Switch-Compiler

1.0

Generated by Doxygen 1.8.10

Tue Dec 29 2015 15:08:43

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 mapper.src.flexpipe.flexpipe_configuration.FlexpipeConfiguration Class Reference

Public Member Functions

- def **__init__** (self, program, switch, preprocess, version)
- def **displayInitialConditions** (self)
- def getPerLogAssignInfo (self)
- def getEarliestStageFromAssign (self, assignInfo)
- def **configure** (self, startRow, numberOfRows)
- def display (self)
- def **getColorsFromTableGroups** (self, tableGroups)
- def **showPic**

Public Attributes

- **logger**
- **program**
- **switch**
- **preprocess**
- **version**
- **da**
- **stMax**
- **logMax**
- **startRow**
- **numberOfRows**
- **colorNames**

3.1.1 Detailed Description

```
Module to describe and log a FlexPipe switch configuration
for a given program
```

3.1.2 Member Function Documentation

3.1.2.1 def mapper.src.flexpipe.flexpipe_configuration.FlexpipeConfiguration.display (*self*)

```
Display configuration- tables in each memory type,
in each stage.
```

3.1.2.2 def mapper.src.flexpipe.flexpipe_configuration.FlexpipeConfiguration.getEarliestStageFromAssign (*self, assignInfo*)

```
Returns a mapping from tables
to information about the first stage they could have started
in- a table must start after all the tables it depends on
have been completely assigned. The informations is a list of
 lists [@st, @dep, @prev, @prevEnd] which should be read as
Table xx can't start before stage @st because of @dependency
on table @prev which ends in stage @prevEnd, one for each
@prev that the table depends on.
```

3.1.2.3 def mapper.src.flexpipe.flexpipe_configuration.FlexpipeConfiguration.getPerLogAssignInfo (*self*)

```
 Return mapping from table to assignment info
  'start': start stage (or -1 if not valid)
  'end': end stage (or -1 if not valid)
```

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

- src/flexpipe/flexpipe_configuration.py

3.2 mapper.src.flexpipe.flexpipe_dependency_analysis.FlexpipeDependencyAnalysis Class Reference

Public Member Functions

- def **__init__** (self, program)
- def **getDigraph** (self)
- def **getDigraphFromStartNegative** (self)
- def **makePath** (self, nodes, gr)
- def **showPath**
- def **getPerLogProgramInfo** (self)
- def **getDigraphFromEndPositive** (self)
- def **getDigraphFromStartPositive** (self)
- def **showCriticalPath** (self)

Public Attributes

- **logger**
- **program**
- **logMax**

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

- src/flexpipe/flexpipe_dependency_analysis.py

3.3 mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler Class Reference

Public Member Functions

- def __init__
- def **iOSI** (self, mem, order, sl)
- def **iLSI** (self, mem, log, sl)

- def **iLSt** (self, log, st)
- def **iSlOL** (self, mem, sl, order, log)
- def **setDimensionSizes** (self)
- def **computeSum** (self, dictCount)
- def **newVar**
- def **newConstr** (self, expr)
- def blockOverlap (self, mem, sl1, log, sl2)
- def blocksInStage (self, mem, st)
- def blocksInSameStageAs (self, mem, sl)
- def **flattenOrdSl** (self, d, mem)
- def **flattenLogSl** (self, d, mem)
- def **flattenLogSt** (self, d, mem)
- def fillModel (self, startRowDict, numberOfRowsDict, model)
- def setup (self, program, switch, preprocess)
- def **setupVariables** (self, program, switch, preprocess)
- def setupConstraints
- def **setupStartingDict** (self)
- def **checkSolution**
- def **setupConstraintsForProductsAndBinarys**
- def **setupStartAndEndStagesVariables** (self)
- def **getXXOfLog**
- def **addProductConstraint** (self, prod, cont, binary, ub)
- def **addProductBinaryConstraint** (self, prod, binary1, binary2)
- def **addBinaryConstraint** (self, binary, cont, ub, lb)
- def setLb (self)
- def numberOfRowsBounds
- def **getXXOfOrd**
- def **getStartingAndEndingStages**
- def **overlapConstraint**
- def wordLayoutConstraint
- def **oneOrderPerLogConstraint**
- def **maximumTablesPerBlockConstraint**
- def **useMemoryConstraint**
- def **assignmentConstraint**
- def dependencyConstraint
- def **capacityConstraintByBlock**
- def **capacityConstraintByRow**
- def **setIlpResults** (self)
- def solve (self, program, switch, preprocess)

Public Attributes

- **logger**
- **granLb**
- **greedyVersion**
- **timeLimit**
- **outputFileName**
- **checking**
- **dictNumVariables**
- **numVariables**
- **dictNumConstraints**
- **numConstraints**
- **testNumConstraints**
- **dimensionSizes**

- **variablesByName**
- **variables**
- **names**
- **varUb**
- **varLb**
- **varType**
- **program**
- **switch**
- **preprocess**
- orderMax

  Constants.
- **stMax**
- **slMax**
- **totalSlMax**
- **logMax**
- **rowMax**
- **maxSlMax**
- **wordMax**
- results

