair< Clasp::ValueRep, std::string > >
- using BinaryClauses = std::set< std::pair< Clasp::Literal, Clasp::Literal > >
- using LengthConfig = std::vector< std::pair< unsigned int, VarUDWMap * > >

Private Member Functions

- bool addClause (Clasp::LitVec &lits)
- void addCDWConstraint (unsigned int max_cdws)
- void addUDWConstraint (VarUDWMap::value_type &elem)
- void addSBIConstraint (unsigned int max_sbis)
- void buildOverlappings (BitVecCItr l_start, BitVecCItr l_end, unsigned int length, VarUDWMap &cdw_map)
- void processOverlappings ()
- void enforceCoverage ()
- void calculateCDW (LengthConfig &len_cfg)
- void extractModel ()
- void extractModelUDWValue (const VarUDWMap::value_type &elem)
- void modelMergeAnd (Clasp::Var var_b, Clasp::Var var_c, Clasp::Var merge_var, MinimizationType type)
- void modelMinimization (MinimizationType min_type)
- void processMerges (unsigned int length)
- void processSBIMerges ()
- void prepareBitVec (LengthConfig &len_cfg)
- void processBinary ()
- void processSBIs ()
- void initSolver ()
- void solve ()
- void dump (bool force=false) const
- char dumpValue (Clasp::Var var) const

Static Private Member Functions

- static char valToChar (const Clasp::ValueRep var_value)
- static bool valToBool (const Clasp::ValueRep var_value)

Private Attributes

- std::shared_ptr< Clasp::SharedContext > m_ctx
- std::shared_ptr< Clasp::DecisionHeuristic > m_heu
- std::shared_ptr< Clasp::ModelEnumerator > m_enum
- std::shared_ptr< Clasp::SolveParams > m_params
- std::shared_ptr< Clasp::SolveLimits > m_limits
- std::shared_ptr< Clasp::Var > m_act_var1
- std::shared_ptr< Clasp::Var > m_act_var2
- std::shared_ptr< Clasp::Var > m_act_var3
- Clasp::Solver * m_solver
- Clasp::SharedMinimizeData * m_sdata
- Clasp::MinimizeConstraint * m_constr
- Clasp::LitVec m_assumptions
- VarSBIMap m_sbi_map
- VarIndexMap m_idx_map
- VarOverlapMap m_overlap_map
- VarMergeMap m_merge_map
- VarUDWMap m_udw_map
- VarUDWMap m_byteudw_map
- VarModelMap m_var_model
- Clasp::WeightLitVec m_weighted_merge_lits
- Clasp::WeightLitVec m_weighted_codeword_lits
- Clasp::WeightLitVec m_weighted_sbi_lits
- Clasp::WeightLitVec m_weighted_cdw_4_lits
- Clasp::WeightLitVec m_weighted_cdw_8_lits
- BinaryClauses m_bin_clauses
- FormalDecompressorStats m_stats
- BitVecCItr m_bit_vec_begin
- BitVecCItr m_bit_vec_end

Additional Inherited Members

7.11.1 Member Typedef Documentation

7.11.1.1 BinaryClauses

```
using vecthor::FormalDecompressor::BinaryClauses = std::set<std::pair<Clasp::Literal, Clasp↩
::Literal> >  [private]
```

7.11.1.2 LengthConfig

```
using vecthor::FormalDecompressor::LengthConfig = std::vector<std::pair<unsigned int, VarUDWMap
*> > [private]
```

7.11.1.3 VarIndexMap

```
using vecthor::FormalDecompressor::VarIndexMap = boost::bimap<Clasp::Var, std::pair<BitVecCItr,
BitVecCItr> > [private]
```

7.11.1.4 VarMergeMap

```
using vecthor::FormalDecompressor::VarMergeMap = std::map<Clasp::Var, std::pair<Clasp::Var,
Clasp::Var> > [private]
```

7.11.1.5 VarModelMap

```
using vecthor::FormalDecompressor::VarModelMap = std::map<Clasp::Var, std::pair<Clasp::Value↩
Rep, std::string> > [private]
```

7.11.1.6 VarOverlapMap

```
using vecthor::FormalDecompressor::VarOverlapMap = std::map<unsigned int, std::vector<Clasp↩
::Var> > [private]
```

7.11.1.7 VarSBIMap

```
using vecthor::FormalDecompressor::VarSBIMap = std::map<unsigned int, Clasp::Var> [private]
```

7.11.1.8 VarUDWMap

```
using vecthor::FormalDecompressor::VarUDWMap = std::map<boost::dynamic_bitset<>, std::tuple<Clasp↩
::Var, Clasp::Var> > [private]
```

7.11.2 Member Enumeration Documentation

7.11.2.1 MinimizationType

```
enum vecthor::FormalDecompressor::MinimizationType [strong]
```

Enumerator

SBI	
SBIMerge	
MergeSBI	
Merge	
Codwords	

7.11.3 Constructor & Destructor Documentation

7.11.3.1 FormalDecompressor()

```
vecthor::FormalDecompressor::FormalDecompressor (
            const Config * cfg_ptr )
```

7.11.4 Member Function Documentation

7.11.4.1 addCDWConstraint()

```
void vecthor::FormalDecompressor::addCDWConstraint (
            unsigned int max_cdws )  [private]
```

7.11.4.2 addClause()

```
bool vecthor::FormalDecompressor::addClause (
            Clasp::LitVec & lits )  [private]
```

7.11.4.3 addSBIConstraint()

```
void vecthor::FormalDecompressor::addSBIConstraint (
            unsigned int max_sbis )  [private]
```

7.11.4.4 addUDWConstraint()

```
void vecthor::FormalDecompressor::addUDWConstraint (
            VarUDWMap::value_type & elem )  [private]
```

7.11.4.5 buildOverlappings()

```
void vecthor::FormalDecompressor::buildOverlappings (
            BitVecCItr l_start,
            BitVecCItr l_end,
            unsigned int length,
            VarUDWMap & cdw_map )  [private]
```

7.11.4.6 calculateCDW()

```
void vecthor::FormalDecompressor::calculateCDW (
            LengthConfig & len_cfg )  [private]
```

7.11.4.7 clear()

```
void vecthor::FormalDecompressor::clear ( )  [override], [virtual]
```

Reimplemented from vecthor::Decompressor.

7.11.4.8 determineCDW()

```
void vecthor::FormalDecompressor::determineCDW (
            BitVecCItr & bv_begin,
            BitVecCItr & bv_end )  [override], [virtual]
```

Reimplemented from vecthor::Decompressor.

7.11.4.9 dump()

```
void vecthor::FormalDecompressor::dump (
            bool force = false ) const  [private]
```

7.11.4.10 dumpValue()

```
char vecthor::FormalDecompressor::dumpValue (
            Clasp::Var var ) const  [private]
```

7.11.4.11 enforceCoverage()

```
void vecthor::FormalDecompressor::enforceCoverage ( )  [private]
```

7.11.4.12 extractModel()

```
void vecthor::FormalDecompressor::extractModel ( )  [private]
```

7.11.4.13 extractModelUDWValue()

```
void vecthor::FormalDecompressor::extractModelUDWValue (
            const VarUDWMap::value_type & elem )  [private]
```

7.11.4.14 getStats()

```
Stats& vecthor::FormalDecompressor::getStats ( )  [inline], [override], [virtual]
```

Reimplemented from vecthor::Decompressor.

7.11.4.15 initSolver()

```
void vecthor::FormalDecompressor::initSolver ( )  [private]
```

7.11.4.16 modelMergeAnd()

```
void vecthor::FormalDecompressor::modelMergeAnd (
            Clasp::Var var_b,
            Clasp::Var var_c,
            Clasp::Var merge_var,
            MinimizationType type )  [private]
```

7.11.4.17 modelMinimization()

```
void vecthor::FormalDecompressor::modelMinimization (
            MinimizationType min_type )  [private]
```

7.11.4.18 prepareBitVec()

```
void vecthor::FormalDecompressor::prepareBitVec (
            LengthConfig & len_cfg )  [private]
```

7.11.4.19 processBinary()

```
void vecthor::FormalDecompressor::processBinary ( )  [private]
```

7.11.4.20 processMerges()

```
void vecthor::FormalDecompressor::processMerges (
            unsigned int length )  [private]
```

7.11.4.21 processModel()

```
void vecthor::FormalDecompressor::processModel (
            Route & repl_vec )
```

7.11.4.22 processOverlappings()

```
void vecthor::FormalDecompressor::processOverlappings ( )  [private]
```

7.11.4.23 processSBIMerges()

```
void vecthor::FormalDecompressor::processSBIMerges ( )  [private]
```

7.11.4.24 processSBIs()

```
void vecthor::FormalDecompressor::processSBIs ( )  [private]
```

7.11.4.25 solve()

```
void vecthor::FormalDecompressor::solve ( )  [private]
```

7.11.4.26 valToBool()

```
static bool vecthor::FormalDecompressor::valToBool (
            const Clasp::ValueRep var_value )  [inline], [static], [private]
```

7.11.4.27 valToChar()

```
static char vecthor::FormalDecompressor::valToChar (
            const Clasp::ValueRep var_value )  [inline], [static], [private]
```

7.11.5 Member Data Documentation

7.11.5.1 m_act_var1

```
std::shared_ptr<Clasp::Var> vecthor::FormalDecompressor::m_act_var1  [private]
```

7.11.5.2 m_act_var2

```
std::shared_ptr<Clasp::Var> vecthor::FormalDecompressor::m_act_var2  [private]
```

7.11.5.3 m_act_var3

```
std::shared_ptr<Clasp::Var> vecthor::FormalDecompressor::m_act_var3  [private]
```

7.11.5.4 m_assumptions

```
Clasp::LitVec vecthor::FormalDecompressor::m_assumptions  [private]
```

7.11.5.5 m_bin_clauses

```
BinaryClauses vecthor::FormalDecompressor::m_bin_clauses  [private]
```

7.11.5.6 m_bit_vec_begin

```
BitVecCItr vecthor::FormalDecompressor::m_bit_vec_begin  [private]
```

7.11.5.7 m_bit_vec_end

```
BitVecCItr vecthor::FormalDecompressor::m_bit_vec_end  [private]
```

7.11.5.8 m_byteudw_map

```
VarUDWMap vecthor::FormalDecompressor::m_byteudw_map  [private]
```

7.11.5.9 m_constr

```
Clasp::MinimizeConstraint* vecthor::FormalDecompressor::m_constr  [private]
```

7.11.5.10 m_ctx

```
std::shared_ptr<Clasp::SharedContext> vecthor::FormalDecompressor::m_ctx  [private]
```

7.11.5.11 m_enum

```
std::shared_ptr<Clasp::ModelEnumerator> vecthor::FormalDecompressor::m_enum  [private]
```

7.11.5.12 m_heu

```
std::shared_ptr<Clasp::DecisionHeuristic> vecthor::FormalDecompressor::m_heu  [private]
```

7.11.5.13 m_idx_map

```
VarIndexMap vecthor::FormalDecompressor::m_idx_map  [private]
```

