SignalGP Lite Documentation

Release 0.2.0

Matthew Andres Moreno

Dec 03, 2022

CONTENTS

CHAPTER

ONE

INSTALLATION

The sources for SignalGP Lite can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/mmore500/signalgp-lite
```

Or download the tarball:

```
$ curl  -OL https://github.com/mmore500/signalgp-lite/tarball/master
```

1.1 Docker Container

This project has a containerized build enviroment available with all its dependencies installed. Either build a copy of the container locally from the project's `Dockerfile`, or get a copy of the container from DockerHub.

CHAPTER

TWO

USAGE

To compile this project, run `make` in the root directory.

To use code from this project in your own project, add `-Ipath/to/signalgp-lite/include` to your compiler flags, or add this project as a subrepo and use relative include paths.

CHAPTER

THREE

CITING

Please cite SignalGP-Lite as

bibtex:

```
TODO
```

APA:

TODO

Chicago:

TODO

MLA:

TODO

You can also access metadata to cite SignalGP-Lite in our `CITATION.cff` file. Formatted citations were generated via https://bibtex.online.

CHAPTER

FOUR

PUBLICATIONS

- SignalGP-Lite: Event Driven Genetic Programming Library for Large-Scale Artificial Life Applications. In preparation.

CHAPTER

FIVE

PROJECTS USING SIGNALGP-LITE

We'd love to show off your project! To be listed, make a pull request, open an issue, or send an email to m.more500@gmail.com.

- DISHTINY: A Platform for Studying Open-Ended Evolutionary Transitions in Individuality
- Digital Evolution Homework Assignment for CS361 Evolutionary Computing and Artificial Life @ Carleton

CHAPTER

SIX

LIBRARY API

6.1 Class Hierarchy

6.2 File Hierarchy

6.3 Full API

6.3.1 Namespaces

Namespace sgpl

Contents

Namespaces

Classes

- *Template Class CountingIterator*
- *Template Class Cpu*
- *Class EmptyType*
- *Template Class GarbledOutputIterator*
- *Template Class GlobalAnchorIterator*
- *Template Class MemoizeCtor*
- *Template Class OpCodeRectifier*
- *Template Class OpLibraryCoupler*
- *Template Class OpLookup*
- *Template Class Program*
- *Class RandomBool*
- *Class RandomDraw*
- *Class RepeatingNegativeBinomialCountdown*
- *Template Class Resevoir*
- *Template Class RingResevoir*
- *Class Terminal*
- *Class TransposeWindowDisplacementGenerator_Pareto*
- *Class TransposeWindowSizeGenerator_Pareto*

Functions

- *Template Function sgpl::__attribute__*
- *Template Function sgpl::advance_core*
- *Template Function sgpl::count_cores_with_module_idx*
- *Template Function sgpl::count_instructions*
- *Template Function sgpl::count_modules*
- *Template Function sgpl::count_nop_instructions*
- *Template Function sgpl::count_op_instructions*
- *Template Function sgpl::count_operation_random_touches*
- *Function sgpl::count_thread_local_random_touches*
- *Function sgpl::do_random_walk_approximation*
- *Function sgpl::do_random_walk_exact*
- *Function sgpl::do_random_walk_indexmap_approximation*
- *Function sgpl::do_random_walk_normal_approximation*
- *Template Function sgpl::drag_to*
- *Template Function sgpl::enumerate_module_ids*
- *Template Function sgpl::execute_core(sgpl::Core<Spec>&, const sgpl::Program<Spec>&)*

- *Template Function sgpl::execute_core(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)*
- *Template Function sgpl::execute_core_cycles(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, const size_t)*
- *Template Function sgpl::execute_core_cycles(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&, const size_t)*
- *Template Function sgpl::execute_core_slice(sgpl::Core<Spec>&, const sgpl::Program<Spec>&)*
- *Template Function sgpl::execute_core_slice(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)*
- *Template Function sgpl::execute_cpu(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&)*
- *Template Function sgpl::execute_cpu(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)*
- *Template Function sgpl::execute_cpu_n_cycles(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&)*
- *Template Function sgpl::execute_cpu_n_cycles(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)*
- *Template Function sgpl::execute_cpu_n_slices(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&)*
- *Template Function sgpl::execute_cpu_n_slices(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)*
- *Template Function sgpl::get_cur_module_idx*
- *Template Function sgpl::get_module_length*
- *Template Function sgpl::get_module_pos*
- *Template Function sgpl::get_module_regulator*
- *Template Function sgpl::get_module_tag*
- *Template Function sgpl::inst_indel_copy*
- *Template Function sgpl::load_program*
- *Template Function sgpl::make_module_mask*
- *Template Function sgpl::module_indel_copy*
- *Function sgpl::mutate_bits(const std::span<std::byte>, const size_t)*
- *Function sgpl::mutate_bits(const std::span<std::byte>)*
- *Function sgpl::mutate_bytes(const std::span<std::byte>, const size_t)*
- *Function sgpl::mutate_bytes(const std::span<std::byte>)*
- *Template Function sgpl::mutate_copy*
- *Template Function sgpl::next*
- *Template Function sgpl::nop_out_instruction_category*
- *Template Function sgpl::nop_out_instructions*
- *Template Function sgpl::nop_out_module*
- *Template Function sgpl::nop_out_modules*

- *Template Function sgpl::nop_out_nth_op*
- *Template Function sgpl::operator<<*
- *Template Function sgpl::point_mutate*
- *Template Function sgpl::prev*
- *Template Function sgpl::random_between*
- *Function sgpl::random_sign*
- *Template Function sgpl::sequence_mutate_copy(const sgpl::Program<Spec>&, const Config&)*
- *Template Function sgpl::sequence_mutate_copy(const sgpl::Program<Spec>&, const float, const float, const size_t, const InstRangeCopier&)*
- *Template Function sgpl::slide_n*
- *Template Function sgpl::slide_to*
- *Template Function sgpl::sloppy_copy(const T&, const float, const size_t, const size_t)*
- *Template Function sgpl::sloppy_copy(const T&, const float, const std::pair<size_t, size_t>, const size_t)*
- *Template Function sgpl::summarize_module_expression*
- *Template Function sgpl::summarize_module_regulation*
- *Template Function sgpl::transpose_invert_mutate*
- *Template Function sgpl::transpose_window(const RandomIt, const RandomIt, const WindowDisplacementGenerator, const WindowSizeGenerator)*
- *Template Function sgpl::transpose_window(const RandomIt, const RandomIt, const RandomIt, const WindowDisplacementGenerator&, const WindowSizeGenerator&)*

Typedefs

- *Typedef sgpl::ArithmeticOpLibrary*
- *Typedef sgpl::CompleteOpLibrary*
- *Typedef sgpl::ControlFlowOpLibrary*
- *Typedef sgpl::InstRangeCopier_Default*
- *Typedef sgpl::NopOpLibrary*
- *Typedef sgpl::SansLocalRegulationOpLibrary*
- *Typedef sgpl::SansRegulationOpLibrary*
- *Typedef sgpl::TransposeWindowDisplacementGenerator_Default*
- *Typedef sgpl::TransposeWindowSizeGenerator_Default*

Variables

Namespace sgpl::global

Contents

Classes

Namespace sgpl::impl

Contents

Functions

Namespace sgpl::internal

Contents

Classes

- *Template Struct LibraryInstantiator*
- *Template Struct LibraryInstantiator< std::tuple< T... > >*
- *Class ThreadLocalRandom*

Namespace sgpl::local

Contents

Classes

- *Struct Anchor*
- *Struct JumpIf*
- *Struct JumpIfNot*
- *Struct RegulatorAdj*
- *Struct RegulatorDecay*
- *Struct RegulatorGet*
- *Struct RegulatorSet*

Namespace std

Contents

Classes

- *Template Struct hash< sgpl::Program< Spec > >*

6.3.2 Classes and Structs

Struct Add

- Defined in file_include_sgpl_operations_binary_Add.hpp

Struct Documentation

struct `sgpl::Add`

Adds `reg[arg_1]` to `reg[arg_2]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct BitwiseAnd

- Defined in file_include_sgpl_operations_bitwise_BitwiseAnd.hpp

Struct Documentation

struct `sgpl::BitwiseAnd`

Performs a bitwise AND of `reg[arg_1]` and `reg[arg_2]` then stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct BitwiseNot

- Defined in file_include_sgpl_operations_bitwise_BitwiseNot.hpp

Struct Documentation

struct sgpl::**BitwiseNot**

Performs a bitwise NOT of `reg[arg_1]` and `reg[arg_2]` then stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct BitwiseOr

- Defined in file_include_sgpl_operations_bitwise_BitwiseOr.hpp

Struct Documentation

struct sgpl::**BitwiseOr**

Performs a bitwise OR of `reg[arg_1]` and `reg[arg_2]` then stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct BitwiseShift

- Defined in file_include_sgpl_operations_bitwise_BitwiseShift.hpp

Struct Documentation

struct sgpl::**BitwiseShift**

Shifts the bits of `reg[arg_1]` by `reg[arg_2]` positions.

(If `reg[arg_2]` is negative, this is a right shift. Otherwise it is a left shift.) Stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct BitwiseXor

- Defined in file_include_sgpl_operations_bitwise_BitwiseXor.hpp

Struct Documentation

struct sgpl::**BitwiseXor**

Performs a bitwise XOR of `reg[arg_1]` and `reg[arg_2]` then stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct CountOnes

- Defined in file_include_sgpl_operations_bitwise_CountOnes.hpp

Struct Documentation

struct `sgpl::CountOnes`

Counts the number of bits set in `reg[arg_1]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Cpu::impl_

- Defined in file_include_sgpl_hardware_Cpu.hpp

Nested Relationships

This struct is a nested type of *Template Class Cpu*.

Struct Documentation

struct `sgpl::Cpu::impl_`

Public Members

sgpl::*RingResevoir*<core_t, *Spec*::*num_cores*> **scheduler**

size_t **active_core_idx** = {}

std::array<global_jump_table_t, *Spec*::*num_global_jump_tables*> **global_jump_tables** = {}

size_t **lifetime_cycle_clock** = {}

Struct Decrement

- Defined in file_include_sgpl_operations_unary_Decrement.hpp

Struct Documentation

struct `sgpl::Decrement`

Takes `reg[arg_0]`, decrements it by one, and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Divide

- Defined in file_include_sgpl_operations_binary_Divide.hpp

Struct Documentation

struct `sgpl::Divide`

Divides `reg[arg_1]` by `reg[arg_2]` and stores the result in `reg[arg_0]`.

Division by zero can result in an `Inf` or `NaN` value.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Equal

- Defined in file_include_sgpl_operations_comparison_Equal.hpp

Struct Documentation

struct sgpl::**Equal**

Checks whether `reg[arg_1]` is equal to `reg[arg_2]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct ForkIf

- Defined in file_include_sgpl_operations_actions_ForkIf.hpp

Struct Documentation

struct sgpl::**ForkIf**

If {reg[arg_0]} is nonzero, registers a request to activate a new core with the module best-matching the current instruction's tag.

These fork requests are only handled when the current core terminates. Each core may only register 3 fork requests.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Anchor

- Defined in file_include_sgpl_operations_flow_global_Anchor.hpp

Struct Documentation

struct sgpl::global::**Anchor**

Marks a module-begin position.

Based on tag-lookup, new cores or global jump instructions may set the program counter to this instruction's program position.

This instruction can also mark a module-end position executing this instruction can terminate the executing core. If no local anchor instruction is present between the current global anchor instruction and the preceding global anchor instruction, this operation will not terminate the executing core. (This way, several global anchors may lead into the same module.)

However, if a local anchor instruction is present between the current global anchor instruction and the preceding global anchor instruction, this operation will terminate the executing core. Local jump instructions will only consider local anchors between the preceding global anchor and the subsequent global anchor instruction.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*>&, **const** sgpl::*Program*<*Spec*> &*program*, **typename** *Spec*::peripheral_t &*peripheral*) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct JumpIf

- Defined in file_include_sgpl_operations_flow_global_JumpIf.hpp

Struct Documentation

struct sgpl::global::**JumpIf**

Conditionally jumps core to execute a new module.

Jumps the current core to a global anchor that matches the instruction tag if `reg[arg_0]` is nonzero. If `reg[arg_1]` is nonzero, resets registers.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct JumpIfNot

- Defined in file_include_sgpl_operations_flow_global_JumpIfNot.hpp

Struct Documentation

struct `sgpl::global::`**JumpIfNot**

Conditionally jumps core to execute a new module.

Jumps the current core to a global anchor that matches the instruction tag if `reg[arg_0]` is zero. If `reg[arg_1]` is zero, resets registers.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct RegulatorAdj

- Defined in file_include_sgpl_operations_flow_global_RegulatorAdj.hpp

Struct Documentation

template<size_t **JUMP_TABLE_IDX** = 0>
struct sgpl::global::**RegulatorAdj**

Adjusts the regulator value of global jump table tags matching this instruction's tag by the amount `reg[arg_0]`.

This regulator value affects the outcome of tag lookup for module activation on global jump table `JUMP_TABLE_IDX`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct RegulatorDecay

- Defined in file_include_sgpl_operations_flow_global_RegulatorDecay.hpp

Struct Documentation

template<size_t **JUMP_TABLE_IDX** = 0>
struct sgpl::global::**RegulatorDecay**

Ages the regulator decay countdown of global jump table tags matching this instruction's tag by the amount `reg[arg_0]`.

If `reg[arg_0]` is negative, this can forestall decay.

This regulator value affects the outcome of tag lookup for module activation on global jump table `JUMP_TABLE_IDX`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>

std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct RegulatorGet

- Defined in file_include_sgpl_operations_flow_global_RegulatorGet.hpp

Struct Documentation

template<size_t **JUMP_TABLE_IDX** = 0>
struct `sgpl::global::`**RegulatorGet**

Gets the regulator value of the global jump table tag that best matches this instruction's tag.

Stores the value in `reg[arg_0]`. If no tag matches, a no-op is performed.

The regulator value gotten affects the outcome of tag lookup for module activation on global jump table `JUMP_TABLE_IDX`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct RegulatorSet

- Defined in file_include_sgpl_operations_flow_global_RegulatorSet.hpp

Struct Documentation

template<size_t **JUMP_TABLE_IDX** = 0>
struct `sgpl::global::`**RegulatorSet**

Sets the regulator value of global jump table tags matching this instruction's tag by the amount `reg[arg_0]`.

This regulator value affects the outcome of tag lookup for module activation on global jump table `JUMP_TABLE_IDX`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct GreaterThan

- Defined in file_include_sgpl_operations_comparison_GreaterThan.hpp

Struct Documentation

struct `sgpl::GreaterThan`
Checks whether `reg[arg_1]` is greater than `reg[arg_2]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Increment

- Defined in file_include_sgpl_operations_unary_Increment.hpp

Struct Documentation

struct sgpl::**Increment**

Takes `reg[arg_0]`, increments it by one, and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct InstRangeCopier_Indel

- Defined in file_include_sgpl_spec_InstRangeCopier_Indel.hpp

Struct Documentation

struct sgpl::**InstRangeCopier_Indel**

Public Functions

InstRangeCopier_Indel(**const** float *p_defect_*, **const** float *p_defect_is_insertion_* = 0.5f, **const** float *p_garble_* = 0.f)

template<typename **Config** = sgpl::StarterConfig>
InstRangeCopier_Indel(**const** *Config* &*cfg* = {})

template<typename **InputIt**, typename **OutputIt**>
size_t **operator()**(*InputIt first*, *InputIt last*, *OutputIt out*) **const**

InstRangeCopier_Indel &**SetPDefect**(**const** float *v*)

InstRangeCopier_Indel &**SetPDefectIsInsertion**(**const** float *v*)

InstRangeCopier_Indel &**SetPGarble**(**const** float *v*)

InstRangeCopier_Indel &**KnockoutInsertionMutations**()

InstRangeCopier_Indel &**KnockoutDeletionMutations**()

Public Members

float **p_defect**

float **p_defect_is_insertion**

float **p_garble**

Struct InstRangeCopier_Perfect

- Defined in file_include_sgpl_spec_InstRangeCopier_Perfect.hpp

Struct Documentation

struct sgpl::**InstRangeCopier_Perfect**

Public Functions

template<typename **Config** = sgpl::StarterConfig>
InstRangeCopier_Perfect(**const** *Config* &*cfg* = {})

template<typename **InputIt**, typename **OutputIt**>
size_t **operator()**(*InputIt first*, *InputIt last*, *OutputIt out*) **const**

Template Struct Instruction

- Defined in file_include_sgpl_program_Instruction.hpp

Struct Documentation

template<typename **Spec**>
struct sgpl::**Instruction**

Public Types

using library_t = **typename** *Spec*::library_t

using rectifier_t = sgpl::*OpCodeRectifier*<*library_t*>