  Variables.
- **m**
- **word**
- **blocks**
- **totalBlocks**
- **totalBlocksBinary**
- **startRow**
- **startRowUnit**
- **numberOfRows**
- **numberOfRowsBound**
- **numberOfRowsUnit**
- **numberOfRowsBinary**
- **startRowTimesNumberOfRowsBinary**
- **firstRowOfLog**
- **numberOfRowsOfLog**
- **numberOfRowsOfLogBinary**
- **firstRowOfOrd**
- **numberOfRowsOfOrd**
- **numberOfRowsOfOrdBinary**
- **ordToLog**
- **firstRowOfLogTimesOrdToLog**
- **numberOfRowsOfLogTimesOrdToLog**
- **blockAllMemBin**
- **ub**
- **lb**
- **startingDict**
- **configs**
- **violated**
- **startAllMem**
- **startAllMemTimesBlockAllMemBin**
- **endAllMem**
- **endAllMemTimesBlockAllMemBin**

3.3.1 Constructor & Destructor Documentation

3.3.1.1 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.__init__ (*self, greedyVersion =* None, *timeLimit =* None, *outputFileName =* None, *granLb =* None)

```
Initialize compiler with CPLEX parameters @timeLimit and
 @greedyVersion is 'lpt'/ None. If none, no starting solution
 is provided for CPLEX.
 @outputFileName is name used to save file containing info
 about CPLEX parameters used.
 @granLb is mapping from memory type to minimum physical table
 size i.e., assignments to logical tables in blocks of this
 memory type have at least granLb[mem] consecutive rows.
```

3.3.2 Member Function Documentation

3.3.2.1 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.blockOverlap (*self, mem, sl1, log, sl2*)

```
Returns true if table @log starting in block @sl1 would
overlap block @sl2
```

3.3.2.2 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.blocksInSameStageAs (*self, mem, sl*)

```
Returns list of blocks for the stage (st) that sl is part of.
```

3.3.2.3 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.blocksInStage (*self, mem, st*)

```
Returns indices of memory blocks of type @mem in stage @st
Indexing starts at 0 and covers all blocks of every memory
type in every stage.
```

3.3.2.4 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.dependencyConstraint (*self, model =* None)

```
If log2 match depends on log1, then last stage (TCAM/ SRAM)
of log1 is strictly before first stage (TCAM/ SRAM) of log2.
```

3.3.2.5 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.fillModel (*self, startRowDict, numberOfRowsDict, model*)

```
Given information about starting row and number of row
of each table in each block, fills in all the ILP
variables. Used to get a starting solution for ILP
from a greedy heuristic's output.
```

3.3.2.6 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.numberOfRowsBounds (*self, model =* None)

```
Enforces "minimum (physical) table size" constraint.
This helps us scale the ILP to handle FlexPipe
where legally a logical table can be assigned
any number of rows in each memory block (e.g., anywhere
from 1 to maxmimum rows availabl) but this leads to
a combinatorial explosion for ILP.
This constraint enforces that tables if assigned to a
block are assigned at least xx consecutive rows at a time.
```

3.3.2.7 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.setLb (*self*)

```
Just initializes lower bounds on physical table sizes
to default values if necessary
```

3.3.2.8 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.setup (*self, program, switch, preprocess*)

```
Set up variables, constraints and starting solution
(if specified) for CPLEX solver
```

3.3.2.9 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.setupConstraints (*self, model* = `None`)

```
Sets up all the different constraints.
```

3.3.2.10 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.solve (*self, program, switch, preprocess*)

```
Returns a configuration for program in switch, given some preprocessed information,
like packing unit sizes for different logical tables.
```

3.3.2.11 def mapper.src.flexpipe.flexpipe_ilp_compiler.FlexpipeIlpCompiler.wordLayoutConstraint (*self, model* = `None`)

```
Relate number of rows of a table per block to
number of blocks used and number of match entries (words)
assigned
```

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

- src/flexpipe/flexpipe_ilp_compiler.py

3.4 mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler Class Reference

Public Member Functions

- def **__init__** (self)
- def **getIndex** (self, table, names)
- def getBlocksInStage (self, mem, st)
- def getNextTable (self)
- def getMem (self, tableIndex)
- def makeGraph (self, program)
- def getMemTypesInStage (self)
- def setupSwitchToNext (self)
- def setupTablesInBlock (self)
- def setupDirty (self)
- def setupAssigned (self)
- def setupLastBlock (self)
- def setupRowsPerBlock (self)
- def setupNextTable (self)
- def getNextRange (self)
- def assignRowsToTable (self)
- def switchToNextStage (self)
- def **logCompilerAttempt** (self, compilerAttempt)
- def solve (self, program, switch, preprocess)

Public Attributes

- **logger**
- **gr**
- **memPerStage**
- **tablesInBlock**
- **dirty**
- **assigned**
- **lastBlockOfRow**
- **lastBlockOfTable**
- **startRowDict**
- **numberOfRowsDict**
- **orderedTables**
- **currentMem**
- **currentStage**
- **program**
- **switch**
- **preprocess**
- **results**
- **table**
- **tableIndex**
- **mems**
- **numWordsLeft**
- **rowRange**
- **numAssigned**
- **numTables**