7.11.5.14 m_limits

```
std::shared_ptr<Clasp::SolveLimits> vecthor::FormalDecompressor::m_limits  [private]
```

7.11.5.15 m_merge_map

```
VarMergeMap vecthor::FormalDecompressor::m_merge_map  [private]
```

7.11.5.16 m_overlap_map

```
VarOverlapMap vecthor::FormalDecompressor::m_overlap_map  [private]
```

7.11.5.17 m_params

```
std::shared_ptr<Clasp::SolveParams> vecthor::FormalDecompressor::m_params  [private]
```

7.11.5.18 m_sbi_map

```
VarSBIMap vecthor::FormalDecompressor::m_sbi_map  [private]
```

7.11.5.19 m_sdata

```
Clasp::SharedMinimizeData* vecthor::FormalDecompressor::m_sdata  [private]
```

7.11.5.20 m_solver

```
Clasp::Solver* vecthor::FormalDecompressor::m_solver  [private]
```

7.11.5.21 m_stats

```
FormalDecompressorStats vecthor::FormalDecompressor::m_stats  [private]
```

7.11.5.22 m_udw_map

```
VarUDWMap vecthor::FormalDecompressor::m_udw_map  [private]
```

7.11.5.23 m_var_model

```
VarModelMap vecthor::FormalDecompressor::m_var_model  [private]
```

7.11.5.24 m_weighted_cdw_4_lits

```
Clasp::WeightLitVec vecthor::FormalDecompressor::m_weighted_cdw_4_lits  [private]
```

7.11.5.25 m_weighted_cdw_8_lits

```
Clasp::WeightLitVec vecthor::FormalDecompressor::m_weighted_cdw_8_lits  [private]
```

7.11.5.26 m_weighted_codeword_lits

```
Clasp::WeightLitVec vecthor::FormalDecompressor::m_weighted_codeword_lits  [private]
```

7.11.5.27 m_weighted_merge_lits

```
Clasp::WeightLitVec vecthor::FormalDecompressor::m_weighted_merge_lits  [private]
```

7.11.5.28 m_weighted_sbi_lits

```
Clasp::WeightLitVec vecthor::FormalDecompressor::m_weighted_sbi_lits  [private]
```

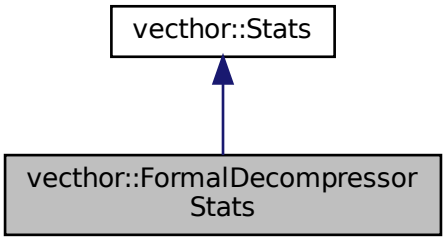
The documentation for this class was generated from the following file:

- src/FormalDecompressor.h

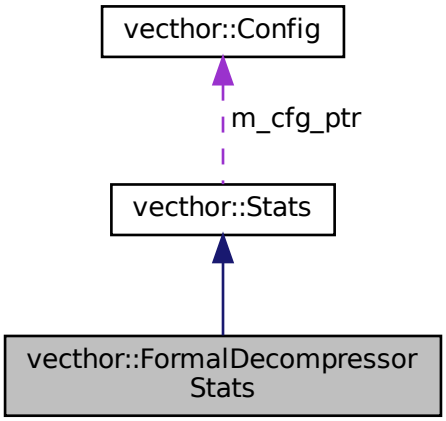
7.12 vecthor::FormalDecompressorStats Class Reference

```
#include <Stats.h>
```

Inheritance diagram for vecthor::FormalDecompressorStats:



Collaboration diagram for vecthor::FormalDecompressorStats:



Public Member Functions

- FormalDecompressorStats (const Config *cfg_ptr)
- void printStats (const std::string &title="", std::ostream &out=std::cout) const
- void collectBenchmarkData () override
- void printBenchmarkData (const std::string &title="") const override

Public Attributes

- unsigned int m_restarts = 0
- unsigned int m_ccs = 0
- unsigned int m_bin_clauses = 0
- unsigned int m_constraints = 0
- unsigned int m_vars = 0
- unsigned int m_det_cdws = 0
- unsigned int m_det_static_cdws = 0
- unsigned int m_det_sbis = 0
- unsigned int m_merge_vars = 0
- unsigned int m_act_merges = 0
- unsigned int m_config_bit = 0
- unsigned int m_overall_config_bit = 0

Additional Inherited Members

7.12.1 Constructor & Destructor Documentation

7.12.1.1 FormalDecompressorStats()

```
vecthor::FormalDecompressorStats::FormalDecompressorStats (
            const Config * cfg_ptr )  [inline]
```

7.12.2 Member Function Documentation

7.12.2.1 collectBenchmarkData()

```
void vecthor::FormalDecompressorStats::collectBenchmarkData ( )  [override], [virtual]
```

Implements vecthor::Stats.

7.12.2.2 printBenchmarkData()

```
void vecthor::FormalDecompressorStats::printBenchmarkData (
            const std::string & title = "" ) const  [override], [virtual]
```

Implements vecthor::Stats.

7.12.2.3 printStats()

```
void vecthor::FormalDecompressorStats::printStats (
            const std::string & title = "",
            std::ostream & out = std::cout ) const  [virtual]
```

Implements vecthor::Stats.

7.12.3 Member Data Documentation

7.12.3.1 m_act_merges

```
unsigned int vecthor::FormalDecompressorStats::m_act_merges = 0
```

7.12.3.2 m_bin_clauses

```
unsigned int vecthor::FormalDecompressorStats::m_bin_clauses = 0
```

7.12.3.3 m_ccs

```
unsigned int vecthor::FormalDecompressorStats::m_ccs = 0
```

7.12.3.4 m_config_bit

```
unsigned int vecthor::FormalDecompressorStats::m_config_bit = 0
```

7.12.3.5 m_constraints

```
unsigned int vecthor::FormalDecompressorStats::m_constraints = 0
```

7.12.3.6 m_det_cdws

```
unsigned int vecthor::FormalDecompressorStats::m_det_cdws = 0
```

7.12.3.7 m_det_sbis

```
unsigned int vecthor::FormalDecompressorStats::m_det_sbis = 0
```

7.12.3.8 m_det_static_cdws

```
unsigned int vecthor::FormalDecompressorStats::m_det_static_cdws = 0
```

7.12.3.9 m_merge_vars

```
unsigned int vecthor::FormalDecompressorStats::m_merge_vars = 0
```

7.12.3.10 m_overall_config_bit

```
unsigned int vecthor::FormalDecompressorStats::m_overall_config_bit = 0
```

7.12.3.11 m_restarts

```
unsigned int vecthor::FormalDecompressorStats::m_restarts = 0
```

7.12.3.12 m_vars

```
unsigned int vecthor::FormalDecompressorStats::m_vars = 0
```

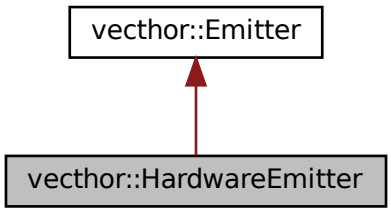
The documentation for this class was generated from the following file:

- src/Stats.h

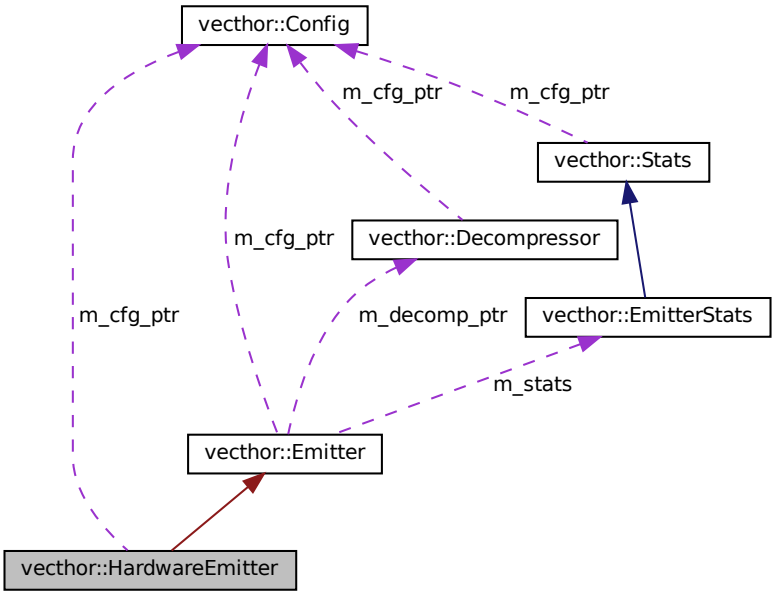
7.13 vecthor::HardwareEmitter Class Reference

```
#include <HardwareEmitter.h>
```

Inheritance diagram for vecthor::HardwareEmitter:



Collaboration diagram for vecthor::HardwareEmitter:



Public Member Functions

- HardwareEmitter (const Config *config, const Signals *signals)
- ~HardwareEmitter ()
- void init ()
- void finalize ()
- void operator() ()

Private Member Functions

- void dump ()
- void writeValues ()

Private Attributes

- bool m_tdi_state
- bool m_tms_state
- bool m_tck_state
- const Signals * m_signal_ptr
- const Config * m_cfg_ptr
- pthread_t * m_pthread

7.13.1 Constructor & Destructor Documentation

7.13.1.1 HardwareEmitter()

```
vecthor::HardwareEmitter::HardwareEmitter (
            const Config * config,
            const Signals * signals )
```

7.13.1.2 ~HardwareEmitter()

```
vecthor::HardwareEmitter::~HardwareEmitter ( )
```

7.13.2 Member Function Documentation

7.13.2.1 dump()

```
void vecthor::HardwareEmitter::dump ( )  [private]
```

7.13.2.2 finalize()

```
void vecthor::HardwareEmitter::finalize ( )
```

7.13.2.3 init()

```
void vecthor::HardwareEmitter::init ( )
```

7.13.2.4 operator()()

```
void vecthor::HardwareEmitter::operator() ( )
```

7.13.2.5 writeValues()

```
void vecthor::HardwareEmitter::writeValues ( )  [private]
```

7.13.3 Member Data Documentation

7.13.3.1 m_cfg_ptr

```
const Config* vecthor::HardwareEmitter::m_cfg_ptr  [private]
```

7.13.3.2 m_pthread

```
pthread_t* vecthor::HardwareEmitter::m_pthread  [private]
```

7.13.3.3 m_signal_ptr

```
const Signals* vecthor::HardwareEmitter::m_signal_ptr  [private]
```

7.13.3.4 m_tck_state

```
bool vecthor::HardwareEmitter::m_tck_state  [private]
```

7.13.3.5 m_tdi_state

```
bool vecthor::HardwareEmitter::m_tdi_state  [private]
```

7.13.3.6 m_tms_state

```
bool vecthor::HardwareEmitter::m_tms_state  [private]
```

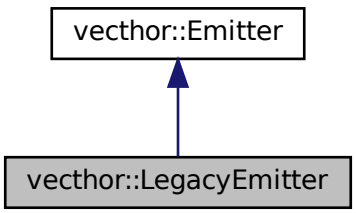
The documentation for this class was generated from the following file:

- src/HardwareEmitter.h

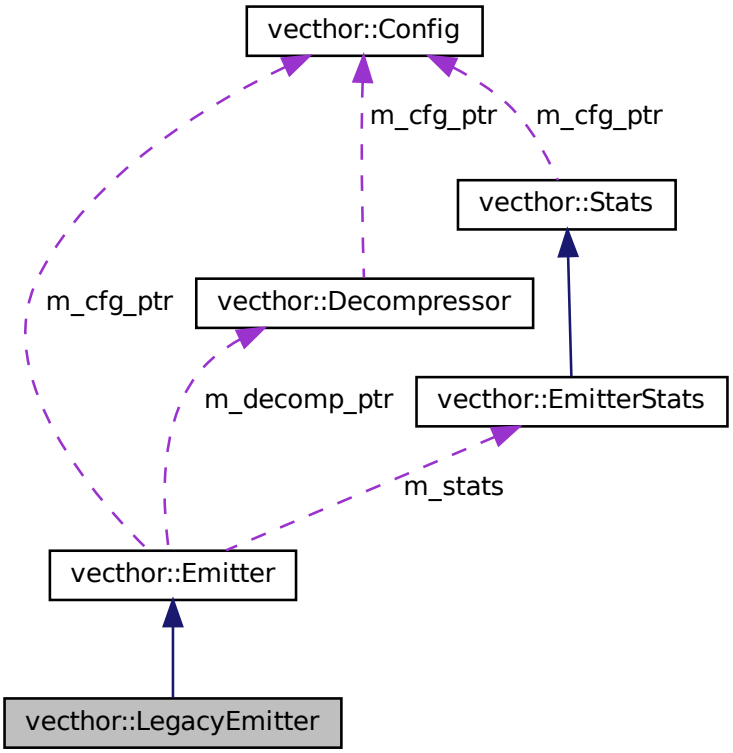
7.14 vecthor::LegacyEmitter Class Reference

```
#include <Emitter.h>
```

Inheritance diagram for vecthor::LegacyEmitter:



Collaboration diagram for vecthor::LegacyEmitter:



Public Member Functions

- LegacyEmitter (const Config ∗config)
- void operator() (const BitVec &bit_vec)

Private Member Functions

- void writeJTAG (const BitVec &bit_vec, std::ostream ∗stream)

Additional Inherited Members

7.14.1 Constructor & Destructor Documentation

7.14.1.1 LegacyEmitter()

```
vecthor::LegacyEmitter::LegacyEmitter (
            const Config * config )  [inline]
```

7.14.2 Member Function Documentation

7.14.2.1 operator()()

```
void vecthor::LegacyEmitter::operator() (
            const BitVec & bit_vec )
```

7.14.2.2 writeJTAG()

```
void vecthor::LegacyEmitter::writeJTAG (
            const BitVec & bit_vec,
            std::ostream * stream )  [private]
```

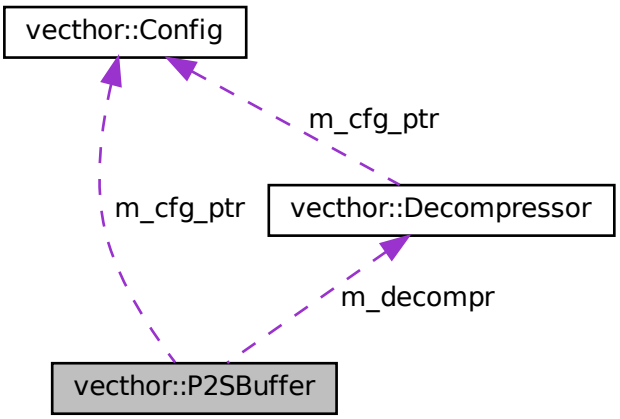
The documentation for this class was generated from the following file:

- src/Emitter.h

7.15 vecthor::P2SBuffer Class Reference

```
#include <P2SBuffer.h>
```

Collaboration diagram for vecthor::P2SBuffer:



Public Types

- using DataBuffer = std::vector< int >
- using DataCollector = std::vector< std::pair< unsigned int, unsigned int > >

Public Member Functions

- P2SBuffer (const Config *config, const Decompressor *decompr)
- const DataCollector & getCollector () const
- unsigned int processRoute (Route &route, std::size_t max_cycles)

Private Member Functions

- void simulateDataSink (unsigned int delay=0)
- unsigned int determineDelay (unsigned int delay, std::size_t max_cycles)
- void dumpBuffer (bool force=false) const
- void dumpCollector (bool force=false) const
- void plot (std::string plot_name) const

Private Attributes

- unsigned int m_max_buf
- DataBuffer m_buf
- DataCollector m_collector
- const Decompressor * m_decompr
- const Config * m_cfg_ptr

7.15.1 Member Typedef Documentation

7.15.1.1 DataBuffer

```
using vecthor::P2SBuffer::DataBuffer = std::vector<int>
```

7.15.1.2 DataCollector

```
using vecthor::P2SBuffer::DataCollector = std::vector<std::pair<unsigned int, unsigned int> >
```

7.15.2 Constructor & Destructor Documentation

7.15.2.1 P2SBuffer()

```
vecthor::P2SBuffer::P2SBuffer (
            const Config * config,
            const Decompressor * decompr )  [inline]
```

7.15.3 Member Function Documentation

7.15.3.1 determineDelay()

```
unsigned int vecthor::P2SBuffer::determineDelay (
            unsigned int delay,
            std::size_t max_cycles )  [private]
```

7.15.3.2 dumpBuffer()

```
void vecthor::P2SBuffer::dumpBuffer (
            bool force = false ) const  [private]
```

7.15.3.3 dumpCollector()

```
void vecthor::P2SBuffer::dumpCollector (
            bool force = false ) const  [private]
```

7.15.3.4 getCollector()

```
const DataCollector& vecthor::P2SBuffer::getCollector ( ) const  [inline]
```

7.15.3.5 plot()

```
void vecthor::P2SBuffer::plot (
            std::string plot_name ) const  [private]
```

7.15.3.6 processRoute()

```
unsigned int vecthor::P2SBuffer::processRoute (
            Route & route,
            std::size_t max_cycles )
```

7.15.3.7 simulateDataSink()

```
void vecthor::P2SBuffer::simulateDataSink (
            unsigned int delay = 0 )  [private]
```

7.15.4 Member Data Documentation

7.15.4.1 m_buf

```
DataBuffer vecthor::P2SBuffer::m_buf  [private]
```

7.15.4.2 m_cfg_ptr

```
const Config* vecthor::P2SBuffer::m_cfg_ptr  [private]
```

7.15.4.3 m_collector

```
DataCollector vecthor::P2SBuffer::m_collector  [private]
```

7.15.4.4 m_decompr

```
const Decompressor* vecthor::P2SBuffer::m_decompr  [private]
```

7.15.4.5 m_max_buf

```
unsigned int vecthor::P2SBuffer::m_max_buf  [private]
```

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

- src/P2SBuffer.h

7.16 vecthor::Plotter Class Reference

`#include <Plotter.h>`

Public Types

- enum PlotType { ScatterPlot, HistoPlot, Plot3D }
- enum CFGATTR {
  CFGATTR::NAME, CFGATTR::DATAFILENAME, CFGATTR::XLABEL, CFGATTR::YLABEL,
  CFGATTR::ZLABEL, CFGATTR::XTICS, CFGATTR::YTICS, CFGATTR::ZTICS,
  CFGATTR::XRANGE, CFGATTR::YRANGE, CFGATTR::ZRANGE, CFGATTR::TITLE,
  CFGATTR::TERMINAL, CFGATTR::OUTPUT, CFGATTR::SIZE, CFGATTR::GRID,
  CFGATTR::DESCRIPTION, CFGATTR::DESCRIPTIONPOS, CFGATTR::STYLEDATA, CFGATTR::STYLEFILL,
  CFGATTR::FONT, CFGATTR::FONTSIZE, CFGATTR::LABEL, CFGATTR::LABELPOS,
  CFGATTR::AZIMUT, CFGATTR::ELEVATION, CFGATTR::PLOT, CFGATTR::SPLOT,
  CFGATTR::GRAPHTITLE, CFGATTR::USING, CFGATTR::EVERY, CFGATTR::SYMBOL }
- using ConfigMap = std::map< CFGATTR, std::string >
- using ConfigEntry = ConfigMap::value_type
- using ConfigLookupMap = std::map< CFGATTR, std::string >

Public Member Functions

- Plotter ()
- template<PlotType T>
  void initTypeConfig ()
- int writeConfig (ConfigMap &user_settings)
- template<typename S , typename T >
  int writeData (std::vector< S > &fst_values, std::vector< T > &snd_values)

Private Member Functions

- bool isSkippable (CFGATTR attr) const
- bool isEmpty (CFGATTR attr) const
- bool isQuoted (CFGATTR attr) const
- std::string getAttribute (CFGATTR attr) const
- void generatePlotCfg (std::ofstream &out) const
- void generatePlot (std::ofstream &out) const
- int getConfigLength ()
- void initConfig ()
- template<typename S , typename T , typename U >
  int writeData (std::vector< T > &fst_values, std::vector< S > &snd_values, std::vector< U > &rd_values)

Private Attributes

- ConfigMap m_cfg
- ConfigLookupMap m_lookup
- std::vector< std::string > m_raw_config
- int m_counter = 0
- bool m_init = false

7.16.1 Member Typedef Documentation

7.16.1.1 ConfigEntry

```
using vecthor::Plotter::ConfigEntry = ConfigMap::value_type
```

7.16.1.2 ConfigLookupMap

```
using vecthor::Plotter::ConfigLookupMap = std::map<CFGATTR, std::string>
```

7.16.1.3 ConfigMap

```
using vecthor::Plotter::ConfigMap = std::map<CFGATTR, std::string>
```

7.16.2 Member Enumeration Documentation

7.16.2.1 CFGATTR

```
enum vecthor::Plotter::CFGATTR  [strong]
```

Enumerator

NAME	
DATAFILENAME	
XLABEL	
YLABEL	
ZLABEL	
XTICS	
YTICS	
ZTICS	
XRANGE	
YRANGE	
ZRANGE	
TITLE	
TERMINAL	
OUTPUT	
SIZE	
GRID	
DESCRIPTION	

Enumerator

DESCRIPTIONPOS	
STYLEDATA	
STYLEFILL	
FONT	
FONTSIZE	
LABEL	
LABELPOS	
AZIMUT	
ELEVATION	
PLOT	
SPLOT	
GRAPHTITLE	
USING	
EVERY	
SYMBOL	

7.16.2.2 PlotType

```
enum vecthor::Plotter::PlotType
```

Enumerator

ScatterPlot	
HistoPlot	
Plot3D	

7.16.3 Constructor & Destructor Documentation

7.16.3.1 Plotter()

```
vecthor::Plotter::Plotter ( )
```

7.16.4 Member Function Documentation

7.16.4.1 generatePlot()

```
void vecthor::Plotter::generatePlot (
            std::ofstream & out ) const  [private]
```

7.16.4.2 generatePlotCfg()

```
void vecthor::Plotter::generatePlotCfg (
            std::ofstream & out ) const  [private]
```