using tag_t = **typename** *Spec*::tag_t

Public Functions

void **RectifyArgs**()

void **RectifyOpCode**(**const** *rectifier_t* &*r* = *rectifier_t*{})

void **Rectify**(**const** *rectifier_t* &*r* = *rectifier_t*{})

Instruction() = default

Instruction(uit_emp::Random &*rand*)

void **NopOut**()

void **NopOutIfNotAnchor**()

bool **IsNop**() **const noexcept**

bool **IsOp**() **const noexcept**

bool **operator==**(**const** *Instruction* &*other*) **const**

bool **operator!=**(**const** *Instruction* &*other*) **const**

bool **operator<**(**const** *Instruction* &*other*) **const**

std::string **GetOpName**() **const**

const *tag_t* &**GetTag**() **const noexcept**

auto **GetDescriptors**() **const**

auto **GetCategories**() **const**

template<typename **Archive**, cereal::traits::EnableIf<cereal::traits::is_text_archive<*Archive*>::value> = cereal::traits::sfinae>
void **save**(*Archive* &*archive*) **const**

template<typename **Archive**, cereal::traits::EnableIf<cereal::traits::is_text_archive<*Archive*>::value> = cereal::traits::sfinae>
void **load**(*Archive* &*archive*)

template<typename **Archive**, cereal::traits::DisableIf<cereal::traits::is_text_archive<*Archive*>::value> = cereal::traits::sfinae>
void **serialize**(*Archive* &*archive*)

Public Members

unsigned char **op_code**

std::array<unsigned char, 3> **args**

tag_t **tag**

Template Struct LibraryInstantiator

- Defined in file_include_sgpl_library_OpLibraryCoupler.hpp

Struct Documentation

template<typename>
struct LibraryInstantiator

Template Struct LibraryInstantiator< std::tuple< T... > >

- Defined in file_include_sgpl_library_OpLibraryCoupler.hpp

Struct Documentation

template<typename ...**T**>
struct sgpl::internal::**LibraryInstantiator**<std::tuple<*T*...>>

Public Types

using type = sgpl::*OpLibrary*<*T*...>

Template Struct JumpTable

- Defined in file_include_sgpl_hardware_JumpTable.hpp

Struct Documentation

template<typename **Spec**, typename **Impl**>
struct sgpl::**JumpTable**

Public Types

using tag_t = **typename** *Impl*::query_t

using library_t = **typename** *Spec*::library_t

using program_t = sgpl::*Program*<*Spec*>

using uid_t = size_t

Public Functions

auto **MatchRaw**(**const** *tag_t* &*query*) **noexcept**

auto **MatchRegulated**(**const** *tag_t* &*query*) **noexcept**

void **SetRegulator**(**const** *uid_t uid*, **const** float *set*) **noexcept**

void **AdjRegulator**(**const** *uid_t uid*, **const** float *set*) **noexcept**

void **DecayRegulator**(**const** *uid_t uid*, **const** float *amt*) **noexcept**

float **ViewRegulator**(**const** *uid_t uid*) **const noexcept**

void **DecayRegulators**() **noexcept**

uid_t **GetUid**(**const** *tag_t tag*) **const noexcept**

uid_t **GetUid**(**const** size_t *pos*) **const noexcept**

auto **GetVal**(**const** *uid_t uid*) **const noexcept**

bool **HasVal**(**const** size_t *pos*) **const noexcept**

void **Clear**() **noexcept**

size_t **GetSize**() **const noexcept**

void **InitializeLocalAnchors**(**const** *program_t* &*prog*, **const** size_t *start_pos*) **noexcept**

void **InitializeGlobalAnchors**(**const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *inclusion_mod* = 1) **noexcept**

bool **operator==**(**const** *JumpTable* &*other*) **const**

Public Members

Impl **match_bin**

Struct LessThan

- Defined in file_include_sgpl_operations_comparison_LessThan.hpp

Struct Documentation

struct `sgpl::`**LessThan**

Checks whether `reg[arg_1]` is less than `reg[arg_2]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Anchor

- Defined in file_include_sgpl_operations_flow_local_Anchor.hpp

Struct Documentation

struct `sgpl::local::`**Anchor**

Marks a program location local jump instructions may route to.

Local jump instructions enable conditionals and looping within modules. This program location is tagged with the instruction's tag.

As described in `dish2::global::Anchor`'s docstring, this operation also plays a role in determining whether global anchor instructions close a module.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*>&, **const** sgpl::*Instruction*<*Spec*>&, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct JumpIf

- Defined in file_include_sgpl_operations_flow_local_JumpIf.hpp

Struct Documentation

struct `sgpl::local::JumpIf`

Jumps to a local anchor that matches the instruction tag if `reg[arg_0]` is nonzero.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct JumpIfNot

- Defined in file_include_sgpl_operations_flow_local_JumpIfNot.hpp

Struct Documentation

struct sgpl::local::**JumpIfNot**

Jumps to a local anchor that matches the instruction tag if `reg[arg_0]` is zero.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct RegulatorAdj

- Defined in file_include_sgpl_operations_flow_local_RegulatorAdj.hpp

Struct Documentation

struct sgpl::local::**RegulatorAdj**

Adjusts the regulator value of local jump table tags matching this instruction's tag by the amount `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct RegulatorDecay

- Defined in file_include_sgpl_operations_flow_local_RegulatorDecay.hpp

Struct Documentation

struct sgpl::local::**RegulatorDecay**

Ages the regulator decay countdown of local jump table tags matching this instruction's tag by the amount `reg[arg_0]`.

When a regulator ages past a threshold, it is reset to default. If `reg[arg_0]` is negative, this operation can forestall decay.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct RegulatorGet

- Defined in file_include_sgpl_operations_flow_local_RegulatorGet.hpp

Struct Documentation

struct sgpl::local::**RegulatorGet**

Gets the regulator value of the local jump table tag that best matches this instruction's tag.

Stores the value in `reg[arg\_0]`. If no tag matches, a no-op is performed.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct RegulatorSet

- Defined in file_include_sgpl_operations_flow_local_RegulatorSet.hpp

Struct Documentation

struct `sgpl::local::RegulatorSet`
Sets the regulator value of global jump table tags matching this instruction's tag to `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct LogicalAnd

- Defined in file_include_sgpl_operations_comparison_LogicalAnd.hpp

Struct Documentation

struct `sgpl::LogicalAnd`
Performs a logical AND of `reg[arg_1]` and `reg[arg_2]` then stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct LogicalOr

- Defined in file_include_sgpl_operations_comparison_LogicalOr.hpp

Struct Documentation

struct sgpl::**LogicalOr**

Performs a bitwise OR of `reg[arg_1]` and `reg[arg_2]` then stores the result in `reg[arg_0]}`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Modulo

- Defined in file_include_sgpl_operations_binary_Modulo.hpp

Struct Documentation

struct sgpl::**Modulo**

Calculates the modulus of `reg[arg_1]` by `reg[arg_2]` and stores the result in `reg[arg_0]`.

Mod by zero can result in a `NaN` value.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Multiply

- Defined in file_include_sgpl_operations_binary_Multiply.hpp

Struct Documentation

struct sgpl::**Multiply**

Multiplies `reg[arg_1]` by `reg[arg_2]` and stores the result in `reg[arg_0]}`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Negate

- Defined in file_include_sgpl_operations_unary_Negate.hpp

Struct Documentation

struct sgpl::**Negate**

Negates `reg[arg_0]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

size_t **num_registers_to_print**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct Nop

- Defined in file_include_sgpl_operations_actions_Nop.hpp

Struct Documentation

template<size_t **NumRngTouches** = 0, size_t **Prevalence** = 1>
struct `sgpl::Nop`

Performs no operation for one virtual CPU cycle.

Advances the RNG engine `NumRngTouches` times. (Important to nop-out operations that perform one RNG touch without causing side effects.)

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*>&, **const** sgpl::*Instruction*<*Spec*>&, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct Not

- Defined in file_include_sgpl_operations_unary_Not.hpp

Struct Documentation

struct `sgpl::Not`

Performs a logical not on `reg[arg_0]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct NotEqual

- Defined in file_include_sgpl_operations_comparison_NotEqual.hpp

Struct Documentation

struct `sgpl::NotEqual`
Checks whether `reg[arg_1]` is not equal to `reg[arg_2]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct OpLibrary

- Defined in file_include_sgpl_library_OpLibrary.hpp

Inheritance Relationships

Base Type

- `public std::tuple< Ops... >`

Struct Documentation

template<typename ...**Ops**>
struct `sgpl::OpLibrary` : **public** std::tuple<*Ops*...>

Public Types

using **parent_t** = std::tuple<*Ops*...>

using **this_t** = sgpl::*OpLibrary*<*Ops*...>

using **Operation** = **typename** std::tuple_element<I, *parent_t*>::type

Public Static Functions

constexpr bool **IsAnchorLocalOpCode**(**const** size_t *op_code*) **noexcept**

constexpr bool **IsAnchorGlobalOpCode**(**const** size_t *op_code*) **noexcept**

bool **IsNopOpCode**(**const** size_t *op_code*) **noexcept**

constexpr bool **IsAnchorOpCode**(**const** size_t *op_code*) **noexcept**

constexpr size_t **GetSize**() **noexcept**

std::string **GetOpName**(**const** size_t *op_code*)

template<typename **Spec**>
size_t **GetOpNumRngTouches**(**const** size_t *op_code*)

unsigned char **GetOpCode**(**const** std::string &*op_name*)

unsigned char **GetNopOpCode**(**const** size_t *num_rng_touches* = 0)

size_t **GetOpPrevalence**(**const** size_t *op_code*)

template<typename **Instruction**>
auto **GetOpDescriptors**(**const** size_t *op_code*, **const** *Instruction* &*instruction*)

template<typename **Instruction**>
auto **GetOpCategories**(**const** size_t *op_code*, **const** *Instruction* &*instruction*)

Public Static Attributes

sgpl::*OpLookup*<*this_t*> **lookup_table**

Struct RandomFill

- Defined in file_include_sgpl_operations_bitwise_RandomFill.hpp

Struct Documentation

struct `sgpl::`**RandomFill**

Fills register pointed to by `reg[arg_0]` with random bits chosen from a uniform distribution.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct Spec

- Defined in file_include_sgpl_spec_Spec.hpp

Struct Documentation

template<typename **Library** = sgpl::*CompleteOpLibrary*, typename **Peripheral** = sgpl::*EmptyType*>
struct `sgpl::Spec`

Public Types

using library_t = *Library*

using peripheral_t = *Peripheral*

using global_matching_t = uit_emp::MatchDepository<unsigned short, uit_emp::OptimizedApproxDualStreakMetric<6
What matching implementation should we use for global jump tables?

using local_matching_t = uit_emp::MatchDepository<unsigned short, uit_emp::OptimizedApproxDualStreakMetric<64
What matching datastructure implementation should we use for local jump tables?

using tag_t = **typename** *global_matching_t*::tag_t

Public Static Attributes

constexpr size_t **num_cores** = {16}
How many virtual cores should a virtual CPU be able to support?

constexpr size_t **num_fork_requests** = {3}
How many fork requests can a virtual core make at most?

constexpr size_t **num_registers** = {8}
How many registers should each virtual core contain?

constexpr size_t **switch_steps** = {8}
Maximum num steps executed on one core before next core is executed.

constexpr std::array<size_t, 2> **global_jump_table_inclusion_mods** = {1, 2}

constexpr size_t **num_global_jump_tables** = *global_jump_table_inclusion_mods*.size()

Struct Subtract

- Defined in file_include_sgpl_operations_binary_Subtract.hpp

Struct Documentation

struct sgpl::**Subtract**

Subtracts `reg[arg_2]` from `reg[arg_1]` and stores the result in `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Struct TerminateIf

- Defined in file_include_sgpl_operations_actions_TerminateIf.hpp

Struct Documentation

struct sgpl::**TerminateIf**

Terminates current core if `reg[arg_0]` is nonzero.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Template Struct hash< sgpl::Program< Spec > >

- Defined in file_include_sgpl_program_Program.hpp

Struct Documentation

template<typename **Spec**>
struct std::**hash**<sgpl::*Program*<*Spec*>>

Public Functions

size_t **operator()**(**const** sgpl::*Program*<*Spec*> &*program*) **const**

Template Class CappedOutputIterator

- Defined in file_include_sgpl_utility_CappedOutputIterator.hpp

Inheritance Relationships

Base Type

- protected OutputIterator

Class Documentation

template<typename **OutputIterator**>
class sgpl::**CappedOutputIterator** : **protected** *OutputIterator*

Public Types

using container_type = **typename** parent_t::container_type

using value_type = **typename** parent_t::value_type

using pointer = **typename** parent_t::pointer

using reference = **typename** parent_t::reference

using iterator_category = std::output_iterator_tag

using difference_type = **typename** parent_t::difference_type

Public Functions

CappedOutputIterator(**const** *OutputIterator* &*out*, **const** size_t *cap*)

CappedOutputIterator &**operator***()

CappedOutputIterator &**operator->**()

CappedOutputIterator &**operator++**()

CappedOutputIterator &**operator++**(int)

CappedOutputIterator &**operator=**(**const** container_value_type &*value*)

Template Class CappedSet

- Defined in file_include_sgpl_utility_CappedSet.hpp

Class Documentation

template<typename **T**, size_t **N**>
class sgpl::**CappedSet**

Public Functions

T &**operator[]**(**const** size_t *pos*)

const *T* &**operator[]**(**const** size_t *pos*) **const**

bool **try_push_back**(**const** *T* &*value*)

bool **try_push_back**(*T* &&*value*)

void **push_back**(**const** *T* &*value*)

void **push_back**(*T* &&*value*)

void **push_back**()

void **clear**()

size_t **size**() **const**

constexpr size_t **max_size**() **const**

bool **empty**() **const**

bool **full**() **const**

void **pop_back**()

T &**back**()

const *T* &**back**() **const**

T &**front**()

const *T* &**front**() **const**

void **erase**(**const** size_t *pos*)

T ***begin**()

const *T* ***begin**() **const**

T ***end**()

const *T* ***end**() **const**

bool **operator==**(**const** *CappedSet* &*other*) **const**

Template Class Core

- Defined in file_include_sgpl_hardware_Core.hpp

Class Documentation

template<typename **Spec**>
class sgpl::**Core**

Public Types

using **registers_t** = std::array<float, *Spec*::num_registers>

Public Functions

Core() = default

Core(global_jump_table_array_t &*global_jump_tables_*)

Core(**const** *registers_t* &*registers_*)

void **Terminate**() **noexcept**

__attribute__((hot)) **inline** bool **HasTerminated**() **const** **noexcept**

__attribute__((hot)) **inline** size_t **GetProgramCounter**() **const** **noexcept**

__attribute__((hot)) void **AdvanceProgramCounter**(**const** size_t *program_length*) **noexcept**

bool **HasLocalAnchors**() **const** **noexcept**

void **LoadLocalAnchors**(**const** sgpl::*Program*<*Spec*> &*program*) **noexcept**

void **JumpToGlobalAnchorMatch**(**const** tag_t &*query*, **const** size_t *jt_idx* = 0) **noexcept**

void **JumpToLocalAnchorMatch**(**const** tag_t &*query*) **noexcept**

auto &**GetLocalJumpTable**() **noexcept**

auto &**GetGlobalJumpTable**(**const** size_t *jt_idx* = 0) **noexcept**

bool **RequestFork**(**const** tag_t &*tag*) **noexcept**

void **ResetRegisters**() **noexcept**

void **SetRegisters**(**const** *registers_t* &*set*) **noexcept**

const *registers_t* &**GetRegisters**() **noexcept**

void **Reset**() **noexcept**

void **SetGlobalJumpTables**(global_jump_table_array_t &*j_tables*) **noexcept**

void **DecayRegulators**() **noexcept**

bool **operator==**(**const** *Core* &*other*) **const**

Public Members

registers_t **registers** = {}

sgpl::*CappedSet*<tag_t, *Spec*::num_fork_requests> **fork_requests** = {}

Template Class CountingIterator

- Defined in file_include_sgpl_utility_CountingIterator.hpp

Class Documentation

template<typename **T** = size_t>
class sgpl::**CountingIterator**

Public Types

using value_type = *T*

using pointer = *value_type**

using reference = *value_type*&

using iterator_category = std::forward_iterator_tag

using difference_type = int

Public Functions

CountingIterator() = default

CountingIterator(**const** *T* &*t*)

value_type **operator***() **const**

CountingIterator **operator++**(int)

CountingIterator &**operator++**()

CountingIterator **operator+**(**const** size_t *rhs*)

bool **operator==**(**const** *CountingIterator* &*other*) **const**

bool **operator!=**(**const** *CountingIterator* &*other*) **const**

Template Class Cpu

- Defined in file_include_sgpl_hardware_Cpu.hpp

Nested Relationships

Nested Types

- *Struct Cpu::impl_*

Class Documentation

template<typename **Spec**>
class sgpl::**Cpu**

Public Functions

Cpu() **noexcept**
Default constructor.

Cpu(**const** *Cpu* &*other*) **noexcept**
Copy constructor.

Cpu(*Cpu* &&*other*) **noexcept**
Move constructor.

Cpu &**operator=**(**const** *Cpu* &*other*) **noexcept**
Copy assignment operator.

Cpu &**operator=**(*Cpu* &&*other*) **noexcept**
Move assignment operator.

void **ActivateNextCore**() **noexcept**

bool **TryActivateNextCore**() **noexcept**

void **ActivatePrevCore**() **noexcept**

bool **TryActivatePrevCore**() **noexcept**

__attribute__((hot)) core_t &**GetActiveCore**() **noexcept**

core_t &**GetFreshestCore**() **noexcept**

void **KillActiveCore**() **noexcept**

void **KillStaleCore**() **noexcept**

void **DoLaunchCore**() **noexcept**

bool **TryLaunchCore**() **noexcept**

void **ForceLaunchCore**() **noexcept**

void **DoLaunchCore**(**const** tag_t &*tag*, **const** size_t *jt_idx* = 0) **noexcept**

bool **TryLaunchCore**(**const** tag_t &*tag*, **const** size_t *jt_idx* = 0) **noexcept**

void **ForceLaunchCore**(**const** tag_t &*tag*, **const** size_t *jt_idx* = 0) **noexcept**

size_t **GetNumBusyCores**() **const noexcept**

size_t **GetNumFreeCores**() **const noexcept**

size_t **GetMaxCores**() **const noexcept**

__attribute__((hot)) bool **HasActiveCore**() **const noexcept**

__attribute__((hot)) bool **HasFreeCore**() **const noexcept**

void **Reset**() **noexcept**

void **InitializeAnchors**(**const** sgpl::*Program*<*Spec*> &*program*) **noexcept**

const core_t &**GetCore**(**const** size_t *idx*) **const noexcept**

const global_jump_table_t &**GetGlobalJumpTable**(**const** size_t *idx* = 0) **const noexcept**

void **DecayGlobalRegulators**() **noexcept**

void **AdvanceCycleClock**(**const** size_t *amt*) **noexcept**

size_t **GetCyclesSinceConstruction**() **const noexcept**

Class EmptyType

- Defined in file_include_sgpl_utility_EmptyType.hpp

Class Documentation

class EmptyType

Template Class GarbledOutputIterator

- Defined in file_include_sgpl_utility_GarbledOutputIterator.hpp

Inheritance Relationships

Base Type

- `protected OutputIterator`

Class Documentation

template<typename **OutputIterator**>
class `sgpl::` **GarbledOutputIterator** : **protected** *OutputIterator*

Public Types

using container_type = **typename** parent_t::container_type

using value_type = **typename** parent_t::value_type

using pointer = **typename** parent_t::pointer

using reference = **typename** parent_t::reference

using iterator_category = std::output_iterator_tag

using difference_type = **typename** parent_t::difference_type

Public Functions

GarbledOutputIterator(**const** *OutputIterator* &*out*)

void **AddGarble**(int *amount*)

GarbledOutputIterator &**operator***()

GarbledOutputIterator &**operator->**()

GarbledOutputIterator &**operator++**()

GarbledOutputIterator &**operator++**(int)

GarbledOutputIterator &**operator=**(**const** container_value_type &*value*)

Template Class GlobalAnchorIterator

- Defined in file_include_sgpl_program_GlobalAnchorIterator.hpp

Inheritance Relationships

Base Type

- `protected std::vector::const_iterator< sgpl::Instruction< Spec > >`

Class Documentation

template<typename **Spec**>
class `sgpl::GlobalAnchorIterator` : **protected** std::vector::const_iterator<sgpl::*Instruction*<*Spec*>>

Public Types

using value_type = inst_t

using pointer = *value_type**

using reference = *value_type*&

using iterator_category = std::forward_iterator_tag

using difference_type = **typename** parent_t::difference_type

Public Functions

const *value_type* &**operator*** ()

const *value_type* ***operator->** ()

parent_t **begin** () **const**

parent_t **end** () **const**

GlobalAnchorIterator &**operator++** ()

GlobalAnchorIterator **operator++** (int)

bool **operator==** (**const** *GlobalAnchorIterator* &*other*) **const**

bool **operator!=** (**const** *GlobalAnchorIterator* &*other*) **const**

size_t **CalcDistance** (**const** parent_t &*from*) **const**

Public Static Functions

GlobalAnchorIterator **make_begin** (**const** container_t &*container*)

GlobalAnchorIterator **make_end** (**const** container_t &*container*)

Class ThreadLocalRandom

- Defined in file_include_sgpl_utility_ThreadLocalRandom.hpp

Class Documentation

class sgpl::internal::**ThreadLocalRandom**

Public Functions

uit_emp::Random &**Get** ()

std::byte **GetByte** ()

void **Reseed** (**const** int *seed*)

void **Initialize** (**const** int *seed*)

void **SeedStochastically** ()

Template Class MemoizeCtor

- Defined in file_include_sgpl_utility_MemoizeCtor.hpp

Inheritance Relationships

Base Type

- `public T`

Class Documentation

template<typename **T**>
class `sgpl::MemoizeCtor` : **public** *T*

Public Functions

template<typename ...**Args**>
MemoizeCtor(*Args*&&... *args*)

MemoizeCtor(**const** *MemoizeCtor*&) = default

MemoizeCtor(*MemoizeCtor*&&) = default

Public Static Functions

template<typename ...**Args**>
const *T* &**lookup**(*Args*&&... *args*)

Template Class OpCodeRectifier

- Defined in file_include_sgpl_program_OpCodeRectifier.hpp

Class Documentation

template<typename **Library**>
class `sgpl::OpCodeRectifier`

Public Functions

OpCodeRectifier()

unsigned char **Rectify**(**const** unsigned char *oc*) **const**

Template Class OpLibraryCoupler

- Defined in file_include_sgpl_library_OpLibraryCoupler.hpp

Inheritance Relationships

Base Type

- `public internal::LibraryInstantiator::type< decltype(std::tuple_cat(std::declval< Library::parent_t >(), std::declval< std::tuple< Ops... >>())) >`

Class Documentation

template<typename **Library**, typename ...**Ops**>
class OpLibraryCoupler : **public** internal::*LibraryInstantiator*::type<decltype(std::tuple_cat(std::declval<*Library*::parent_t>

Template Class OpLookup

- Defined in file_include_sgpl_library_OpLookup.hpp

Class Documentation

template<typename **Library**>
class `sgpl::OpLookup`

Public Functions

OpLookup()

unsigned char **GetOpCode**(**const** std::string *op_name*) **const**

Public Static Functions

std::string **GetOpName**(**const** size_t *op_code*)

template<typename **Spec**>
size_t **GetOpNumRngTouches**(**const** size_t *op_code*)

size_t **GetNopOpCode**(**const** size_t *num_rng_touches*)

size_t **GetOpPrevalence**(**const** size_t *op_code*)

template<typename **Instruction**>
auto **GetOpDescriptors**(**const** size_t *op_code*, **const** *Instruction* &*instruction*)

template<typename **Instruction**>
auto **GetOpCategories**(**const** size_t *op_code*, **const** *Instruction* &*instruction*)

Template Class Program

- Defined in file_include_sgpl_program_Program.hpp

Inheritance Relationships

Base Type

- `public std::vector< sgpl::Instruction< Spec > >`

Class Documentation

template<typename **Spec**>
class `sgpl::`**Program** : **public** std::vector<sgpl::*Instruction*<*Spec*>>

Public Functions

Program() = default
Default constructor.

Program(**const** size_t *n*)

Program(**const** char **as_json*)
Deserialize from JSON string.

Program(**const** std::filesystem::path &*path*)
Deserialize from file.

Program(**const** *Program* &*other*)
Copy constructor.

Program(*Program* &&*other*)
Move constructor.

Program(**const** parent_t &*other*)
Raw copy constructor.

Program(parent_t &&*other*)
Raw move constructor.

Program &**operator=**(**const** *Program* &*other*)
Copy assignment operator.

Program &**operator=**(*Program* &&*other*)
Move assignment operator.

Program &**operator=**(parent_t &&*other*)
Raw move assignment operator.

size_t **ApplyPointMutations**(**const** float *p_bit_toggle*, **const** rectifier_t &*rectifier* = rectifier_t{})

void **RotateGlobalAnchorToFront**()

void **Rectify**(**const** rectifier_t &*rectifier* = rectifier_t{})

bool **HasGlobalAnchor**() **const**

Class RandomBool

- Defined in file_include_sgpl_operations_unary_RandomBool.hpp

Class Documentation

class `sgpl::RandomBool`

With probability determined by this instruction's tag, stores 1.0f to `reg[arg_0]`.

Otherwise, stores 0.0f to `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Class RandomDraw

- Defined in file_include_sgpl_operations_unary_RandomDraw.hpp

Class Documentation

class `sgpl::RandomDraw`

Stores a randomly drawn float value to `reg[arg_0]`.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&) **noexcept**

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Class RepeatingNegativeBinomialCountdown

- Defined in file_include_sgpl_utility_RepeatingNegativeBinomialCountdown.hpp

Class Documentation

class sgpl::**RepeatingNegativeBinomialCountdown**

Public Functions

RepeatingNegativeBinomialCountdown(double *p_*, size_t *n_* = 1)

bool **TestAndStep**()

Template Class Resevoir

- Defined in file_include_sgpl_utility_Resevoir.hpp

Class Documentation

template<typename **T**, size_t **N**>
class sgpl::**Resevoir**

Public Functions

T &**operator[]**(**const** size_t *pos*)

const *T* &**operator[]**(**const** size_t *pos*) **const**

bool **try_acquire**()

void **acquire**()

void **clear**()

size_t **size**() **const**

constexpr size_t **max_size**() **const**

bool **empty**() **const**

bool **full**() **const**

void **release_back**()

T &**back**()

const *T* &**back**() **const**

T &**front**()

const *T* &**front**() **const**

void **release**(**const** size_t *pos*)

T ***begin**()

const *T* ***begin**() **const**

T ***end**()

const *T* ***end**() **const**

auto &**buffer**()

const auto &**buffer**() **const**

Template Class RingResevoir

- Defined in file_include_sgpl_utility_RingResevoir.hpp

Class Documentation

template<typename **T**, size_t **N**>
class sgpl::**RingResevoir**

Public Functions

size_t **GetSize**() **const**

constexpr size_t **GetCapacity**() **const**

size_t **GetAvailableCapacity**() **const**

bool **IsEmpty**() **const**

bool **IsFull**() **const**

T &**Get**(**const** size_t *pos*)

const *T* &**Get**(**const** size_t *pos*) **const**

T &**GetTail**()

T &**GetHead**()

T &**Acquire**()

void **ReleaseHead**()

void **ReleaseTail**()

bool **IsTail**(**const** size_t *pos*) **const**

bool **IsHead**(**const** size_t *pos*) **const**

void **Release**(**const** size_t *pos*)

void **Fill**(**const** *T* &*t*)

void **Reset**()

std::array<*T*, *N*> &**GetBuffer**()

Class Terminal

- Defined in file_include_sgpl_operations_unary_Terminal.hpp

Class Documentation

class sgpl::**Terminal**

Stores a genetically-encoded value to `reg[arg_0]`.

This value is determined deterministically using the instruction's tag.

Public Static Functions

template<typename **Spec**>
void **run**(sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Instruction*<*Spec*> &*inst*, **const** sgpl::*Program*<*Spec*>&, **typename** *Spec*::peripheral_t&)

std::string **name**()

size_t **prevalence**()

template<typename **Spec**>
auto **descriptors**(**const** sgpl::*Instruction*<*Spec*> &*inst*)

template<typename **Spec**>
std::set<std::string> **categories**(**const** sgpl::*Instruction*<*Spec*>&)

Class TransposeWindowDisplacementGenerator_Pareto

- Defined in file_include_sgpl_spec_TransposeWindowDisplacementGenerator_Pareto.hpp

Class Documentation

class sgpl::**TransposeWindowDisplacementGenerator_Pareto**

Public Functions

TransposeWindowDisplacementGenerator_Pareto(**const** float *pareto_shape_*, **const** float *pareto_scale_*)

template<typename **Config** = sgpl::StarterConfig>
TransposeWindowDisplacementGenerator_Pareto(**const** *Config* &*cfg*)

TransposeWindowDisplacementGenerator_Pareto()

int **operator()**(**const** size_t *num_sites_before_window*, **const** size_t *num_sites_after_window*) **const**

Class TransposeWindowSizeGenerator_Pareto

- Defined in file_include_sgpl_spec_TransposeWindowSizeGenerator_Pareto.hpp

Class Documentation

class sgpl::**TransposeWindowSizeGenerator_Pareto**

Public Functions

TransposeWindowSizeGenerator_Pareto(**const** float *pareto_shape_*, **const** float *pareto_scale_*)

template<typename **Config** = sgpl::StarterConfig>
TransposeWindowSizeGenerator_Pareto(**const** *Config* &*cfg* = {})

TransposeWindowSizeGenerator_Pareto()

int **operator()**(**const** size_t *genome_length*) **const**

6.3.3 Functions

Template Function sgpl::__attribute__

- Defined in file_include_sgpl_algorithm_advance_core.hpp

Function Documentation

template<typename Spec> sgpl::__attribute__ ((hot)) inline void advance_core(sgpl

Template Function sgpl::advance_core

- Defined in file_include_sgpl_algorithm_advance_core.hpp

Function Documentation

template<typename **Spec**>
void sgpl::**advance_core**(sgpl::*Core*<*Spec*> &*state*, **const** sgpl::*Program*<*Spec*> &*program*)

Template Function sgpl::count_cores_with_module_idx

- Defined in file_include_sgpl_introspection_count_cores_with_module_idx.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::count_cores_with_module_idx`(**const** sgpl::*Cpu*<*Spec*> &*cpu*, **const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *module_idx*)

Template Function sgpl::count_instructions

- Defined in file_include_sgpl_introspection_count_instructions.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::count_instructions`(**const** sgpl::*Program*<*Spec*> &*program*, **const** std::string &*category*)

Template Function sgpl::count_modules

- Defined in file_include_sgpl_introspection_count_modules.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::count_modules`(**const** sgpl::*Program*<*Spec*> &*program*)

Template Function sgpl::count_nop_instructions

- Defined in file_include_sgpl_introspection_count_nop_instructions.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::count_nop_instructions`(**const** sgpl::*Program*<*Spec*> &*program*)

Template Function sgpl::count_op_instructions

- Defined in file_include_sgpl_introspection_count_op_instructions.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::count_op_instructions`(**const** sgpl::*Program*<*Spec*> &*program*)

Template Function sgpl::count_operation_random_touches

- Defined in file_include_sgpl_utility_count_operation_random_touches.hpp

Function Documentation

template<typename **Operation**, typename **Spec**>
size_t `sgpl::count_operation_random_touches`()

Function sgpl::count_thread_local_random_touches

- Defined in file_include_sgpl_utility_count_thread_local_random_touches.hpp

Function Documentation

size_t `sgpl::count_thread_local_random_touches`(**const** std::function<void)
> *routine*

Function sgpl::do_random_walk_approximation

- Defined in file_include_sgpl_utility_do_random_walk_approximation.hpp

Function Documentation

int `sgpl::do_random_walk_approximation`(**const** int *num_steps*)

Function sgpl::do_random_walk_exact

- Defined in file_include_sgpl_utility_do_random_walk_exact.hpp

Function Documentation

int `sgpl::do_random_walk_exact`(**const** int *num_steps*)

Function sgpl::do_random_walk_indexmap_approximation

- Defined in file_include_sgpl_utility_do_random_walk_indexmap_approximation.hpp

Function Documentation

int `sgpl::do_random_walk_indexmap_approximation`(**const** int *num_steps*)

Function sgpl::do_random_walk_normal_approximation

- Defined in file_include_sgpl_utility_do_random_walk_normal_approximation.hpp

Function Documentation

int `sgpl::do_random_walk_normal_approximation`(**const** int *num_steps*)

Template Function sgpl::drag_to

- Defined in file_include_sgpl_algorithm_drag_to.hpp

Function Documentation

template<typename **RandomIt**>
RandomIt `sgpl::drag_to`(**const** *RandomIt first*, **const** *RandomIt last*, **const** *RandomIt to*)

Move the front of a selected range to a target position.

The front of the selected range is repositioned so as if inserted before the target position. If the target position is within the selected range, performs a rotate so that the front of the selection reaches the target position. Note that for target position equal or one after the front of the selected range, this will result in a no-op. In the case of an empty selected range, the target position is returned.

Return iterator at the new position of the front of the selected range

Parameters

- `first`: front of selection
- `last`: back of selection
- `to`: target to slide to

Template Parameters

- `random`: access iterator type

Template Function sgpl::enumerate_module_ids

- Defined in file_include_sgpl_introspection_enumerate_module_ids.hpp

Function Documentation

template<typename **Spec**>
std::vector<size_t> `sgpl::`**`enumerate_module_ids`**`(`**`const`** sgpl::*Program*<*Spec*> &*program*`)`

Template Function sgpl::execute_core(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)

- Defined in file_include_sgpl_algorithm_execute_core.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_core" with arguments (sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> size_t execute_core(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program)
- template<typename Spec> size_t execute_core(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, typename Spec::peripheral_t &peripheral)
```