7.16.4.3 getAttribute()

```
std::string vecthor::Plotter::getAttribute (
            CFGATTR attr ) const  [private]
```

7.16.4.4 getConfigLength()

```
int vecthor::Plotter::getConfigLength ( )  [inline], [private]
```

7.16.4.5 initConfig()

```
void vecthor::Plotter::initConfig ( )  [private]
```

7.16.4.6 initTypeConfig()

```
template<PlotType T>
void vecthor::Plotter::initTypeConfig ( )
```

7.16.4.7 isEmpty()

```
bool vecthor::Plotter::isEmpty (
            CFGATTR attr ) const  [private]
```

7.16.4.8 isQuoted()

```
bool vecthor::Plotter::isQuoted (
            CFGATTR attr ) const  [private]
```

7.16.4.9 isSkippable()

```
bool vecthor::Plotter::isSkippable (
            CFGATTR attr ) const  [private]
```

7.16.4.10 writeConfig()

```
int vecthor::Plotter::writeConfig (
            ConfigMap & user_settings )
```

7.16.4.11 writeData() [1/2]

```
template<typename S , typename T >
int vecthor::Plotter::writeData (
            std::vector< S > & fst_values,
            std::vector< T > & snd_values )  [inline]
```

7.16.4.12 writeData() [2/2]

```
template<typename S , typename T , typename U >
int vecthor::Plotter::writeData (
            std::vector< T > & fst_values,
            std::vector< S > & snd_values,
            std::vector< U > & rd_values )  [inline], [private]
```

7.16.5 Member Data Documentation

7.16.5.1 m_cfg

```
ConfigMap vecthor::Plotter::m_cfg  [private]
```

7.16.5.2 m_counter

```
int vecthor::Plotter::m_counter = 0  [private]
```

7.16.5.3 m_init

```
bool vecthor::Plotter::m_init = false  [private]
```

7.16.5.4 m_lookup

```
ConfigLookupMap vecthor::Plotter::m_lookup  [private]
```

7.16.5.5 m_raw_config

```
std::vector<std::string> vecthor::Plotter::m_raw_config  [private]
```

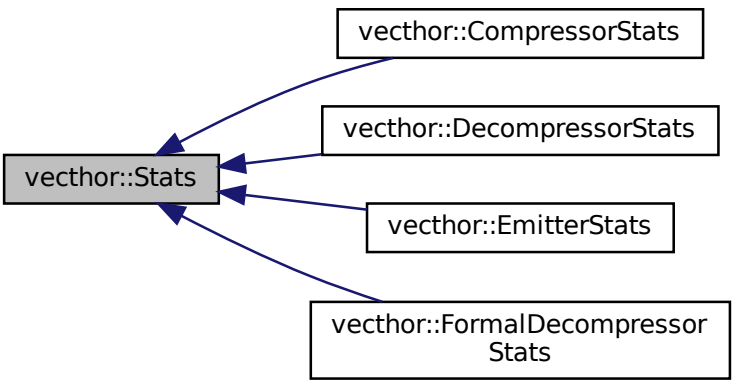
The documentation for this class was generated from the following file:

- src/Plotter.h

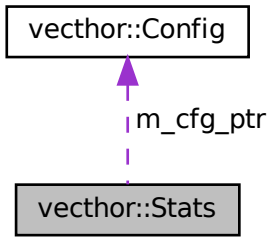
7.17 vecthor::Stats Class Reference

```
#include <Stats.h>
```

Inheritance diagram for vecthor::Stats:



Collaboration diagram for vecthor::Stats:



Public Member Functions

- Stats (const Config ∗cfg_ptr)
- virtual void printStats (const std::string &title="", std::ostream &out=std::cout) const =0
- virtual void collectBenchmarkData ()=0
- virtual void printBenchmarkData (const std::string &title="") const =0
- std::string separatorToken ()

Protected Attributes

- const Config ∗ m_cfg_ptr
- std::map< std::string, void ∗ > m_stats_db

7.17.1 Constructor & Destructor Documentation

7.17.1.1 Stats()

```
vecthor::Stats::Stats (
            const Config * cfg_ptr )  [inline]
```

7.17.2 Member Function Documentation

7.17.2.1 collectBenchmarkData()

```
virtual void vecthor::Stats::collectBenchmarkData ( )  [pure virtual]
```

Implemented in vecthor::FormalDecompressorStats, vecthor::DecompressorStats, vecthor::EmitterStats, and vecthor::CompressorStats.

7.17.2.2 printBenchmarkData()

```
virtual void vecthor::Stats::printBenchmarkData (
            const std::string & title = "" ) const  [pure virtual]
```

Implemented in vecthor::FormalDecompressorStats, vecthor::DecompressorStats, vecthor::EmitterStats, and vecthor::CompressorStats.

7.17.2.3 printStats()

```
virtual void vecthor::Stats::printStats (
            const std::string & title = "",
            std::ostream & out = std::cout ) const  [pure virtual]
```

Implemented in vecthor::FormalDecompressorStats, vecthor::DecompressorStats, vecthor::EmitterStats, and vecthor::CompressorStats.

7.17.2.4 separatorToken()

```
std::string vecthor::Stats::separatorToken ( )  [inline]
```

7.17.3 Member Data Documentation

7.17.3.1 m_cfg_ptr

```
const Config* vecthor::Stats::m_cfg_ptr  [protected]
```

7.17.3.2 m_stats_db

```
std::map<std::string, void *> vecthor::Stats::m_stats_db  [protected]
```

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

- src/Stats.h

7.18 vecthor::TDRGen Class Reference

```
#include <TDRGen.h>
```

Static Public Member Functions

- static const BitVec generateRTDR (unsigned int num_bytes, bool allow_x=false)

Static Private Member Functions

- static unsigned int generateRBit (unsigned int &ctr, bool allow_x)

7.18.1 Member Function Documentation

7.18.1.1 generateRBit()

```
static unsigned int vecthor::TDRGen::generateRBit (
            unsigned int & ctr,
            bool allow_x )  [static], [private]
```

7.18.1.2 generateRTDR()

```
static const BitVec vecthor::TDRGen::generateRTDR (
            unsigned int num_bytes,
            bool allow_x = false )  [static]
```

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

- src/TDRGen.h

7.19 vecthor::TDRReader Class Reference

```
#include <TDRReader.h>
```

Static Public Member Functions

- static const BitVec readTDR (const std::string &ext_file_name)
- static const BitVec readHexTDR (const std::string &ext_file_name)

7.19.1 Member Function Documentation

7.19.1.1 readHexTDR()

```
static const BitVec vecthor::TDRReader::readHexTDR (
            const std::string & ext_file_name )  [static]
```

7.19.1.2 readTDR()

```
static const BitVec vecthor::TDRReader::readTDR (
            const std::string & ext_file_name )  [static]
```

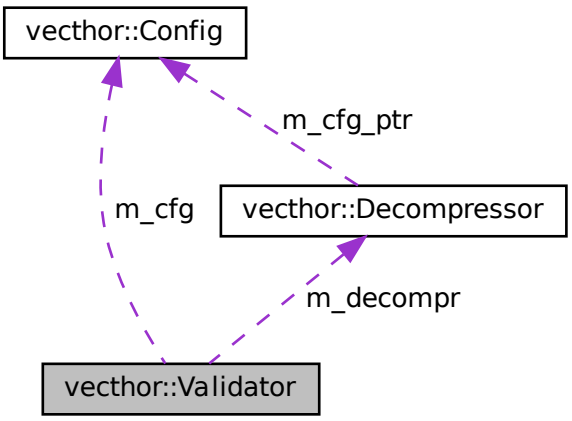
The documentation for this class was generated from the following file:

- src/TDRReader.h

7.20 vecthor::Validator Class Reference

```
#include <Validator.h>
```

Collaboration diagram for vecthor::Validator:



Public Member Functions

- Validator (BitVec *bit_vec, Config *cfg, Decompressor *decompr)
- void storeReplace (const Decompressor::UDWStringMap &udws)
- void storeChunk (const Route &route)
- bool validate ()

Private Attributes

- BitVec * m_bit_vec_golden
- Config * m_cfg
- Decompressor * m_decompr
- std::vector< Decompressor::CDWStringMap > m_udw_map_vec
- std::vector< std::vector< CDW > > m_bit_vec_chunk
- std::ofstream m_valid_file

7.20.1 Constructor & Destructor Documentation

7.20.1.1 Validator()

```
vecthor::Validator::Validator (
            BitVec * bit_vec,
            Config * cfg,
            Decompressor * decompr )
```

7.20.2 Member Function Documentation

7.20.2.1 storeChunk()

```
void vecthor::Validator::storeChunk (
            const Route & route )
```

7.20.2.2 storeReplace()

```
void vecthor::Validator::storeReplace (
            const Decompressor::UDWStringMap & udws )
```

7.20.2.3 validate()

```
bool vecthor::Validator::validate ( )
```

7.20.3 Member Data Documentation

7.20.3.1 m_bit_vec_chunk

```
std::vector<std::vector<CDW> > vecthor::Validator::m_bit_vec_chunk  [private]
```

7.20.3.2 m_bit_vec_golden

```
BitVec* vecthor::Validator::m_bit_vec_golden  [private]
```

7.20.3.3 m_cfg

```
Config* vecthor::Validator::m_cfg  [private]
```

7.20.3.4 m_decompr

```
Decompressor* vecthor::Validator::m_decompr  [private]
```

7.20.3.5 m_udw_map_vec

```
std::vector<Decompressor::CDWStringMap> vecthor::Validator::m_udw_map_vec  [private]
```

7.20.3.6 m_valid_file

```
std::ofstream vecthor::Validator::m_valid_file  [private]
```

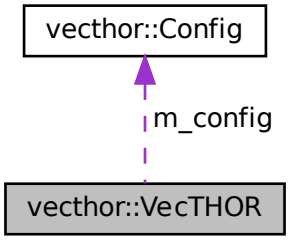
The documentation for this class was generated from the following file:

- src/Validator.h

7.21 vecthor::VecTHOR Class Reference

`#include <VecTHOR.h>`

Collaboration diagram for vecthor::VecTHOR:



Public Member Functions

- Config & getConfig ()
- void init ()
- void finalize ()
- bool prepare ()
- bool run ()
- void reset ()
- void validate ()

Private Attributes

- Config m_config
- std::string m_run_name
- BitVec m_bit_vec
- CompressorPtr m_compressor = nullptr
- DecompressorPtr m_decompressor = nullptr
- CompressedEmitterPtr m_emitter = nullptr
- ValidatorPtr m_validator = nullptr

7.21.1 Member Function Documentation

7.21.1.1 finalize()

`void vecthor::VecTHOR::finalize ( )`

7.21.1.2 getConfig()

```
Config& vecthor::VecTHOR::getConfig ( )  [inline]
```

7.21.1.3 init()

```
void vecthor::VecTHOR::init ( )
```

7.21.1.4 prepare()

```
bool vecthor::VecTHOR::prepare ( )
```

7.21.1.5 reset()

```
void vecthor::VecTHOR::reset ( )
```

7.21.1.6 run()

```
bool vecthor::VecTHOR::run ( )
```

7.21.1.7 validate()

```
void vecthor::VecTHOR::validate ( )
```

7.21.2 Member Data Documentation

7.21.2.1 m_bit_vec

```
BitVec vecthor::VecTHOR::m_bit_vec  [private]
```

7.21.2.2 m_compressor

```
CompressorPtr vecthor::VecTHOR::m_compressor = nullptr  [private]
```