Template Function sgpl::execute_core(sgpl::Core<Spec>&, const sgpl::Program<Spec>&)

- Defined in file_include_sgpl_algorithm_execute_core.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_core" with arguments (sgpl::Core<Spec>&, const sgpl::Program<Spec>&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> size_t execute_core(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program)
- template<typename Spec> size_t execute_core(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, typename Spec::peripheral_t &peripheral)
```

Template Function sgpl::execute_core_cycles(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&, const size_t)

- Defined in file_include_sgpl_algorithm_execute_core_cycles.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_core_cycles" with arguments (sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&, const size_t) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> size_t execute_core_cycles(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, const size_t max_cycles)
- template<typename Spec> size_t execute_core_cycles(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, typename Spec::peripheral_t &peripheral, const size_
↪t max_cycles)
```

Template Function sgpl::execute_core_cycles(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, const size_t)

- Defined in file_include_sgpl_algorithm_execute_core_cycles.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_core_cycles" with arguments (sgpl::Core<Spec>&, const sgpl::Program<Spec>&, const size_t) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> size_t execute_core_cycles(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, const size_t max_cycles)
- template<typename Spec> size_t execute_core_cycles(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, typename Spec::peripheral_t &peripheral, const size_
↪t max_cycles)
```

Template Function sgpl::execute_core_slice(sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)

- Defined in file_include_sgpl_algorithm_execute_core_slice.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_core_slice" with arguments (sgpl::Core<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> size_t execute_core_slice(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program)
- template<typename Spec> size_t execute_core_slice(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, typename Spec::peripheral_t &peripheral)
```

Template Function sgpl::execute_core_slice(sgpl::Core<Spec>&, const sgpl::Program<Spec>&)

- Defined in file_include_sgpl_algorithm_execute_core_slice.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_core_slice" with arguments (sgpl::Core<Spec>&, const sgpl::Program<Spec>&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> size_t execute_core_slice(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program)
- template<typename Spec> size_t execute_core_slice(sgpl::Core<Spec> &state, const␣
↪sgpl::Program<Spec> &program, typename Spec::peripheral_t &peripheral)
```

Template Function sgpl::execute_cpu(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)

- Defined in file_include_sgpl_algorithm_execute_cpu.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_cpu" with arguments (const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> void execute_cpu(const size_t max_slices, sgpl::Cpu<Spec>␣
↪&state, const sgpl::Program<Spec> &program)
- template<typename Spec> void execute_cpu(const size_t max_slices, sgpl::Cpu<Spec>␣
↪&state, const sgpl::Program<Spec> &program, typename Spec::peripheral_t &
↪peripheral)
```

Template Function sgpl::execute_cpu(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&)

- Defined in file_include_sgpl_algorithm_execute_cpu.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_cpu" with arguments (const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec> void execute_cpu(const size_t max_slices, sgpl::Cpu<Spec>␣
↪&state, const sgpl::Program<Spec> &program)
- template<typename Spec> void execute_cpu(const size_t max_slices, sgpl::Cpu<Spec>␣
↪&state, const sgpl::Program<Spec> &program, typename Spec::peripheral_t &
↪peripheral)
```

Template Function sgpl::execute_cpu_n_cycles(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)

- Defined in file_include_sgpl_algorithm_execute_cpu_n_cycles.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_cpu_n_cycles" with arguments (const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec = sgpl::Spec<>> void execute_cpu_n_cycles(const size_t␣
↪cycles, sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program)
- template<typename Spec> void execute_cpu_n_cycles(const size_t max_cycles,␣
↪sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program, typename␣
↪Spec::peripheral_t &peripheral)
```

Template Function sgpl::execute_cpu_n_cycles(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&)

- Defined in file_include_sgpl_algorithm_execute_cpu_n_cycles.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_cpu_n_cycles" with arguments (const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec = sgpl::Spec<>> void execute_cpu_n_cycles(const size_t␣
↪cycles, sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program)
- template<typename Spec> void execute_cpu_n_cycles(const size_t max_cycles,␣
↪sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program, typename␣
↪Spec::peripheral_t &peripheral)
```

Template Function sgpl::execute_cpu_n_slices(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&)

- Defined in file_include_sgpl_algorithm_execute_cpu_n_slices.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_cpu_n_slices" with arguments (const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&, typename Spec::peripheral_t&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec = sgpl::Spec<>> void execute_cpu_n_slices(const size_t max_
↪slices, sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program)
- template<typename Spec> void execute_cpu_n_slices(const size_t max_slices,␣
↪sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program, typename␣
↪Spec::peripheral_t &peripheral)
```

Template Function sgpl::execute_cpu_n_slices(const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&)

- Defined in file_include_sgpl_algorithm_execute_cpu_n_slices.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::execute_cpu_n_slices" with arguments (const size_t, sgpl::Cpu<Spec>&, const sgpl::Program<Spec>&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec = sgpl::Spec<>> void execute_cpu_n_slices(const size_t max_
↪slices, sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program)
- template<typename Spec> void execute_cpu_n_slices(const size_t max_slices,␣
↪sgpl::Cpu<Spec> &state, const sgpl::Program<Spec> &program, typename␣
↪Spec::peripheral_t &peripheral)
```

Template Function sgpl::get_cur_module_idx

- Defined in file_include_sgpl_introspection_get_cur_module_idx.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::get_cur_module_idx`(**const** sgpl::*Core*<*Spec*> &*core*, **const** sgpl::*Program*<*Spec*> &*program*)

Template Function sgpl::get_module_length

- Defined in file_include_sgpl_introspection_get_module_length.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::get_module_length`(**const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *module_idx*)

Template Function sgpl::get_module_pos

- Defined in file_include_sgpl_introspection_get_module_pos.hpp

Function Documentation

template<typename **Spec**>
size_t `sgpl::get_module_pos`(**const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *module_idx*)

Template Function sgpl::get_module_regulator

- Defined in file_include_sgpl_introspection_get_module_regulator.hpp

Function Documentation

template<typename **Spec**>
std::optional<float> `sgpl::get_module_regulator`(**const** sgpl::*Cpu*<*Spec*> &*cpu*, **const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *module_idx*, **const** size_t *jump_table_idx* = 0)

Template Function sgpl::get_module_tag

- Defined in file_include_sgpl_introspection_get_module_tag.hpp

Function Documentation

template<typename **Spec**>
Spec::tag_t `sgpl::get_module_tag`(**const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *module_idx*)

Template Function sgpl::impl::transpose_window

- Defined in file_include_sgpl_algorithm_transpose_window.hpp

Function Documentation

template<typename **RandomIt**, typename **WindowDisplacementGenerator**>
std::tuple<*RandomIt*, *RandomIt*, *RandomIt*> `sgpl::impl::transpose_window`(**const** *RandomIt genome_first*, **const** *RandomIt genome_last*, **const** *RandomIt window_first*, **const** *RandomIt window_last*, **const** *WindowDisplacementGenerator* &*window_displacement_generator*)

Template Function sgpl::inst_indel_copy

- Defined in file_include_sgpl_algorithm_inst_indel_copy.hpp

Function Documentation

template<typename **InputIt**, typename **OutputIt**>
size_t `sgpl::inst_indel_copy`(*InputIt first*, *InputIt last*, *OutputIt out_iter*, **const** float *p_defect*, **const** float *p_garble* = 0.f, **const** float *p_defect_is_insertion* = 0.5f)

Template Function sgpl::load_program

- Defined in file_include_sgpl_program_load_program.hpp

Function Documentation

template<typename **Spec**>
sgpl::*Program*<*Spec*> `sgpl::load_program`(**const** std::filesystem::path &*path*)

Template Function sgpl::make_module_mask

- Defined in file_include_sgpl_introspection_make_module_mask.hpp

Function Documentation

template<typename **Spec**>
std::vector<char> `sgpl::make_module_mask`(**const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *module_idx*)

Template Function sgpl::module_indel_copy

- Defined in file_include_sgpl_algorithm_module_indel_copy.hpp

Function Documentation

template<typename **ModuleIt**, typename **OutputIt**, typename **InstRangeCopier** = sgpl::*InstRangeCopier_Default*>
size_t `sgpl::module_indel_copy`(*ModuleIt program_begin*, *ModuleIt program_end*, *OutputIt out_iter*, **const** float *p_defect*, **const** float *p_defect_is_insertion* = 0.5f, **const** *InstRangeCopier* &*range_copier* = *InstRangeCopier*())

Function sgpl::mutate_bits(const std::span<std::byte>)

- Defined in file_include_sgpl_algorithm_mutate_bits.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::mutate_bits" with arguments (const std::span<std::byte>) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxy-output/xml. Potential matches:

```
- void mutate_bits(const std::span<std::byte> target)
- void mutate_bits(const std::span<std::byte> target, const size_t num_muts)
```

Function sgpl::mutate_bits(const std::span<std::byte>, const size_t)

- Defined in file_include_sgpl_algorithm_mutate_bits.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::mutate_bits" with arguments (const std::span<std::byte>, const size_t) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- void mutate_bits(const std::span<std::byte> target)
- void mutate_bits(const std::span<std::byte> target, const size_t num_muts)
```

Function sgpl::mutate_bytes(const std::span<std::byte>)

- Defined in file_include_sgpl_algorithm_mutate_bytes.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::mutate_bytes" with arguments (const std::span<std::byte>) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- void mutate_bytes(const std::span<std::byte> target)
- void mutate_bytes(const std::span<std::byte> target, const size_t num_muts)
```

Function sgpl::mutate_bytes(const std::span<std::byte>, const size_t)

- Defined in file_include_sgpl_algorithm_mutate_bytes.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::mutate_bytes" with arguments (const std::span<std::byte>, const size_t) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- void mutate_bytes(const std::span<std::byte> target)
- void mutate_bytes(const std::span<std::byte> target, const size_t num_muts)
```

Template Function sgpl::mutate_copy

- Defined in file_include_sgpl_mutate_mutate_copy.hpp

Function Documentation

template<typename **Spec**, typename **Config**>
auto sgpl::**mutate_copy**(**const** sgpl::*Program*<*Spec*> &*original*, **const** *Config* &*cfg*)

Template Function sgpl::next

- Defined in file_include_sgpl_algorithm_next.hpp

Function Documentation

template<typename **InputIt**>
InputIt sgpl::**next**(*InputIt it*, *InputIt bound*, **typename** std::iterator_traits<*InputIt*>::difference_type *n* = 1)

Template Function sgpl::nop_out_instruction_category

- Defined in file_include_sgpl_morph_nop_out_instruction_category.hpp

Function Documentation

template<typename **Spec**>
sgpl::*Program*<*Spec*> sgpl::**nop_out_instruction_category**(sgpl::*Program*<*Spec*> *program*, **const** std::string &*category*)

Template Function sgpl::nop_out_instructions

- Defined in file_include_sgpl_morph_nop_out_instructions.hpp

Function Documentation

template<typename **Spec**>
sgpl::*Program*<*Spec*> sgpl::**nop_out_instructions**(sgpl::*Program*<*Spec*> *program*, **const** std::vector<char> &*should_nop*)

Template Function sgpl::nop_out_module

- Defined in file_include_sgpl_morph_nop_out_module.hpp

Function Documentation

template<typename **Spec**>
sgpl::*Program*<*Spec*> `sgpl::`**`nop_out_module`**(sgpl::*Program*<*Spec*> *program*, **const** size_t *module_idx*)

Template Function sgpl::nop_out_modules

- Defined in file_include_sgpl_morph_nop_out_modules.hpp

Function Documentation

template<typename **Spec**>
sgpl::*Program*<*Spec*> `sgpl::`**`nop_out_modules`**(sgpl::*Program*<*Spec*> *program*, **const** std::vector<char> *modulewise_should_nop*)

Template Function sgpl::nop_out_nth_op

- Defined in file_include_sgpl_morph_nop_out_nth_op.hpp

Function Documentation

template<typename **Spec**>
sgpl::*Program*<*Spec*> `sgpl::`**`nop_out_nth_op`**(sgpl::*Program*<*Spec*> *program*, **const** size_t *n*)

Template Function sgpl::operator<<

- Defined in file_include_sgpl_program_Instruction.hpp

Function Documentation

template<typename **Spec**>
std::ostream &`sgpl::`**`operator<<`**(std::ostream &*os*, **const** *Instruction*<*Spec*> &*inst*)

Template Function sgpl::point_mutate

- Defined in file_include_sgpl_mutate_point_mutate.hpp

Function Documentation

template<typename **Config**, typename **Spec**>
size_t sgpl::**point_mutate**(sgpl::*Program*<*Spec*> &*program*, **const** *Config* &*cfg*)

Template Function sgpl::prev

- Defined in file_include_sgpl_algorithm_prev.hpp

Function Documentation

template<typename **InputIt**>
InputIt sgpl::**prev**(*InputIt* *it*, *InputIt* *bound*, **typename** std::iterator_traits<*InputIt*>::difference_type *n* = 1)

Template Function sgpl::random_between

- Defined in file_include_sgpl_utility_random_between.hpp

Function Documentation

template<typename **InputIt**>
InputIt sgpl::**random_between**(**const** *InputIt* *first*, **const** *InputIt* *last*)

Function sgpl::random_sign

- Defined in file_include_sgpl_utility_random_sign.hpp

Function Documentation

int sgpl::**random_sign**(**const** double *p_positive* = 0.5)

Template Function sgpl::sequence_mutate_copy(const sgpl::Program<Spec>&, const float, const float, const size_t, const InstRangeCopier&)

- Defined in file_include_sgpl_mutate_sequence_mutate_copy.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::sequence_mutate_copy" with arguments (const sgpl::Program<Spec>&, const float, const float, const size_t, const InstRangeCopier&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec, typename Config> auto sequence_mutate_copy(const␣
↪sgpl::Program<Spec> &original, const Config &cfg)
- template<typename Spec, typename InstRangeCopier = sgpl::InstRangeCopier_Indel>␣
↪auto sequence_mutate_copy(const sgpl::Program<Spec> &original, const float p_
↪module_defect, const float p_module_defect_is_insertion, const size_t program_
↪size_cap, const InstRangeCopier &range_copier = {})
```

Template Function sgpl::sequence_mutate_copy(const sgpl::Program<Spec>&, const Config&)

- Defined in file_include_sgpl_mutate_sequence_mutate_copy.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::sequence_mutate_copy" with arguments (const sgpl::Program<Spec>&, const Config&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Spec, typename Config> auto sequence_mutate_copy(const␣
↪sgpl::Program<Spec> &original, const Config &cfg)
- template<typename Spec, typename InstRangeCopier = sgpl::InstRangeCopier_Indel>␣
↪auto sequence_mutate_copy(const sgpl::Program<Spec> &original, const float p_
↪module_defect, const float p_module_defect_is_insertion, const size_t program_
↪size_cap, const InstRangeCopier &range_copier = {})
```

Template Function sgpl::slide_n

- Defined in file_include_sgpl_algorithm_slide_n.hpp

Function Documentation

template<typename **RandomIt**>
RandomIt `sgpl::`**`slide_n`**(*RandomIt* *first*, *RandomIt* *last*, **const** int *amount*)

Slide a selected window over n positions.

Empty space is filled by shifting existing elements outside the window, like dragging and dropping a contiguous selection up or down a list.

Return new position of selection window front

Parameters

- `first`: front of sliding window
- `last`: back of sliding window

- `amount`: how far to slide, left if negative and right if positive

Template Parameters

- `RandomIt`: random access iterator

Template Function sgpl::slide_to

- Defined in file_include_sgpl_algorithm_slide_to.hpp

Function Documentation

template<typename **RandomIt**>
RandomIt `sgpl::slide_to`(**const** *RandomIt first*, **const** *RandomIt last*, **const** *RandomIt to*)

Move the a selected range to a target position.

The front of the selected range is repositioned so as if inserted before the target position. If the target position is within the selected range this will result in a no-op. In the case of an empty selected range, the target position is returned.

Return iterator at the new position of the front of the selected range

Parameters

- `first`: front of selection
- `last`: back of selection
- `to`: target to slide to

Template Parameters

- `random`: access iterator type

Template Function sgpl::sloppy_copy(const T&, const float, const std::pair<size_t, size_t>, const size_t)

- Defined in file_include_sgpl_algorithm_sloppy_copy.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::sloppy_copy" with arguments (const T&, const float, const std::pair<size_t, size_t>, const size_t) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T, bool Scramble = false> auto sloppy_copy(const T &original,␣
↪const float p_defect, const size_t defect_bound, const size_t res_size_limit =␣
↪std::numeric_limits<size_t>::max())
- template<typename T, bool Scramble = false> auto sloppy_copy(const T &original,␣
↪const float p_defect, const std::pair<size_t, size_t> defect_bounds, const size_t␣
↪res_size_limit = std::numeric_limits<size_t>::max())
```

Template Function sgpl::sloppy_copy(const T&, const float, const size_t, const size_t)

- Defined in file_include_sgpl_algorithm_sloppy_copy.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::sloppy_copy" with arguments (const T&, const float, const size_t, const size_t) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T, bool Scramble = false> auto sloppy_copy(const T &original,␣
↪const float p_defect, const size_t defect_bound, const size_t res_size_limit =␣
↪std::numeric_limits<size_t>::max())
- template<typename T, bool Scramble = false> auto sloppy_copy(const T &original,␣
↪const float p_defect, const std::pair<size_t, size_t> defect_bounds, const size_t␣
↪res_size_limit = std::numeric_limits<size_t>::max())
```

Template Function sgpl::summarize_module_expression

- Defined in file_include_sgpl_introspection_summarize_module_expression.hpp

Function Documentation

template<typename **Spec**>
std::vector<size_t> `sgpl::`**`summarize_module_expression`**(**const** sgpl::*Cpu*<*Spec*> &*cpu*, **const** sgpl::*Program*<*Spec*> &*program*)

Template Function sgpl::summarize_module_regulation

- Defined in file_include_sgpl_introspection_summarize_module_regulation.hpp

Function Documentation

template<typename **Spec**>
std::vector<float> `sgpl::`**`summarize_module_regulation`**(**const** sgpl::*Cpu*<*Spec*> &*cpu*, **const** sgpl::*Program*<*Spec*> &*program*, **const** size_t *jump_table_idx* = 0)

Template Function sgpl::transpose_invert_mutate

- Defined in file_include_sgpl_mutate_transpose_invert_mutate.hpp

Function Documentation

template<typename **Config**, typename **Spec**>
size_t sgpl::**transpose_invert_mutate**(sgpl::*Program*<*Spec*> &*program*, **const** *Config* &*cfg*)

Template Function sgpl::transpose_window(const RandomIt, const RandomIt, const RandomIt, const WindowDisplacementGenerator&, const WindowSizeGenerator&)

- Defined in file_include_sgpl_algorithm_transpose_window.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::transpose_window" with arguments (const RandomIt, const RandomIt, const RandomIt, const WindowDisplacementGenerator&, const WindowSizeGenerator&) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename RandomIt, typename WindowDisplacementGenerator =␣
↪sgpl::TransposeWindowDisplacementGenerator_Default, typename WindowSizeGenerator␣
↪= sgpl::TransposeWindowSizeGenerator_Default> std::tuple<RandomIt, RandomIt,␣
↪RandomIt> transpose_window(const RandomIt genome_first, const RandomIt genome_
↪last, const WindowDisplacementGenerator window_displacement_generator = {}, const␣
↪WindowSizeGenerator window_size_generator = {})
- template<typename RandomIt, typename WindowDisplacementGenerator, typename␣
↪WindowSizeGenerator> std::tuple<RandomIt, RandomIt, RandomIt> transpose_
↪window(const RandomIt genome_first, const RandomIt genome_last, const RandomIt␣
↪target_site, const WindowDisplacementGenerator &window_displacement_generator,␣
↪const WindowSizeGenerator &window_size_generator)
```

Template Function sgpl::transpose_window(const RandomIt, const RandomIt, const WindowDisplacementGenerator, const WindowSizeGenerator)

- Defined in file_include_sgpl_algorithm_transpose_window.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "sgpl::transpose_window" with arguments (const RandomIt, const RandomIt, const WindowDisplacementGenerator, const WindowSizeGenerator) in doxygen xml output for project "Matthew Andres Moreno" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename RandomIt, typename WindowDisplacementGenerator =␣
↪sgpl::TransposeWindowDisplacementGenerator_Default, typename WindowSizeGenerator␣
↪= sgpl::TransposeWindowSizeGenerator_Default> std::tuple<RandomIt, RandomIt,␣
↪RandomIt> transpose_window(const RandomIt genome_first, const RandomIt genome_
↪last, const WindowDisplacementGenerator window_displacement_generator = {}, const␣
↪WindowSizeGenerator window_size_generator = {})
- template<typename RandomIt, typename WindowDisplacementGenerator, typename␣
↪WindowSizeGenerator> std::tuple<RandomIt, RandomIt, RandomIt> transpose_
↪window(const RandomIt genome_first, const RandomIt genome_last, const RandomIt␣
↪target_site, const WindowDisplacementGenerator &window_displacement_generator,␣
↪const WindowSizeGenerator &window_size_generator)
```

6.3.4 Variables

Variable sgpl::EMP_CALL_BY_PACKS_impl

- Defined in file_include_sgpl_spec_StarterConfig.hpp

Variable Documentation

sgpl::EMP_CALL_BY_PACKS_impl (EMP_WRAP_EACH_, EMP_CONFIG__ERROR_CHECK, EMP_DEC_TO_PACK(EMP_

Variable sgpl::tlrand

- Defined in file_include_sgpl_utility_ThreadLocalRandom.hpp

Variable Documentation

thread_local sgpl::internal::*ThreadLocalRandom* `sgpl::tlrand` = {}

6.3.5 Defines

Define SGPL_ALGORITHM_ADVANCE_CORE_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_advance_core.hpp

Define Documentation

SGPL_ALGORITHM_ADVANCE_CORE_HPP_INCLUDE

Define SGPL_ALGORITHM_DRAG_TO_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_drag_to.hpp

Define Documentation

SGPL_ALGORITHM_DRAG_TO_HPP_INCLUDE

Define SGPL_ALGORITHM_EXECUTE_CORE_CYCLES_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_execute_core_cycles.hpp

Define Documentation

SGPL_ALGORITHM_EXECUTE_CORE_CYCLES_HPP_INCLUDE

Define SGPL_ALGORITHM_EXECUTE_CORE_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_execute_core.hpp

Define Documentation

SGPL_ALGORITHM_EXECUTE_CORE_HPP_INCLUDE

Define SGPL_ALGORITHM_EXECUTE_CORE_SLICE_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_execute_core_slice.hpp

Define Documentation

SGPL_ALGORITHM_EXECUTE_CORE_SLICE_HPP_INCLUDE

Define SGPL_ALGORITHM_EXECUTE_CPU_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_execute_cpu.hpp

Define Documentation

SGPL_ALGORITHM_EXECUTE_CPU_HPP_INCLUDE

Define SGPL_ALGORITHM_EXECUTE_CPU_N_CYCLES_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_execute_cpu_n_cycles.hpp

Define Documentation

SGPL_ALGORITHM_EXECUTE_CPU_N_CYCLES_HPP_INCLUDE

Define SGPL_ALGORITHM_EXECUTE_CPU_N_SLICES_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_execute_cpu_n_slices.hpp

Define Documentation

SGPL_ALGORITHM_EXECUTE_CPU_N_SLICES_HPP_INCLUDE

Define SGPL_ALGORITHM_INST_INDEL_COPY_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_inst_indel_copy.hpp

Define Documentation

SGPL_ALGORITHM_INST_INDEL_COPY_HPP_INCLUDE

Define SGPL_ALGORITHM_MODULE_INDEL_COPY_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_module_indel_copy.hpp

Define Documentation

SGPL_ALGORITHM_MODULE_INDEL_COPY_HPP_INCLUDE

Define SGPL_ALGORITHM_MUTATE_BITS_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_mutate_bits.hpp

Define Documentation

SGPL_ALGORITHM_MUTATE_BITS_HPP_INCLUDE

Define SGPL_ALGORITHM_MUTATE_BYTES_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_mutate_bytes.hpp

Define Documentation

SGPL_ALGORITHM_MUTATE_BYTES_HPP_INCLUDE

Define SGPL_ALGORITHM_NEXT_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_next.hpp

Define Documentation

SGPL_ALGORITHM_NEXT_HPP_INCLUDE

Define SGPL_ALGORITHM_PREV_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_prev.hpp

Define Documentation

SGPL_ALGORITHM_PREV_HPP_INCLUDE

Define SGPL_ALGORITHM_SLIDE_N_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_slide_n.hpp

Define Documentation

SGPL_ALGORITHM_SLIDE_N_HPP_INCLUDE

Define SGPL_ALGORITHM_SLIDE_TO_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_slide_to.hpp

Define Documentation

SGPL_ALGORITHM_SLIDE_TO_HPP_INCLUDE

Define SGPL_ALGORITHM_SLOPPY_COPY_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_sloppy_copy.hpp

Define Documentation

SGPL_ALGORITHM_SLOPPY_COPY_HPP_INCLUDE

Define SGPL_ALGORITHM_TRANSPOSE_WINDOW_HPP_INCLUDE

- Defined in file_include_sgpl_algorithm_transpose_window.hpp

Define Documentation

SGPL_ALGORITHM_TRANSPOSE_WINDOW_HPP_INCLUDE

Define sgpl_always_assert

- Defined in file_include_sgpl_debug_sgpl_always_assert.hpp

Define Documentation

sgpl_always_assert(*expr*, ...)

Define sgpl_always_error

- Defined in file_include_sgpl_debug_sgpl_always_error.hpp

Define Documentation

sgpl_always_error(...)

Define sgpl_assert

- Defined in file_include_sgpl_debug_sgpl_assert.hpp

Define Documentation

sgpl_assert(*expr*, ...)

Define SGPL_BYTE_ENUMERATION

- Defined in file_include_sgpl_utility_ByteEnumeration.hpp

Define Documentation

SGPL_BYTE_ENUMERATION

Define SGPL_CASE_PAYLOAD

- Defined in file_include_sgpl_algorithm_advance_core.hpp

Define Documentation

SGPL_CASE_PAYLOAD(*N*)

Define SGPL_DEBUG_SGPL_ALWAYS_ASSERT_HPP_INCLUDE

- Defined in file_include_sgpl_debug_sgpl_always_assert.hpp

Define Documentation

SGPL_DEBUG_SGPL_ALWAYS_ASSERT_HPP_INCLUDE

Define SGPL_DEBUG_SGPL_ALWAYS_ERROR_HPP_INCLUDE

- Defined in file_include_sgpl_debug_sgpl_always_error.hpp

Define Documentation

SGPL_DEBUG_SGPL_ALWAYS_ERROR_HPP_INCLUDE

Define SGPL_DEBUG_SGPL_ASSERT_HPP_INCLUDE

- Defined in file_include_sgpl_debug_sgpl_assert.hpp

Define Documentation

SGPL_DEBUG_SGPL_ASSERT_HPP_INCLUDE

Define SGPL_DEBUG_SGPL_ERROR_HPP_INCLUDE

- Defined in file_include_sgpl_debug_sgpl_error.hpp

Define Documentation

SGPL_DEBUG_SGPL_ERROR_HPP_INCLUDE

Define SGPL_DEBUG_SGPL_STRINGIFY_HPP_INCLUDE

- Defined in file_include_sgpl_debug_SGPL_STRINGIFY.hpp

Define Documentation

SGPL_DEBUG_SGPL_STRINGIFY_HPP_INCLUDE

Define sgpl_error

- Defined in file_include_sgpl_debug_sgpl_error.hpp

Define Documentation

sgpl_error(...)

Define SGPL_HARDWARE_CORE_HPP_INCLUDE

- Defined in file_include_sgpl_hardware_Core.hpp

Define Documentation

SGPL_HARDWARE_CORE_HPP_INCLUDE

Define SGPL_HARDWARE_CPU_HPP_INCLUDE

- Defined in file_include_sgpl_hardware_Cpu.hpp

Define Documentation

SGPL_HARDWARE_CPU_HPP_INCLUDE

Define SGPL_HARDWARE_JUMPTABLE_HPP_INCLUDE

- Defined in file_include_sgpl_hardware_JumpTable.hpp

Define Documentation

SGPL_HARDWARE_JUMPTABLE_HPP_INCLUDE

Define SGPL_INTROSPECTION_COUNT_CORES_WITH_MODULE_IDX_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_count_cores_with_module_idx.hpp

Define Documentation

SGPL_INTROSPECTION_COUNT_CORES_WITH_MODULE_IDX_HPP_INCLUDE

Define SGPL_INTROSPECTION_COUNT_INSTRUCTIONS_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_count_instructions.hpp

Define Documentation

SGPL_INTROSPECTION_COUNT_INSTRUCTIONS_HPP_INCLUDE

Define SGPL_INTROSPECTION_COUNT_MODULES_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_count_modules.hpp

Define Documentation

SGPL_INTROSPECTION_COUNT_MODULES_HPP_INCLUDE

Define SGPL_INTROSPECTION_COUNT_NOP_INSTRUCTIONS_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_count_nop_instructions.hpp

Define Documentation

SGPL_INTROSPECTION_COUNT_NOP_INSTRUCTIONS_HPP_INCLUDE

Define SGPL_INTROSPECTION_COUNT_OP_INSTRUCTIONS_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_count_op_instructions.hpp

Define Documentation

```
SGPL_INTROSPECTION_COUNT_OP_INSTRUCTIONS_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_ENUMERATE_MODULE_IDS_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_enumerate_module_ids.hpp

Define Documentation

```
SGPL_INTROSPECTION_ENUMERATE_MODULE_IDS_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_GET_CUR_MODULE_IDX_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_get_cur_module_idx.hpp

Define Documentation

```
SGPL_INTROSPECTION_GET_CUR_MODULE_IDX_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_GET_MODULE_LENGTH_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_get_module_length.hpp

Define Documentation

```
SGPL_INTROSPECTION_GET_MODULE_LENGTH_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_GET_MODULE_POS_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_get_module_pos.hpp

Define Documentation

```
SGPL_INTROSPECTION_GET_MODULE_POS_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_GET_MODULE_REGULATOR_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_get_module_regulator.hpp

Define Documentation

```
SGPL_INTROSPECTION_GET_MODULE_REGULATOR_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_GET_MODULE_TAG_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_get_module_tag.hpp

Define Documentation

```
SGPL_INTROSPECTION_GET_MODULE_TAG_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_MAKE_MODULE_MASK_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_make_module_mask.hpp

Define Documentation

```
SGPL_INTROSPECTION_MAKE_MODULE_MASK_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_SUMMARIZE_MODULE_EXPRESSION_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_summarize_module_expression.hpp

Define Documentation

```
SGPL_INTROSPECTION_SUMMARIZE_MODULE_EXPRESSION_HPP_INCLUDE
```

Define SGPL_INTROSPECTION_SUMMARIZE_MODULE_REGULATION_HPP_INCLUDE

- Defined in file_include_sgpl_introspection_summarize_module_regulation.hpp

Define Documentation

```
SGPL_INTROSPECTION_SUMMARIZE_MODULE_REGULATION_HPP_INCLUDE
```

Define SGPL_LIBRARY_OPLIBRARY_HPP_INCLUDE

- Defined in file_include_sgpl_library_OpLibrary.hpp

Define Documentation

SGPL_LIBRARY_OPLIBRARY_HPP_INCLUDE

Define SGPL_LIBRARY_OPLIBRARYCOUPLER_HPP_INCLUDE

- Defined in file_include_sgpl_library_OpLibraryCoupler.hpp

Define Documentation

SGPL_LIBRARY_OPLIBRARYCOUPLER_HPP_INCLUDE

Define SGPL_LIBRARY_OPLOOKUP_HPP_INCLUDE

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_LIBRARY_OPLOOKUP_HPP_INCLUDE

Define SGPL_LIBRARY_PREFAB_ARITHMETICOPLIBRARY_HPP_INCLUDE

- Defined in file_include_sgpl_library_prefab_ArithmeticOpLibrary.hpp

Define Documentation

SGPL_LIBRARY_PREFAB_ARITHMETICOPLIBRARY_HPP_INCLUDE

Define SGPL_LIBRARY_PREFAB_COMPLETEOPLIBRARY_HPP_INCLUDE

- Defined in file_include_sgpl_library_prefab_CompleteOpLibrary.hpp

Define Documentation

SGPL_LIBRARY_PREFAB_COMPLETEOPLIBRARY_HPP_INCLUDE

Define SGPL_LIBRARY_PREFAB_CONTROLFLOWOPLIBRARY_HPP_INCLUDE

- Defined in file_include_sgpl_library_prefab_ControlFlowOpLibrary.hpp

Define Documentation

SGPL_LIBRARY_PREFAB_CONTROLFLOWOPLIBRARY_HPP_INCLUDE

Define SGPL_LIBRARY_PREFAB_NOPOPLIBRARY_HPP_INCLUDE

- Defined in file_include_sgpl_library_prefab_NopOpLibrary.hpp

Define Documentation

SGPL_LIBRARY_PREFAB_NOPOPLIBRARY_HPP_INCLUDE

Define SGPL_LIBRARY_PREFAB_PREFAB_HPP_INCLUDE

- Defined in file_include_sgpl_library_prefab_prefab.hpp

Define Documentation

SGPL_LIBRARY_PREFAB_PREFAB_HPP_INCLUDE

Define SGPL_LIBRARY_PREFAB_SANSLOCALREGULATIONOPLIBRARY_HPP_INCLUDE

- Defined in file_include_sgpl_library_prefab_SansLocalRegulationOpLibrary.hpp

Define Documentation

SGPL_LIBRARY_PREFAB_SANSLOCALREGULATIONOPLIBRARY_HPP_INCLUDE

Define SGPL_LIBRARY_PREFAB_SANSREGULATIONOPLIBRARY_HPP_INCLUDE

- Defined in file_include_sgpl_library_prefab_SansRegulationOpLibrary.hpp

Define Documentation

SGPL_LIBRARY_PREFAB_SANSREGULATIONOPLIBRARY_HPP_INCLUDE

Define SGPL_MORPH_NOP_OUT_INSTRUCTION_CATEGORY_HPP_INCLUDE

- Defined in file_include_sgpl_morph_nop_out_instruction_category.hpp

Define Documentation

SGPL_MORPH_NOP_OUT_INSTRUCTION_CATEGORY_HPP_INCLUDE

Define SGPL_MORPH_NOP_OUT_INSTRUCTIONS_HPP_INCLUDE

- Defined in file_include_sgpl_morph_nop_out_instructions.hpp

Define Documentation

SGPL_MORPH_NOP_OUT_INSTRUCTIONS_HPP_INCLUDE

Define SGPL_MORPH_NOP_OUT_MODULE_HPP_INCLUDE

- Defined in file_include_sgpl_morph_nop_out_module.hpp

Define Documentation

SGPL_MORPH_NOP_OUT_MODULE_HPP_INCLUDE

Define SGPL_MORPH_NOP_OUT_MODULES_HPP_INCLUDE

- Defined in file_include_sgpl_morph_nop_out_modules.hpp

Define Documentation

SGPL_MORPH_NOP_OUT_MODULES_HPP_INCLUDE

Define SGPL_MORPH_NOP_OUT_NTH_OP_HPP_INCLUDE

- Defined in file_include_sgpl_morph_nop_out_nth_op.hpp

Define Documentation

SGPL_MORPH_NOP_OUT_NTH_OP_HPP_INCLUDE

Define SGPL_MUTATE_MUTATE_COPY_HPP_INCLUDE

- Defined in file_include_sgpl_mutate_mutate_copy.hpp

Define Documentation

SGPL_MUTATE_MUTATE_COPY_HPP_INCLUDE

Define SGPL_MUTATE_POINT_MUTATE_HPP_INCLUDE

- Defined in file_include_sgpl_mutate_point_mutate.hpp

Define Documentation

SGPL_MUTATE_POINT_MUTATE_HPP_INCLUDE

Define SGPL_MUTATE_SEQUENCE_MUTATE_COPY_HPP_INCLUDE

- Defined in file_include_sgpl_mutate_sequence_mutate_copy.hpp

Define Documentation

SGPL_MUTATE_SEQUENCE_MUTATE_COPY_HPP_INCLUDE

Define SGPL_MUTATE_TRANSPOSE_INVERT_MUTATE_HPP_INCLUDE

- Defined in file_include_sgpl_mutate_transpose_invert_mutate.hpp

Define Documentation

SGPL_MUTATE_TRANSPOSE_INVERT_MUTATE_HPP_INCLUDE

Define SGPL_NOP_OP_CODE_PAYLOAD

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_NOP_OP_CODE_PAYLOAD(*N*)