7.21.2.3 m_config

```
Config vecthor::VecTHOR::m_config  [private]
```

7.21.2.4 m_decompressor

```
DecompressorPtr vecthor::VecTHOR::m_decompressor = nullptr  [private]
```

7.21.2.5 m_emitter

```
CompressedEmitterPtr vecthor::VecTHOR::m_emitter = nullptr  [private]
```

7.21.2.6 m_run_name

```
std::string vecthor::VecTHOR::m_run_name  [private]
```

7.21.2.7 m_validator

```
ValidatorPtr vecthor::VecTHOR::m_validator = nullptr  [private]
```

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

- src/VecTHOR.h

Chapter 8

File Documentation

8.1 build-huhn-linux/CMakeCache.txt File Reference

Variables

- BUILD_TESTING __pad0__
- iostreams
- program_options
- regex

8.1.1 Variable Documentation

8.1.1.1 __pad0__

```
BUILD_TESTING __pad0__
```

8.1.1.2 iostreams

```
iostreams
```

8.1.1.3 program_options

```
program_options
```

8.1.1.4 regex

```
regex
```

8.2 build-huhn-linux/CMakeFiles/3.16.4/CompilerIdC/CMakeCCompilerId.c File Reference

Macros

- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X) #X
- #define STRINGIFY(X) STRINGIFY_HELPER(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define C_DIALECT

Functions

- int main (int argc, char *argv[ ])

Variables

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char * info_language_dialect_default

8.2.1 Macro Definition Documentation

8.2.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

8.2.1.2 C_DIALECT

```
#define C_DIALECT
```

8.2.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```

8.2.1.4 DEC

```
#define DEC(
              n )
```

Value:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

8.2.1.5 HEX

```
#define HEX(
              n )
```

Value:

```
  ('0' + ((n)>>28 & 0xF)), \
  ('0' + ((n)>>24 & 0xF)), \
  ('0' + ((n)>>20 & 0xF)), \
  ('0' + ((n)>>16 & 0xF)), \
  ('0' + ((n)>>12 & 0xF)), \
  ('0' + ((n)>>8  & 0xF)), \
  ('0' + ((n)>>4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```

8.2.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```

8.2.1.7 STRINGIFY

```
#define STRINGIFY(
              X ) STRINGIFY_HELPER(X)
```

8.2.1.8 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X ) #X
```

8.2.2 Function Documentation

8.2.2.1 main()

```
int main (
             int argc,
             char * argv[] )
```

8.2.3 Variable Documentation

8.2.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

8.2.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

8.2.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```

Initial value:

```
=
  "INFO" ":" "dialect_default[" C_DIALECT "]"
```

8.2.3.4 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

8.3 build-huhn-linux/CMakeFiles/3.16.4/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

Macros

- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X) #X
- #define STRINGIFY(X) STRINGIFY_HELPER(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define CXX_STD __cplusplus

Functions

- int main (int argc, char *argv[])

Variables

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char * info_language_dialect_default

8.3.1 Macro Definition Documentation

8.3.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

8.3.1.2 COMPILER_ID

```
#define COMPILER_ID ""
```

8.3.1.3 CXX_STD

```
#define CXX_STD __cplusplus
```

8.3.1.4 DEC

```
#define DEC(
              n )
```

Value:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

8.3.1.5 HEX

```
#define HEX(
              n )
```

Value:

```
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```

8.3.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```

8.3.1.7 STRINGIFY

```
#define STRINGIFY(
              X ) STRINGIFY_HELPER(X)
```

8.3.1.8 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X ) #X
```

8.3.2 Function Documentation

8.3.2.1 main()

```
int main (
             int argc,
             char * argv[] )
```

8.3.3 Variable Documentation

8.3.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

8.3.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

8.3.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```

Initial value:

```
= "INFO" ":" "dialect_default["
  "98"
"]"
```

8.3.3.4 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

8.4 build-huhn-linux/CMakeFiles/CMakeRuleHashes.txt File Reference

8.5 build-huhn-linux/CMakeFiles/TargetDirectories.txt File Reference

8.6 build-huhn-linux/lib/clasp-3.1.4/CMakeFiles/libclasp.dir/link.txt File Reference

8.7 build-huhn-linux/src/CMakeFiles/VecTHOR.dir/link.txt File Reference

8.8 build-huhn-linux/src/CMakeFiles/VecTHOR_LIB.dir/link.txt File Reference

8.9 build-huhn-linux/test/struct_dynmergecompressed/plot_stage1_↩ data.txt File Reference

8.10 build-huhn-linux/test/struct_dynmergecompressed/plot_stage2_↩ data.txt File Reference

8.11 build-huhn-linux/Testing/Temporary/CTestCostData.txt File Reference

8.12 CMakeConfig.txt File Reference

Functions

- set (CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11 -fPIC -Wall -Wextra -pedantic -W -Wunused-variable -Wunused-parameter -Wunused-function -Wunused -Wno-system-headers -Wno-deprecated -Woverloaded-virtual -Wwrite-strings ${CMAKE_CXX_OPT_FLAGS} ") set(CMAKE_CXX_↩ FLAGS_DEBUG "-g") set(CMAKE_LIBRARY_PATH $
- lib if (${CMAKE_CXX_COMPILER} MATCHES "(.∗)clang\\+\\+") message(STATUS "Changing CXX_↩ FLAGS for using clang++") set(CMAKE_CXX_FLAGS "-Wall") set(CMAKE_CXX_FLAGS_DEBUG "-g") set(CMAKE_CXX_FLAGS_MINSIZEREL "-Os -DNDEBUG") set(CMAKE_CXX_FLAGS_RELEASE "-O2 -DNDEBUG") set(CMAKE_CXX_FLAGS_RELWITHDEBINFO "-O2 -g") endif() add_definitions(-DWITH_T↩ HREADS=0) add_definitions("-DBOOST_NO_CXX11_SCOPED_ENUMS") set(TEST_DUMP_DIR "dump") set(TEST_SUB_DIR "out") set(TEST_FV_SUB_DIR "fv") set(TEST_RESULT_DIR "results") set(DESIGN↩ _INC_DIR_SC $
- src verilog_compressed set (DESIGN_INC_DIR_SMC ${CMAKE_CURRENT_SOURCE_DIR}/../src/verilog↩ _mergecompressed) set(DESIGN_INC_DIR_SSC $
- src verilog_dynmergecompressed set (DESIGN_INC_DIR_SBSC ${CMAKE_CURRENT_SOURCE_D↩ IR}/../src/verilog_bufferdynmergecompressed) set(VECTHOR_CONFIG_FILE "default.conf") if(NOT CMA↩ KE_BUILD_TYPE) set(CMAKE_BUILD_TYPE Release) endif() if($
- MATCHES Debug message (STATUS "Using debug compiler flags") set(CMAKE_CXX_FLAGS_DEBUG "$

8.12.1 Function Documentation

8.12.1.1 if()

```
lib if (
              ${CMAKE_CXX_COMPILER} MATCHES "(.*)clang\\+\\+" )  [pure virtual]
```

8.12.1.2 message()

```
MATCHES Debug message (
             STATUS "Using debug compiler flags"  )
```

8.12.1.3 set() [1/3]

```
set ( )
```

8.12.1.4 set() [2/3]

```
src verilog_dynmergecompressed set (
             DESIGN_INC_DIR_SBSC ${CMAKE_CURRENT_SOURCE_DIR}/../src/ verilog_bufferdynmergecompressed
)
```

8.12.1.5 set() [3/3]

```
src verilog_compressed set (
             DESIGN_INC_DIR_SMC ${CMAKE_CURRENT_SOURCE_DIR}/../src/ verilog_mergecompressed )
```

8.13 CMakeLists.txt File Reference

Functions

- project (VecTHOR) cmake_minimum_required(VERSION 2.8) include(CTest) include(CMakeConfig.txt) if($
- MATCHES x86_64 set (SYSTEM_TYPE "m64") else() set(SYSTEM_TYPE "m32") endif() option(USE_C↩ LASP "Build clasp [req. for formal approach]" on) option(USE_PI "Build hardware emitter for RaspPi" off) option(USE_STIL "Build stil parser infrastructure" off) option(RUN_TESTS "Run test file" on) option(FAST_↩ TESTING "Run only fast tests" on) option(VSIM_BATCH "Run VSIM in batch mode" on) find_package(Boost "1.41.0" REQUIRED COMPONENTS filesystem system iostreams regex program_options) find_library(YA↩ ML_CPP yaml-cpp) if($
- add_definitions (-DPI=true) set(PI_LIB "wiringPi") endif() if($
- message (WARNING "-- Could NOT find YAML") add_subdirectory(lib/yaml-cpp-master) include_↩ directories(lib/yaml-cpp-master/include) endif() if($
- if (${CLASP_INCLUDE_DIR}) message(WARNING "-- Could NOT find CLASP") else() add_↩ subdirectory(lib/clasp-3.1.4) include_directories(lib/clasp-3.1.4/) endif() endif() if($
- endif () if($
- message (FATAL_ERROR "-- Could NOT find Boost") else() include_directories($

8.13.1 Function Documentation

8.13.1.1 add_definitions()

```
add_definitions (
             - DPI = true )
```

8.13.1.2 endif()

```
endif ( )
```

8.13.1.3 if()

```
if (
              ${CLASP_INCLUDE_DIR} )
```

8.13.1.4 message() [1/2]

```
message (
             FATAL_ERROR "-- Could NOT find Boost"  )
```

8.13.1.5 message() [2/2]

```
message (
             WARNING "-- Could NOT find YAML"  )
```

8.13.1.6 project()

```
project (
             VecTHOR  )
```

8.13.1.7 set()

```
MATCHES x86_64 set (
             SYSTEM_TYPE "m64"  )
```

8.14 src/CMakeLists.txt File Reference

Functions

- set (SRC TypeDefs.h Config.C Config.h CompressorCore.C CompressorCore.h Compressor.C Compressor.↩ h Decompressor.C Decompressor.h DynDecompressor.C DynDecompressor.h FormalDecompressor.C FormalDecompressor.h Emitter.C Emitter.h P2SBuffer.h P2SBuffer.C Plotter.C Plotter.h Stats.C Stats.↩ h TDRGen.C TDRGen.h TDRReader.C TDRReader.h TypeDefs.h Utils.C Utils.h Validator.C Validator.h VecTHOR.C VecTHOR.h) add_library($

8.14.1 Function Documentation

8.14.1.1 set()

```
set (
             SRC TypeDefs.h Config.C Config.h CompressorCore.C CompressorCore.h Compressor.↩
C Compressor.h Decompressor.C Decompressor.h DynDecompressor.C DynDecompressor.h Formal↩
Decompressor.C FormalDecompressor.h Emitter.C Emitter.h P2SBuffer.h P2SBuffer.C Plotter.↩
C Plotter.h Stats.C Stats.h TDRGen.C TDRGen.h TDRReader.C TDRReader.h TypeDefs.h Utils.↩
C Utils.h Validator.C Validator.h VecTHOR.C VecTHOR. h )
```

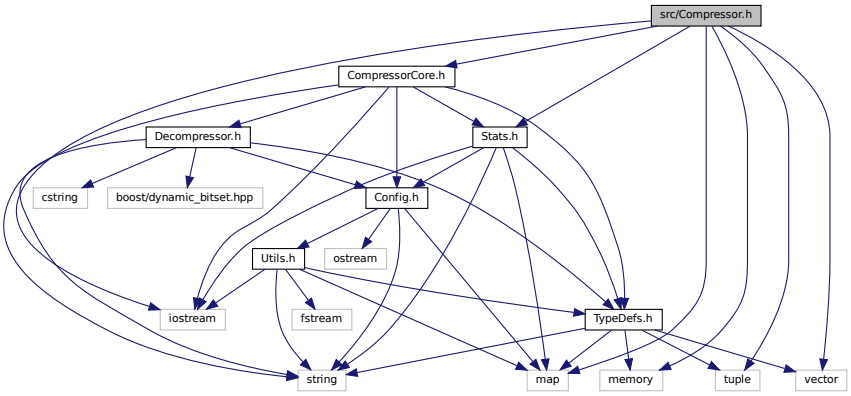
8.15 README.md File Reference

8.16 src/Compressor.h File Reference

```
#include <CompressorCore.h>
#include <Stats.h>
#include <iostream>
#include <map>
#include <memory>
#include <tuple>
```

```
#include <vector>
```

Include dependency graph for Compressor.h:



Classes

- class vecthor::Compressor

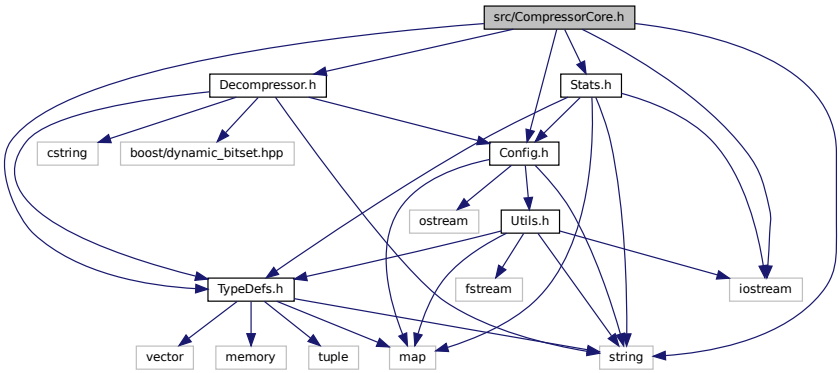
Namespaces

- vecthor

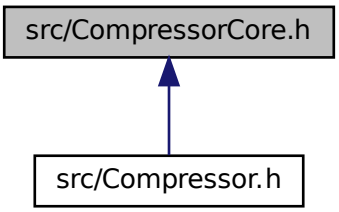
8.17 src/CompressorCore.h File Reference

```
#include <Config.h>
#include <Decompressor.h>
#include <Stats.h>
#include <TypeDefs.h>
#include <iostream>
#include <string>
```

Include dependency graph for CompressorCore.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vecthor::CompressorCore

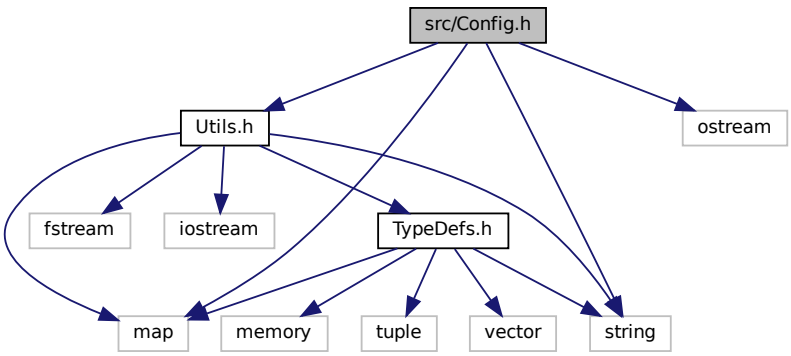
Namespaces

- vecthor

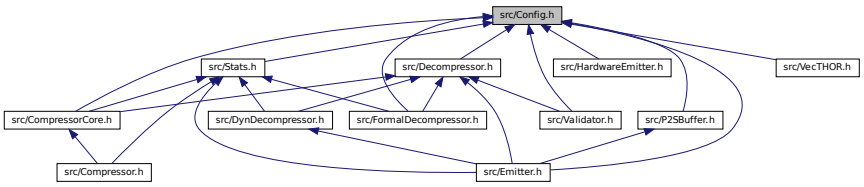
8.18 src/Config.h File Reference

```
#include <Utils.h>
#include <map>
#include <ostream>
#include <string>
```

Include dependency graph for Config.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vecthor::Config

Namespaces

- vecthor

Macros

- #define CONFIG_H

Enumerations

- enum vecthor::CFG : u_int8_t {
  vecthor::CFG::UNSUPPORTED, vecthor::CFG::MERGING, vecthor::CFG::DYNAMIC, vecthor::CFG::HEUR_INNER_FREQ,
  vecthor::CFG::HEUR_OUTER_FREQ, vecthor::CFG::HEUR_WEIGHT, vecthor::CFG::HEUR_PERMUTE,
  vecthor::CFG::SAT,
  vecthor::CFG::SAT_SEC, vecthor::CFG::SAT_CONFL, vecthor::CFG::SAT_RESTART, vecthor::CFG::MAX_CDWS,
  vecthor::CFG::PART_SIZE, vecthor::CFG::EXT_CDWS, vecthor::CFG::VERBOSE, vecthor::CFG::DEBUG,
  vecthor::CFG::STATS, vecthor::CFG::BENCHMARK, vecthor::CFG::PLOT, vecthor::CFG::HEX,
  vecthor::CFG::P2S_BUFFER, vecthor::CFG::USE_EXT_FILE, vecthor::CFG::USE_CONF_FILE, vecthor::CFG::GEN_LEGACY
  vecthor::CFG::GEN_COMPRESSED, vecthor::CFG::GEN_GOLDEN, vecthor::CFG::HW_EMIT, vecthor::CFG::ALLOW_X,
  vecthor::CFG::VALIDATE }
- enum vecthor::FILE : u_int8_t {
  vecthor::FILE::UNSUPPORTED, vecthor::FILE::LEGACY_PREFIX, vecthor::FILE::LEGACY_SUFFIX,
  vecthor::FILE::DYNCOMPRESSED_INFIX,
  vecthor::FILE::DYNCOMPRESSED_PRELOAD, vecthor::FILE::COMPRESSED_PREFIX, vecthor::FILE::COMPRESSED_SUF
  vecthor::FILE::COMPRESSED_FILE,
  vecthor::FILE::CONFIG_FILE, vecthor::FILE::RESYNC_FILE, vecthor::FILE::EXT_FILE, vecthor::FILE::LEGACY_FILE,
  vecthor::FILE::GOLDEN_FILE, vecthor::FILE::VALIDATION_FILE }

8.18.1 Macro Definition Documentation

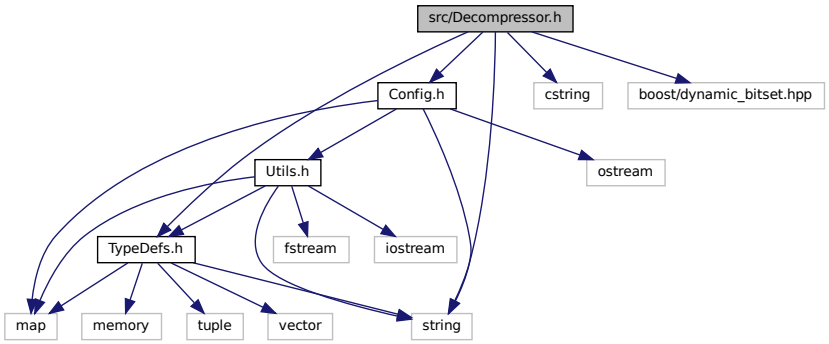
8.18.1.1 CONFIG_H

```
#define CONFIG_H
```

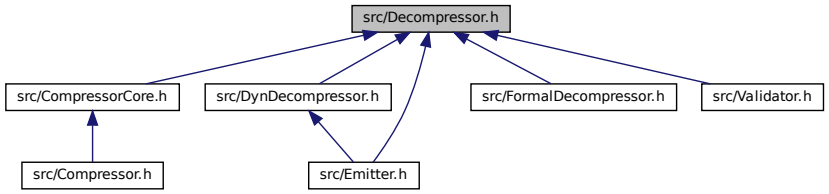
8.19 src/Decompressor.h File Reference

```
#include <Config.h>
#include <TypeDefs.h>
#include <cstring>
#include <string>
#include <boost/dynamic_bitset.hpp>
```

Include dependency graph for Decompressor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vecthor::Decompressor

Namespaces

- vecthor

Macros

- #define DECOMPRESSOR_H

8.19.1 Macro Definition Documentation

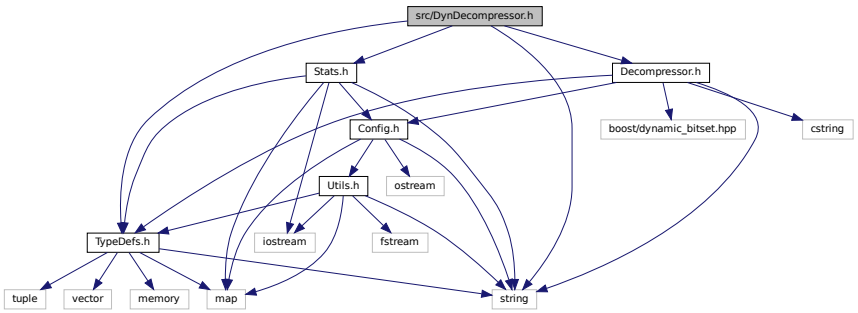
8.19.1.1 DECOMPRESSOR_H

```
#define DECOMPRESSOR_H
```

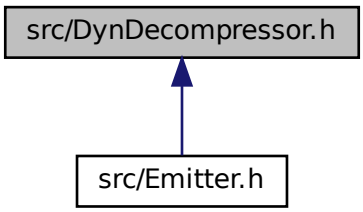
8.20 src/DynDecompressor.h File Reference

```
#include <Decompressor.h>
#include <Stats.h>
#include <TypeDefs.h>
#include <string>
```

Include dependency graph for DynDecompressor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vecthor::DynDecompressor