Define SGPL_OP_GET_CATEGORIES

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_OP_GET_CATEGORIES (*N*)

Define SGPL_OP_GET_DESCRIPTORS

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_OP_GET_DESCRIPTORS (*N*)

Define SGPL_OP_LOOKUP_PAYLOAD

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_OP_LOOKUP_PAYLOAD (*N*)

Define SGPL_OP_NAME_PAYLOAD

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_OP_NAME_PAYLOAD (*N*)

Define SGPL_OP_NUM_RNG_TOUCHES_PAYLOAD

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_OP_NUM_RNG_TOUCHES_PAYLOAD (*N*)

Define SGPL_OP_PREVALENCE_PAYLOAD

- Defined in file_include_sgpl_library_OpLookup.hpp

Define Documentation

SGPL_OP_PREVALENCE_PAYLOAD(*N*)

Define SGPL_OPERATIONS_ACTIONS_ACTIONS_HPP_INCLUDE

- Defined in file_include_sgpl_operations_actions_actions.hpp

Define Documentation

SGPL_OPERATIONS_ACTIONS_ACTIONS_HPP_INCLUDE

Define SGPL_OPERATIONS_ACTIONS_FORKIF_HPP_INCLUDE

- Defined in file_include_sgpl_operations_actions_ForkIf.hpp

Define Documentation

SGPL_OPERATIONS_ACTIONS_FORKIF_HPP_INCLUDE

Define SGPL_OPERATIONS_ACTIONS_NOP_HPP_INCLUDE

- Defined in file_include_sgpl_operations_actions_Nop.hpp

Define Documentation

SGPL_OPERATIONS_ACTIONS_NOP_HPP_INCLUDE

Define SGPL_OPERATIONS_ACTIONS_TERMINATEIF_HPP_INCLUDE

- Defined in file_include_sgpl_operations_actions_TerminateIf.hpp

Define Documentation

SGPL_OPERATIONS_ACTIONS_TERMINATEIF_HPP_INCLUDE

Define SGPL_OPERATIONS_BINARY_ADD_HPP_INCLUDE

- Defined in file_include_sgpl_operations_binary_Add.hpp

Define Documentation

SGPL_OPERATIONS_BINARY_ADD_HPP_INCLUDE

Define SGPL_OPERATIONS_BINARY_BINARY_HPP_INCLUDE

- Defined in file_include_sgpl_operations_binary_binary.hpp

Define Documentation

SGPL_OPERATIONS_BINARY_BINARY_HPP_INCLUDE

Define SGPL_OPERATIONS_BINARY_DIVIDE_HPP_INCLUDE

- Defined in file_include_sgpl_operations_binary_Divide.hpp

Define Documentation

SGPL_OPERATIONS_BINARY_DIVIDE_HPP_INCLUDE

Define SGPL_OPERATIONS_BINARY_MODULO_HPP_INCLUDE

- Defined in file_include_sgpl_operations_binary_Modulo.hpp

Define Documentation

SGPL_OPERATIONS_BINARY_MODULO_HPP_INCLUDE

Define SGPL_OPERATIONS_BINARY_MULTIPLY_HPP_INCLUDE

- Defined in file_include_sgpl_operations_binary_Multiply.hpp

Define Documentation

SGPL_OPERATIONS_BINARY_MULTIPLY_HPP_INCLUDE

Define SGPL_OPERATIONS_BINARY_SUBTRACT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_binary_Subtract.hpp

Define Documentation

SGPL_OPERATIONS_BINARY_SUBTRACT_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_BITWISE_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_bitwise.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_BITWISE_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_BITWISEAND_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_BitwiseAnd.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_BITWISEAND_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_BITWISENOT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_BitwiseNot.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_BITWISENOT_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_BITWISEOR_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_BitwiseOr.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_BITWISEOR_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_BITWISESHIFT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_BitwiseShift.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_BITWISESHIFT_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_BITWISEXOR_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_BitwiseXor.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_BITWISEXOR_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_COUNTONES_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_CountOnes.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_COUNTONES_HPP_INCLUDE

Define SGPL_OPERATIONS_BITWISE_RANDOMFILL_HPP_INCLUDE

- Defined in file_include_sgpl_operations_bitwise_RandomFill.hpp

Define Documentation

SGPL_OPERATIONS_BITWISE_RANDOMFILL_HPP_INCLUDE

Define SGPL_OPERATIONS_COMPARISON_COMPARISON_HPP_INCLUDE

- Defined in file_include_sgpl_operations_comparison_comparison.hpp

Define Documentation

SGPL_OPERATIONS_COMPARISON_COMPARISON_HPP_INCLUDE

Define SGPL_OPERATIONS_COMPARISON_EQUAL_HPP_INCLUDE

- Defined in file_include_sgpl_operations_comparison_Equal.hpp

Define Documentation

SGPL_OPERATIONS_COMPARISON_EQUAL_HPP_INCLUDE

Define SGPL_OPERATIONS_COMPARISON_GREATERTHAN_HPP_INCLUDE

- Defined in file_include_sgpl_operations_comparison_GreaterThan.hpp

Define Documentation

SGPL_OPERATIONS_COMPARISON_GREATERTHAN_HPP_INCLUDE

Define SGPL_OPERATIONS_COMPARISON_LESSTHAN_HPP_INCLUDE

- Defined in file_include_sgpl_operations_comparison_LessThan.hpp

Define Documentation

SGPL_OPERATIONS_COMPARISON_LESSTHAN_HPP_INCLUDE

Define SGPL_OPERATIONS_COMPARISON_LOGICALAND_HPP_INCLUDE

- Defined in file_include_sgpl_operations_comparison_LogicalAnd.hpp

Define Documentation

SGPL_OPERATIONS_COMPARISON_LOGICALAND_HPP_INCLUDE

Define SGPL_OPERATIONS_COMPARISON_LOGICALOR_HPP_INCLUDE

- Defined in file_include_sgpl_operations_comparison_LogicalOr.hpp

Define Documentation

SGPL_OPERATIONS_COMPARISON_LOGICALOR_HPP_INCLUDE

Define SGPL_OPERATIONS_COMPARISON_NOTEQUAL_HPP_INCLUDE

- Defined in file_include_sgpl_operations_comparison_NotEqual.hpp

Define Documentation

SGPL_OPERATIONS_COMPARISON_NOTEQUAL_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_GLOBAL_ANCHOR_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_Anchor.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_GLOBAL_ANCHOR_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_GLOBAL_FLOW_GLOBAL_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_flow_global.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_GLOBAL_FLOW_GLOBAL_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_GLOBAL_JUMPIF_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_JumpIf.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_GLOBAL_JUMPIF_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_GLOBAL_JUMPIFNOT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_JumpIfNot.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_GLOBAL_JUMPIFNOT_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORADJ_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_RegulatorAdj.hpp

Define Documentation

```
SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORADJ_HPP_INCLUDE
```

Define SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORDECAY_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_RegulatorDecay.hpp

Define Documentation

```
SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORDECAY_HPP_INCLUDE
```

Define SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORGET_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_RegulatorGet.hpp

Define Documentation

```
SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORGET_HPP_INCLUDE
```

Define SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORSET_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_global_RegulatorSet.hpp

Define Documentation

```
SGPL_OPERATIONS_FLOW_GLOBAL_REGULATORSET_HPP_INCLUDE
```

Define SGPL_OPERATIONS_FLOW_LOCAL_ANCHOR_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_Anchor.hpp

Define Documentation

```
SGPL_OPERATIONS_FLOW_LOCAL_ANCHOR_HPP_INCLUDE
```

Define SGPL_OPERATIONS_FLOW_LOCAL_FLOW_LOCAL_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_flow_local.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_LOCAL_FLOW_LOCAL_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_LOCAL_JUMPIF_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_JumpIf.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_LOCAL_JUMPIF_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_LOCAL_JUMPIFNOT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_JumpIfNot.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_LOCAL_JUMPIFNOT_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_LOCAL_REGULATORADJ_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_RegulatorAdj.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_LOCAL_REGULATORADJ_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_LOCAL_REGULATORDECAY_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_RegulatorDecay.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_LOCAL_REGULATORDECAY_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_LOCAL_REGULATORGET_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_RegulatorGet.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_LOCAL_REGULATORGET_HPP_INCLUDE

Define SGPL_OPERATIONS_FLOW_LOCAL_REGULATORSET_HPP_INCLUDE

- Defined in file_include_sgpl_operations_flow_local_RegulatorSet.hpp

Define Documentation

SGPL_OPERATIONS_FLOW_LOCAL_REGULATORSET_HPP_INCLUDE

Define SGPL_OPERATIONS_OPERATIONS_HPP_INCLUDE

- Defined in file_include_sgpl_operations_operations.hpp

Define Documentation

SGPL_OPERATIONS_OPERATIONS_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_DECREMENT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_Decrement.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_DECREMENT_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_INCREMENT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_Increment.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_INCREMENT_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_NEGATE_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_Negate.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_NEGATE_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_NOT_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_Not.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_NOT_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_RANDOMBOOL_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_RandomBool.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_RANDOMBOOL_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_RANDOMDRAW_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_RandomDraw.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_RANDOMDRAW_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_TERMINAL_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_Terminal.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_TERMINAL_HPP_INCLUDE

Define SGPL_OPERATIONS_UNARY_UNARY_HPP_INCLUDE

- Defined in file_include_sgpl_operations_unary_unary.hpp

Define Documentation

SGPL_OPERATIONS_UNARY_UNARY_HPP_INCLUDE

Define SGPL_PROGRAM_GLOBALANCHORITERATOR_HPP_INCLUDE

- Defined in file_include_sgpl_program_GlobalAnchorIterator.hpp

Define Documentation

SGPL_PROGRAM_GLOBALANCHORITERATOR_HPP_INCLUDE

Define SGPL_PROGRAM_INSTRUCTION_HPP_INCLUDE

- Defined in file_include_sgpl_program_Instruction.hpp

Define Documentation

SGPL_PROGRAM_INSTRUCTION_HPP_INCLUDE

Define SGPL_PROGRAM_LOAD_PROGRAM_HPP_INCLUDE

- Defined in file_include_sgpl_program_load_program.hpp

Define Documentation

SGPL_PROGRAM_LOAD_PROGRAM_HPP_INCLUDE

Define SGPL_PROGRAM_OPCODERECTIFIER_HPP_INCLUDE

- Defined in file_include_sgpl_program_OpCodeRectifier.hpp

Define Documentation

SGPL_PROGRAM_OPCODERECTIFIER_HPP_INCLUDE

Define SGPL_PROGRAM_PROGRAM_HPP_INCLUDE

- Defined in file_include_sgpl_program_Program.hpp

Define Documentation

SGPL_PROGRAM_PROGRAM_HPP_INCLUDE

Define SGPL_SPEC_INSTRANGECOPIER_DEFAULT_HPP_INCLUDE

- Defined in file_include_sgpl_spec_InstRangeCopier_Default.hpp

Define Documentation

SGPL_SPEC_INSTRANGECOPIER_DEFAULT_HPP_INCLUDE

Define SGPL_SPEC_INSTRANGECOPIER_INDEL_HPP_INCLUDE

- Defined in file_include_sgpl_spec_InstRangeCopier_Indel.hpp

Define Documentation

SGPL_SPEC_INSTRANGECOPIER_INDEL_HPP_INCLUDE

Define SGPL_SPEC_INSTRANGECOPIER_PERFECT_HPP_INCLUDE

- Defined in file_include_sgpl_spec_InstRangeCopier_Perfect.hpp

Define Documentation

SGPL_SPEC_INSTRANGECOPIER_PERFECT_HPP_INCLUDE

Define SGPL_SPEC_SPEC_HPP_INCLUDE

- Defined in file_include_sgpl_spec_Spec.hpp

Define Documentation

SGPL_SPEC_SPEC_HPP_INCLUDE

Define SGPL_SPEC_STARTERCONFIG_HPP_INCLUDE

- Defined in file_include_sgpl_spec_StarterConfig.hpp

Define Documentation

```
SGPL_SPEC_STARTERCONFIG_HPP_INCLUDE
```

Define SGPL_SPEC_TRANSPOSEWINDOWDISPLACEMENTGENERATOR_DEFAULT_HPP_INCLUDE

- Defined in file_include_sgpl_spec_TransposeWindowDisplacementGenerator_Default.hpp

Define Documentation

```
SGPL_SPEC_TRANSPOSEWINDOWDISPLACEMENTGENERATOR_DEFAULT_HPP_INCLUDE
```

Define SGPL_SPEC_TRANSPOSEWINDOWDISPLACEMENTGENERATOR_PARETO_HPP_INCLUDE

- Defined in file_include_sgpl_spec_TransposeWindowDisplacementGenerator_Pareto.hpp

Define Documentation

```
SGPL_SPEC_TRANSPOSEWINDOWDISPLACEMENTGENERATOR_PARETO_HPP_INCLUDE
```

Define SGPL_SPEC_TRANSPOSEWINDOWSIZEGENERATOR_DEFAULT_HPP_INCLUDE

- Defined in file_include_sgpl_spec_TransposeWindowSizeGenerator_Default.hpp

Define Documentation

```
SGPL_SPEC_TRANSPOSEWINDOWSIZEGENERATOR_DEFAULT_HPP_INCLUDE
```

Define SGPL_SPEC_TRANSPOSEWINDOWSIZEGENERATOR_PARETO_HPP_INCLUDE

- Defined in file_include_sgpl_spec_TransposeWindowSizeGenerator_Pareto.hpp

Define Documentation

```
SGPL_SPEC_TRANSPOSEWINDOWSIZEGENERATOR_PARETO_HPP_INCLUDE
```