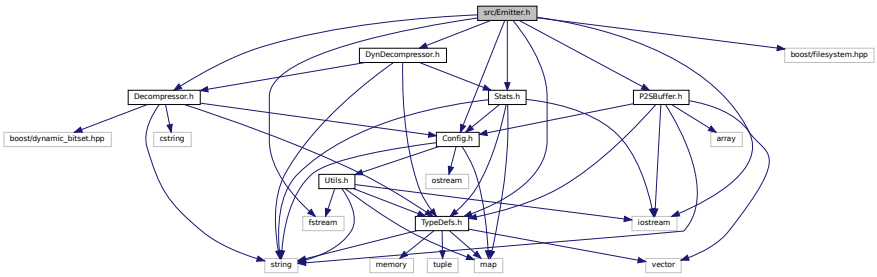
Namespaces

- vecthor

8.21 src/Emitter.h File Reference

```
#include <Config.h>
#include <Decompressor.h>
#include <DynDecompressor.h>
#include <P2SBuffer.h>
#include <Stats.h>
#include <TypeDefs.h>
#include <fstream>
#include <iostream>
#include <boost/filesystem.hpp>
```

Include dependency graph for Emitter.h:



Classes

- class vecthor::Emitter
- class vecthor::LegacyEmitter
- class vecthor::CompressedEmitter

Namespaces

- vecthor

Macros

- #define TESTBENCHEMITTER_H

8.21.1 Macro Definition Documentation

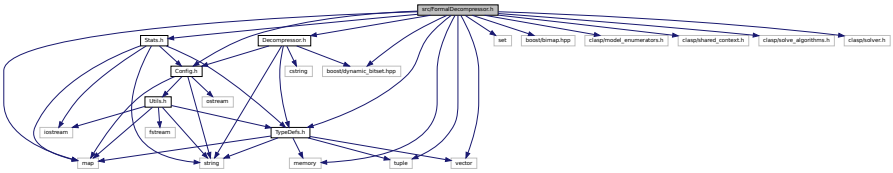
8.21.1.1 TESTBENCHEMITTER_H

```
#define TESTBENCHEMITTER_H
```

8.22 src/FormalDecompressor.h File Reference

```
#include <Config.h>
#include <Decompressor.h>
#include <Stats.h>
#include <TypeDefs.h>
#include <map>
#include <memory>
#include <set>
#include <tuple>
#include <vector>
#include <boost/bimap.hpp>
#include <boost/dynamic_bitset.hpp>
#include "clasp/model_enumerators.h"
#include "clasp/shared_context.h"
#include "clasp/solve_algorithms.h"
#include "clasp/solver.h"
```

Include dependency graph for FormalDecompressor.h:



Classes

- class vecthor::FormalDecompressor

Namespaces

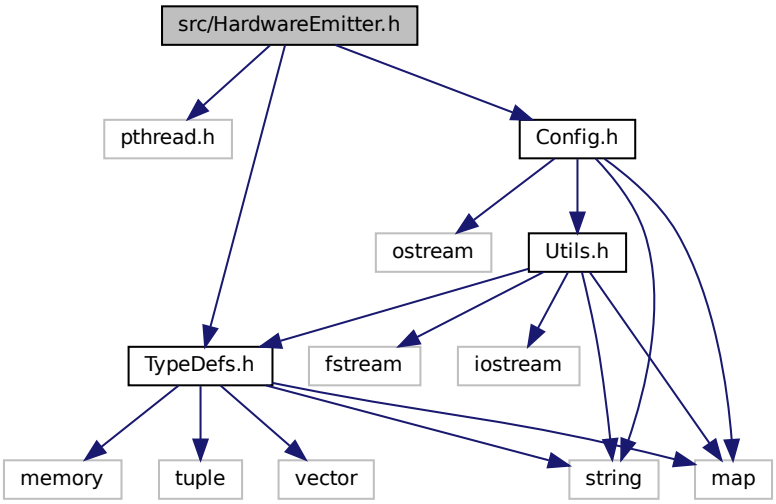
- vecthor

8.23 src/HardwareEmitter.h File Reference

```
#include <pthread.h>
#include "Config.h"
```

```
#include "TypeDefs.h"
```

Include dependency graph for HardwareEmitter.h:



Classes

- class vecthor::HardwareEmitter

Namespaces

- vecthor

Macros

- #define TDI_WIPI_PIN 7
- #define TMS_WIPI_PIN 0
- #define TCK_WIPI_PIN 2
- #define CLOCK_PERIOD 1

8.23.1 Macro Definition Documentation

8.23.1.1 CLOCK_PERIOD

```
#define CLOCK_PERIOD 1
```

8.23.1.2 TCK_WIPI_PIN

```
#define TCK_WIPI_PIN 2
```

8.23.1.3 TDI_WIPI_PIN

```
#define TDI_WIPI_PIN 7
```

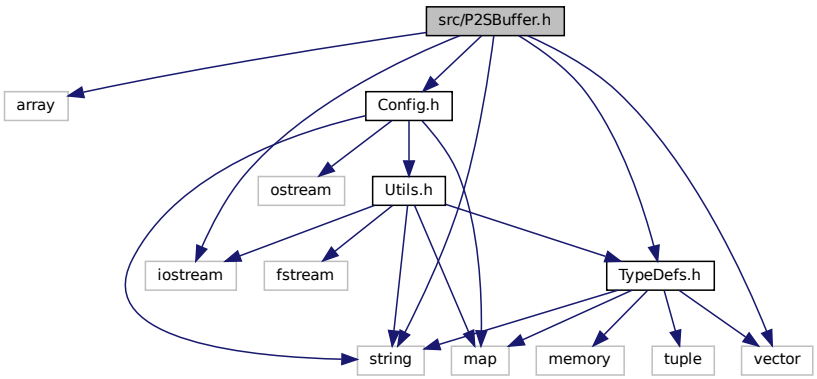
8.23.1.4 TMS_WIPI_PIN

```
#define TMS_WIPI_PIN 0
```

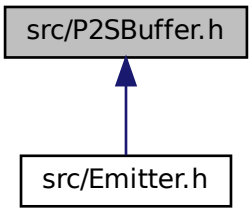
8.24 src/P2SBuffer.h File Reference

```
#include <array>
#include <iostream>
#include <string>
#include <vector>
#include <Config.h>
#include <TypeDefs.h>
```

Include dependency graph for P2SBuffer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vecthor::P2SBuffer

Namespaces

- vecthor

Macros

- #define MAX_BUFFER 100

8.24.1 Macro Definition Documentation

8.24.1.1 MAX_BUFFER

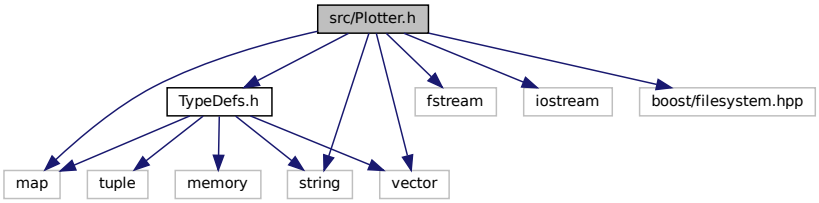
```
#define MAX_BUFFER 100
```

8.25 src/Plotter.h File Reference

```
#include <TypeDefs.h>
#include <fstream>
#include <iostream>
#include <map>
#include <string>
#include <vector>
```

```
#include <boost/filesystem.hpp>
```

Include dependency graph for Plotter.h:



Classes

- class vecthor::Plotter

Namespaces

- vecthor

Macros

- #define MAX_PLOTS 10
- #define MAX_TITLE_LENGTH 40

8.25.1 Macro Definition Documentation

8.25.1.1 MAX_PLOTS

```
#define MAX_PLOTS 10
```

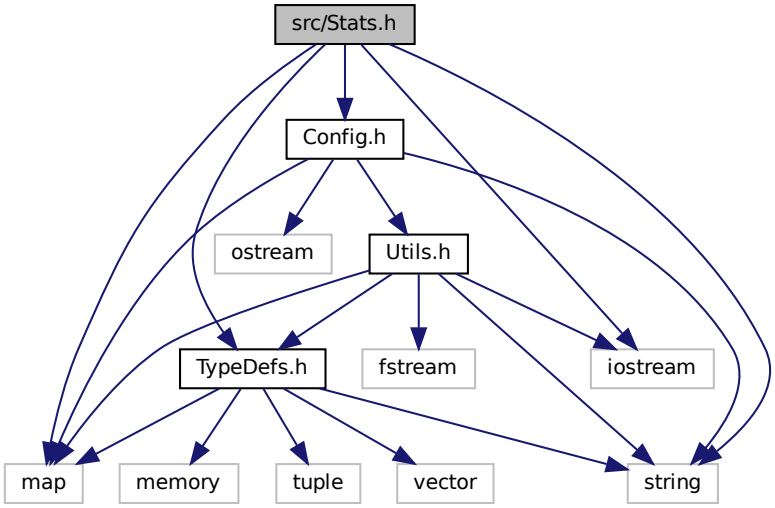
8.25.1.2 MAX_TITLE_LENGTH

```
#define MAX_TITLE_LENGTH 40
```

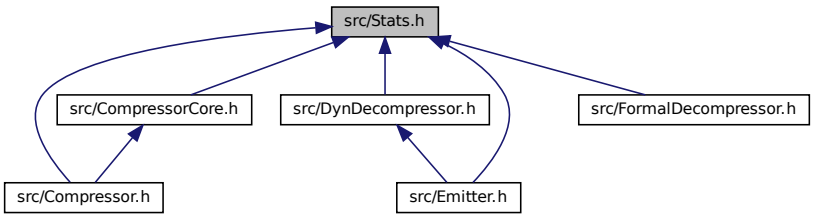
8.26 src/Stats.h File Reference

```
#include <Config.h>
#include <TypeDefs.h>
#include <iostream>
#include <map>
#include <string>
```

Include dependency graph for Stats.h:



This graph shows which files directly or indirectly include this file:



Classes

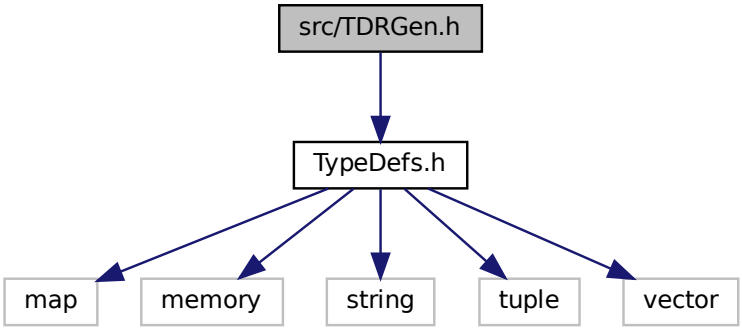
- class vecthor::Stats
- class vecthor::CompressorStats
- class vecthor::EmitterStats
- class vecthor::DecompressorStats
- class vecthor::FormalDecompressorStats

Namespaces

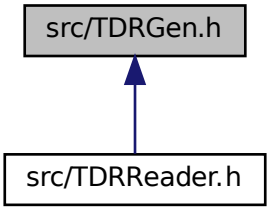
- vecthor

8.27 src/TDRGen.h File Reference

```
#include <TypeDefs.h>
```
Include dependency graph for TDRGen.h:



This graph shows which files directly or indirectly include this file:



Classes

- class vecthor::TDRGen

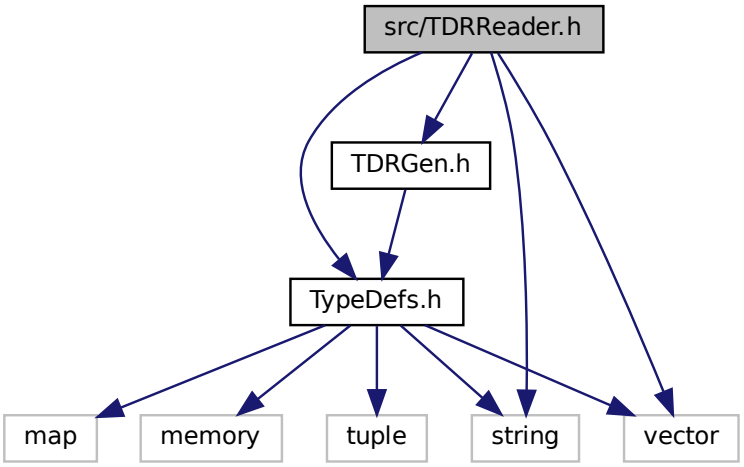
Namespaces

- vecthor

8.28 src/TDRReader.h File Reference

```
#include <TDRGen.h>
#include <TypeDefs.h>
#include <string>
#include <vector>
```

Include dependency graph for TDRReader.h:



Classes

- class vecthor::TDRReader

Namespaces

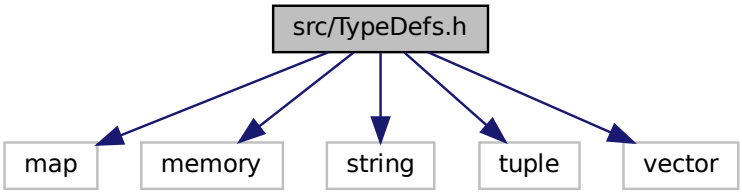
- vecthor

8.29 src/TypeDefs.h File Reference

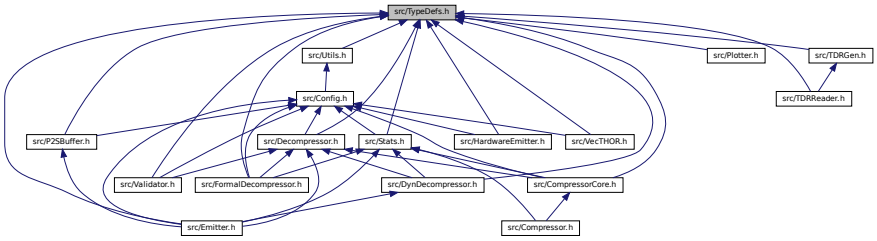
```
#include <map>
#include <memory>
#include <string>
#include <tuple>
```

```
#include <vector>
```

Include dependency graph for TypeDefs.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- vecthor

Macros

- #define CYCLE_TIME 10
- #define BUFFER_CTR_SIZE 12
- #define USE_EXT_CDWS false

Typedefs

- using vecthor::BitVec = std::vector< u_int8_t >
- using vecthor::BitVecCPtr = std::shared_ptr< const BitVec >
- using vecthor::BitVecItr = BitVec::iterator
- using vecthor::BitVecCItr = BitVec::const_iterator
- using vecthor::Replacement = std::tuple< CDW, BitVecCItr, BitVecCItr, short >
- using vecthor::ReplacementPtr = std::shared_ptr< Replacement >
- using vecthor::CDWMap = std::map< BitVecCItr, std::map< BitVecCItr, ReplacementPtr > >
- using vecthor::CDWMapItem = std::pair< BitVecCItr, std::map< BitVecCItr, ReplacementPtr > >
- using vecthor::Edge = std::tuple< short, short, ReplacementPtr >
- using vecthor::Edges = std::vector< Edge >
- using vecthor::Route = std::vector< ReplacementPtr >
- using vecthor::FrequencyContainer = std::map< std::string, unsigned int >
- using vecthor::FrequencyData = std::vector< std::pair< unsigned int, std::string > >
- using vecthor::Signals = std::vector< std::tuple< VALUE, VALUE > >

Enumerations

- enum vecthor::VALUE : uint8_t {
  vecthor::VALUE::LOW = 0, vecthor::VALUE::HIGH = 1, vecthor::VALUE::NOP = 2, vecthor::VALUE::INIT = 3,
  vecthor::VALUE::UNSUPPORTED = 4 }
- enum vecthor::CDW {
  vecthor::CDW::NONE, vecthor::CDW::XXX, vecthor::CDW::HXX, vecthor::CDW::LXX,
  vecthor::CDW::LLX, vecthor::CDW::LHX, vecthor::CDW::HLX, vecthor::CDW::HHX,
  vecthor::CDW::LLL, vecthor::CDW::LLH, vecthor::CDW::LHL, vecthor::CDW::LHH,
  vecthor::CDW::HLL, vecthor::CDW::HLH, vecthor::CDW::HHL, vecthor::CDW::HHH }
- enum vecthor::REPL_FIELD : uint8_t { vecthor::REPL_FIELD::CDW, vecthor::REPL_FIELD::START,
  vecthor::REPL_FIELD::END, vecthor::REPL_FIELD::WEIGHT }

8.29.1 Macro Definition Documentation

8.29.1.1 BUFFER_CTR_SIZE

```
#define BUFFER_CTR_SIZE 12
```

8.29.1.2 CYCLE_TIME

```
#define CYCLE_TIME 10
```

8.29.1.3 USE_EXT_CDWS

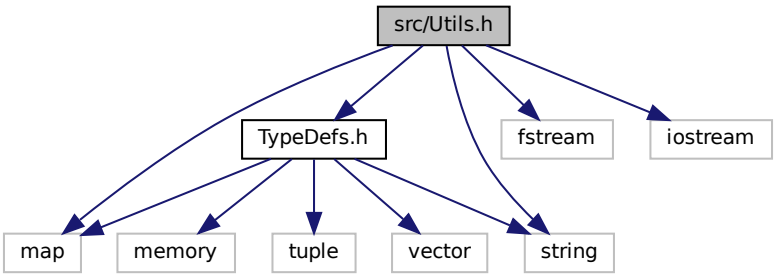
```
#define USE_EXT_CDWS false
```

8.30 src/Utils.h File Reference

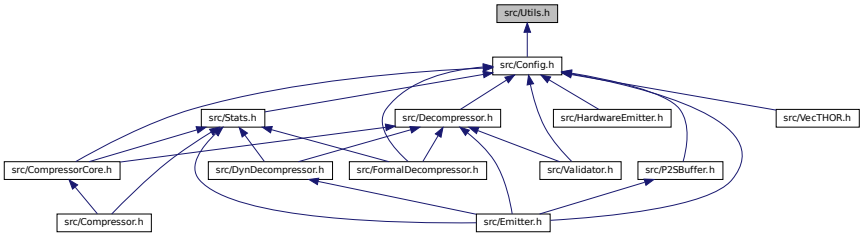
```
#include <TypeDefs.h>
#include <fstream>
#include <iostream>
#include <map>
```

```
#include <string>
```

Include dependency graph for Utils.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- vecthor

Macros

- #define UTILS_H

Functions

- unsigned int vecthor::countBitVecX (BitVecCItr start, BitVecCItr end)
- std::string vecthor::serializeBitVec (BitVecCItr start, BitVecCItr end)
- void vecthor::writeBitVec (const BitVec &bit_vec, std::ostream *stream=&std::cout, bool header=true)
- void vecthor::writeBitVec (BitVecCItr start, BitVecCItr end, std::ostream *stream=&std::cout)

8.30.1 Macro Definition Documentation

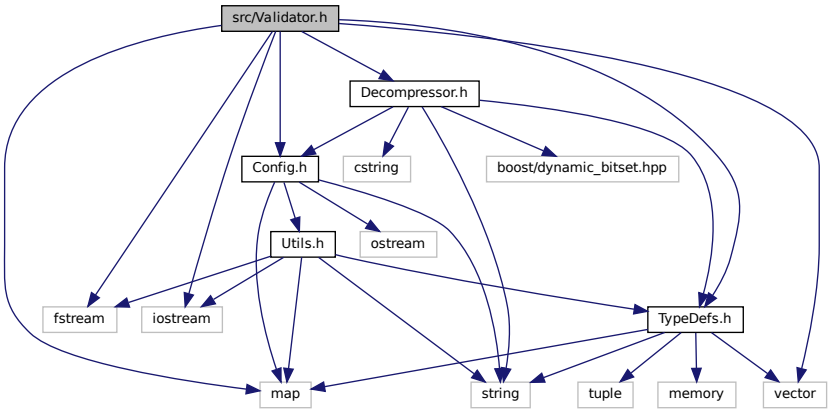
8.30.1.1 UTILS_H

```
#define UTILS_H
```

8.31 src/Validator.h File Reference

```
#include <fstream>
#include <iostream>
#include <map>
#include <vector>
#include <Config.h>
#include <Decompressor.h>
#include <TypeDefs.h>
```

Include dependency graph for Validator.h:



Classes

- class vecthor::Validator

Namespaces

- vecthor

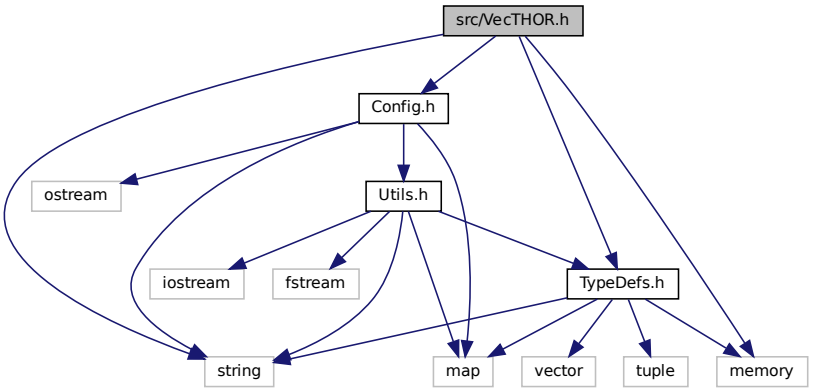
Functions

- template<class T1 , class T2 >
  std::map< T2, T1 > vecthor::swapPairs (std::map< T1, T2 > m)

8.32 src/VecTHOR.h File Reference

```
#include <Config.h>
#include <TypeDefs.h>
#include <memory>
#include <string>
```

Include dependency graph for VecTHOR.h:



Classes

- class vecthor::VecTHOR

Namespaces

- vecthor

Typedefs

- using vecthor::CompressorPtr = std::unique_ptr< Compressor >
- using vecthor::DecompressorPtr = std::unique_ptr< Decompressor >
- using vecthor::CompressedEmitterPtr = std::unique_ptr< CompressedEmitter >
- using vecthor::ValidatorPtr = std::unique_ptr< Validator >

Functions

- int main (int argc, char **argv)

8.32.1 Function Documentation

8.32.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

Index