Define SGPL_STRINGIFY

- Defined in file_include_sgpl_debug_SGPL_STRINGIFY.hpp

Define Documentation

SGPL_STRINGIFY(*x*)

Define SGPL_UITSL_NAMESPACE

- Defined in file_include_sgpl_debug_sgpl_always_assert.hpp

Define Documentation

SGPL_UITSL_NAMESPACE()

Define SGPL_UTILITY_BYTEENUMERATION_HPP_INCLUDE

- Defined in file_include_sgpl_utility_ByteEnumeration.hpp

Define Documentation

SGPL_UTILITY_BYTEENUMERATION_HPP_INCLUDE

Define SGPL_UTILITY_CAPPEDOUTPUTITERATOR_HPP_INCLUDE

- Defined in file_include_sgpl_utility_CappedOutputIterator.hpp

Define Documentation

SGPL_UTILITY_CAPPEDOUTPUTITERATOR_HPP_INCLUDE

Define SGPL_UTILITY_CAPPEDSET_HPP_INCLUDE

- Defined in file_include_sgpl_utility_CappedSet.hpp

Define Documentation

SGPL_UTILITY_CAPPEDSET_HPP_INCLUDE

Define SGPL_UTILITY_COUNT_OPERATION_RANDOM_TOUCHES_HPP_INCLUDE

- Defined in file_include_sgpl_utility_count_operation_random_touches.hpp

Define Documentation

SGPL_UTILITY_COUNT_OPERATION_RANDOM_TOUCHES_HPP_INCLUDE

Define SGPL_UTILITY_COUNT_THREAD_LOCAL_RANDOM_TOUCHES_HPP_INCLUDE

- Defined in file_include_sgpl_utility_count_thread_local_random_touches.hpp

Define Documentation

SGPL_UTILITY_COUNT_THREAD_LOCAL_RANDOM_TOUCHES_HPP_INCLUDE

Define SGPL_UTILITY_COUNTINGITERATOR_HPP_INCLUDE

- Defined in file_include_sgpl_utility_CountingIterator.hpp

Define Documentation

SGPL_UTILITY_COUNTINGITERATOR_HPP_INCLUDE

Define SGPL_UTILITY_DO_RANDOM_WALK_APPROXIMATION_HPP_INCLUDE

- Defined in file_include_sgpl_utility_do_random_walk_approximation.hpp

Define Documentation

SGPL_UTILITY_DO_RANDOM_WALK_APPROXIMATION_HPP_INCLUDE

Define SGPL_UTILITY_DO_RANDOM_WALK_EXACT_HPP_INCLUDE

- Defined in file_include_sgpl_utility_do_random_walk_exact.hpp

Define Documentation

SGPL_UTILITY_DO_RANDOM_WALK_EXACT_HPP_INCLUDE

Define SGPL_UTILITY_DO_RANDOM_WALK_INDEXMAP_APPROXIMATION_HPP_INCLUDE

- Defined in file_include_sgpl_utility_do_random_walk_indexmap_approximation.hpp

Define Documentation

```
SGPL_UTILITY_DO_RANDOM_WALK_INDEXMAP_APPROXIMATION_HPP_INCLUDE
```

Define SGPL_UTILITY_DO_RANDOM_WALK_NORMAL_APPROXIMATION_HPP_INCLUDE

- Defined in file_include_sgpl_utility_do_random_walk_normal_approximation.hpp

Define Documentation

```
SGPL_UTILITY_DO_RANDOM_WALK_NORMAL_APPROXIMATION_HPP_INCLUDE
```

Define SGPL_UTILITY_EMPTYTYPE_HPP_INCLUDE

- Defined in file_include_sgpl_utility_EmptyType.hpp

Define Documentation

```
SGPL_UTILITY_EMPTYTYPE_HPP_INCLUDE
```

Define SGPL_UTILITY_GARBLEDOUTPUTITERATOR_HPP_INCLUDE

- Defined in file_include_sgpl_utility_GarbledOutputIterator.hpp

Define Documentation

```
SGPL_UTILITY_GARBLEDOUTPUTITERATOR_HPP_INCLUDE
```

Define SGPL_UTILITY_MEMOIZECTOR_HPP_INCLUDE

- Defined in file_include_sgpl_utility_MemoizeCtor.hpp

Define Documentation

```
SGPL_UTILITY_MEMOIZECTOR_HPP_INCLUDE
```

Define SGPL_UTILITY_RANDOM_BETWEEN_HPP_INCLUDE

- Defined in file_include_sgpl_utility_random_between.hpp

Define Documentation

SGPL_UTILITY_RANDOM_BETWEEN_HPP_INCLUDE

Define SGPL_UTILITY_RANDOM_SIGN_HPP_INCLUDE

- Defined in file_include_sgpl_utility_random_sign.hpp

Define Documentation

SGPL_UTILITY_RANDOM_SIGN_HPP_INCLUDE

Define SGPL_UTILITY_REPEATINGNEGATIVEBINOMIALCOUNTDOWN_HPP_INCLUDE

- Defined in file_include_sgpl_utility_RepeatingNegativeBinomialCountdown.hpp

Define Documentation

SGPL_UTILITY_REPEATINGNEGATIVEBINOMIALCOUNTDOWN_HPP_INCLUDE

Define SGPL_UTILITY_RESEVOIR_HPP_INCLUDE

- Defined in file_include_sgpl_utility_Resevoir.hpp

Define Documentation

SGPL_UTILITY_RESEVOIR_HPP_INCLUDE

Define SGPL_UTILITY_RINGRESEVOIR_HPP_INCLUDE

- Defined in file_include_sgpl_utility_RingResevoir.hpp

Define Documentation

SGPL_UTILITY_RINGRESEVOIR_HPP_INCLUDE

Define SGPL_UTILITY_THREADLOCALRANDOM_HPP_INCLUDE

- Defined in file_include_sgpl_utility_ThreadLocalRandom.hpp

Define Documentation

SGPL_UTILITY_THREADLOCALRANDOM_HPP_INCLUDE

6.3.6 Typedefs

Typedef sgpl::ArithmeticOpLibrary

- Defined in file_include_sgpl_library_prefab_ArithmeticOpLibrary.hpp

Typedef Documentation

using sgpl::**ArithmeticOpLibrary** = sgpl::*OpLibrary*<sgpl::*Add*, sgpl::*Divide*, sgpl::*Multiply*, sgpl::*Subtract*>

Typedef sgpl::CompleteOpLibrary

- Defined in file_include_sgpl_library_prefab_CompleteOpLibrary.hpp

Typedef Documentation

using sgpl::**CompleteOpLibrary** = sgpl::*OpLibrary*<sgpl::*ForkIf*, sgpl::*Nop*<0>, sgpl::*Nop*<1>, sgpl::*Nop*<2>, sgpl::*Termina*

Typedef sgpl::ControlFlowOpLibrary

- Defined in file_include_sgpl_library_prefab_ControlFlowOpLibrary.hpp

Typedef Documentation

using sgpl::**ControlFlowOpLibrary** = sgpl::*OpLibrary*<sgpl::global::*Anchor*, sgpl::global::*JumpIf*, sgpl::global::*JumpIfNot*

Typedef sgpl::InstRangeCopier_Default

- Defined in file_include_sgpl_spec_InstRangeCopier_Default.hpp

Typedef Documentation

using sgpl::**InstRangeCopier_Default** = sgpl::*InstRangeCopier_Perfect*

Typedef sgpl::NopOpLibrary

- Defined in file_include_sgpl_library_prefab_NopOpLibrary.hpp

Typedef Documentation

using sgpl::**NopOpLibrary** = sgpl::*OpLibrary*<sgpl::*Nop*<>>

Typedef sgpl::SansLocalRegulationOpLibrary

- Defined in file_include_sgpl_library_prefab_SansLocalRegulationOpLibrary.hpp

Typedef Documentation

using sgpl::**SansLocalRegulationOpLibrary** = sgpl::*OpLibrary*<sgpl::*ForkIf*, sgpl::*Nop*<>, sgpl::*TerminateIf*, sgpl::*Add*

Typedef sgpl::SansRegulationOpLibrary

- Defined in file_include_sgpl_library_prefab_SansRegulationOpLibrary.hpp

Typedef Documentation

using sgpl::**SansRegulationOpLibrary** = sgpl::*OpLibrary*<sgpl::*ForkIf*, sgpl::*Nop*<>, sgpl::*TerminateIf*, sgpl::*Add*, sgpl::*

Typedef sgpl::TransposeWindowDisplacementGenerator_Default

- Defined in file_include_sgpl_spec_TransposeWindowDisplacementGenerator_Default.hpp

Typedef Documentation

using sgpl::**TransposeWindowDisplacementGenerator_Default** = sgpl::*TransposeWindowDisplacementGenerator_*

Typedef sgpl::TransposeWindowSizeGenerator_Default

- Defined in file_include_sgpl_spec_TransposeWindowSizeGenerator_Default.hpp

Typedef Documentation

using sgpl::**TransposeWindowSizeGenerator_Default** = sgpl::*TransposeWindowSizeGenerator_Pareto*

CHAPTER

SEVEN

CONTRIBUTING

We love your input! We want to make contributing to this project as easy and transparent as possible, whether it's:

- Reporting a bug
- Proposing new features
- Submitting new code
- Writing documentation

7.1 Reporting Bugs

To report bugs, please open an issue on Github clearly stating the observed (buggy) behavior (screenshots are encouraged), the desired behavior, which operating system and compiler you're using (including version), and any additional information that might help us fix the bug. If possible, please provide a minimum example that reproduces the bug.

7.2 Fixing Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

7.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

7.4 Write Documentation

SignalGP Lite could always use more documentation, whether as part of the official SignalGP Lite docs, in docstrings, or even on the web in blog posts, articles, and such.

7.5 Requesting features

We welcome suggestions for how to make this project better! To give us your ideas, open an issue on Github.

7.6 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/mmore500/signalgp-lite/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

7.7 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated.

Please see the *Code of Conduct* for community guidelines for positive behavior.

CHAPTER

EIGHT

CODE OF CONDUCT

8.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

8.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

8.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

8.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

8.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at ofria@msu.edu. The project team will review and investigate all complaints, and will respond in a way that it deems appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

8.6 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at http://contributor-covenant.org/version/1/4

CHAPTER

NINE

CREDITS

9.1 Development Lead

- Matthew Andres Moreno mailto:m.more500@gmail.com

9.2 Contributors

None yet. Why not be the first?

CHAPTER

TEN

SIGNALGP-LITE

- Free software: MIT license
- Documentation: https://signalgp-lite.readthedocs.io
- header-only, namespace-encapsulated software

A genetic programming implementation designed for large-scale artificial life applications. Organized as a header-only C++ library. Inspired by Alex Lalejini's SignalGP.

10.1 Quick Start

This "hello world" example throws together

- a custom hardware peripheral to manage greeting information,
- a custom operation to print a greeting, and
- generation of a random program,
- execution of that random program on a virtual multi-core CPU.

`say-hello.cpp`:

```
#include <iostream>
#include <ratio>
#include <string>

#include "Empirical/include/emp/math/Random.hpp"

#include "sgpl/algorithm/execute_cpu.hpp"
#include "sgpl/spec/Spec.hpp"
#include "sgpl/hardware/Cpu.hpp"
#include "sgpl/library/OpLibraryCoupler.hpp"
#include "sgpl/library/prefab/ControlFlowOpLibrary.hpp"
#include "sgpl/program/Program.hpp"

emp::Random rng;

// custom hardware peripheral, can be written to or read from during execution
struct Peripheral {
  size_t greet_count{};
  std::string name;
};
```

(continues on next page)

(continued from previous page)

```
// custom CPU operation
struct SayHello {

  template<typename Spec>
  static void run(
    sgpl::Core<Spec>&,
    const sgpl::Instruction<Spec>&,
    const sgpl::Program<Spec>&,
    typename Spec::peripheral_t& peripheral
  ) {
    std::cout << "for the " << peripheral.greet_count++ << "th time... ";
    std::cout << "hello there " << peripheral.name << '\n';
  }

  static std::string name() { return "SayHello"; }

  static size_t prevalence() { return 1; }

};

// extends prefab ControlFlowOpLibrary with SayHello operation
using library_t = sgpl::OpLibraryCoupler<sgpl::ControlFlowOpLibrary, SayHello>;

// custom compile-time configurator type
using spec_t = sgpl::Spec<library_t, Peripheral>;

int main() {

  sgpl::Cpu<spec_t> cpu;
  Peripheral peripheral;
  peripheral.name = "Grace Hopper";

  sgpl::Program<spec_t> program{ 100 }; // randomly generated, 100 instructions

  cpu.InitializeAnchors( program ); // load program onto CPU

  // generate random signals to launch available virtual cores
  while ( cpu.TryLaunchCore( emp::BitSet<64>(rng) ) ) ;

  // execute up to one thousand instructions
  sgpl::execute_cpu<spec_t>( std::kilo::num, cpu, program, peripheral );

}
```

compile:

```
g++ --std=c++17 -Iinclude/ -Ithird-party/ say-hello.cpp -o say-hello.out
```

run:

```
./say-hello.out
```

10.2 Benchmarks

signalgp-lite provides several-times speedup over the current "vanilla" SignalGP implementation.

Speedup of mean instruction execution time provided by signalgp-lite compared to vanilla SignalGP. Speedup is measured for random programs generated from different subsets of instructions ("libraries") over different-size populations of virtual CPUs ("num agents").

For randomly-generated programs composed of arbitrary instructions, signalgp-lite approaches a virtual instruction execution rate of around 10Mhz on a 3.5Ghz processor. Virtual nop instructions execute at rate of around 200Mhz.

Wall clock timings of twenty randomly-generated programs composed of instructions from different libraries.

Timings for `nop` and `arithmetic` libraries report the mean time to execute sixteen instructions on one core. Timings for `complete` and `sans_regulation` libraries report timings for executing sixteen instructions, one each across sixteen virtual threads. (`sans_regulation` refers to the `complete` library with tag-matching regulation disabled.)

These results are associated with commit c10ed70, measured at 1602292830 seconds since epoch. Details on the machine used to perform these benchmarks are available via Open Science Framework, e.g., https://osf.io/hu8m2/. mimalloc memory allocator.

Microbenchmarks are performed, graphed, and uploaded as part of the project's CI build, so check the project's OSF page for up-to-the-minute profiling information!

10.3 Credits

This library draws heavily on Alex Lalejini's work with SignalGP.

This package was created with Cookiecutter and the devosoft/cookiecutter-empirical-project project template.

This package uses Empirical, a library of tools for scientific software development, with emphasis on also being able to build web interfaces using Emscripten.

INDEX

S