3.4.1 Detailed Description

```
Greedy heuristic compiler for FlexPipe target,
```

3.4.2 Member Function Documentation

3.4.2.1 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.assignRowsToTable (*self*)

```
Assign self.slRange and self.rowRange (chunk of consecutive blocks/ rows)
to current table self.tableIndex. Here, we basically update the different
data structures that track which table is assigned to which row/ block
i.e., self.startRowDict[mem][t,sl], self.tablesInBlock[mem][sl][table]
self.lastBlockOfRow[mem][st][r], self.numberOfRowsDict[mem][t,sl]
self.dirty[mem][r,sl], self.lastBlockOfTable[mem][st][t], all of
these are initialized in the various setup... functions.
```

3.4.2.2 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.getBlocksInStage (*self, mem, st*)

```
Returns indices of memory blocks of type @mem in stage @st
Indexing starts at 0 and covers all blocks of every memory
type in every stage.
```

3.4.2.3 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.getMem (*self, tableIndex*)

```
Returns valid memory types for given table
```

3.4.2.4 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.getMemTypesInStage (*self*)

```
Get memory types in self.currentStage
```

3.4.2.5 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.getNextRange (*self*)

```
Returns the next chunk of free consecutive blocks/ rows we can use to assign
current self.table in self.currentStage.
```

3.4.2.6 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.getNextTable (*self*)

```
Get the next table from the heuristic order that hasn't been
assigned yet.
```

3.4.2.7 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.makeGraph (*self, program*)

```
Sets up dependency graph, only match edges force
a table to a new stage, so are weighted 1,
others are weighted 0.
Used to order tables in setupNextTable().
```

3.4.2.8 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.setupAssigned (*self*)

```
Maps table name to last stage it was assigned to.
```

3.4.2.9 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.setupDirty (*self*)

```
self.dirty[mem][row,sl] indicates if @row in the
@sl-th @mem block has been assigned to some table.
Filled in and used while assigning tables.
```

3.4.2.10 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.setupLastBlock (*self*)

```
sets up self.lastBlockOfRow[mem][st][row]
which is the last block in @mem, @st where
@row has been assigned to some table. Initial
value is -1.
also sets up self.lastBlockOf[mem][st][tableIndex]
which is the last block in @mem, @st that has been
assigned to @tableIndex.
Filled in and used while assigning tables.
```

3.4.2.11 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.setupNextTable (*self*)

```
Orders tables in order of "most constrained first"
(MCF) where a table is more constrained if it can go in
only a few @possible_stages based on
- its level (maxmimum distance from end of program in
the dependency chain)
- memory types it can go into
Ties are broken in favor of wider tables first.
```

3.4.2.12 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.setupRowsPerBlock (*self*)

```
Sets up the following containers
startRowDict[mem][log,sl] which is @row if the table
@log starts in @row of the @sl-th @mem memory block,
and zero otherwise.

numberOfRows[mem][log,sl] = num which is the number
of rows of table in the @sl-th @mem memory block
(when the table's match entries start in the block).
If there are no match entries in a block, or if
a match entry starts in a previous block but only
spills into this block @sl, then @num is 0.
```

3.4.2.13 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.setupSwitchToNext (*self*)

```
Order available (st, mem) tuples, so we can iterate
through self.memPerStage using switchToNext()
when assigning tables
```

3.4.2.14 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.setupTablesInBlock (*self*)

```
" Setup self.tablesInBlock[mem][sl] which
will contain names of all the tables assigned
to the sl-th block of type mem (index sl starts
at 0 and covers all blocks of type mem in
across all stages)
Filled in and used while assigning tables.
```

3.4.2.15 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.solve (*self, program, switch, preprocess*)

```
Runs the MCF greedy heuristic and returns switch configuration
```

3.4.2.16 def mapper.src.flexpipe.flexpipe_lpt_compiler.FlexpipeLptCompiler.switchToNextStage (*self*)

```
Update current stage/ memory type to next st, mem, where mem is a
valid memory type of the current table
```

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

- src/flexpipe/flexpipe_lpt_compiler.py

3.5 mapper.src.flexpipe.flexpipe_preprocess.FlexpipePreprocess Class Reference

Public Member Functions

- def **__init__** (self)
- def blocksInStage (self, mem, st)
- def setUseMemory (self)
- def **preprocess** (self, program, switch)

Public Attributes

- **logger**
- **use**
- **switch**
- **program**
- **pfBlocks**
- **toposortOrderStages**

3.5.1 Detailed Description

```
Preprocessor module that precomputes information such as valid packing units
for compiler to use.
```

3.5.2 Member Function Documentation

3.5.2.1 def mapper.src.flexpipe.flexpipe_preprocess.FlexpipePreprocess.blocksInStage (*self, mem, st*)

```
Returns indices of mem blocks in stage st
Indexing starts from first mem block in the first
stage and goes on till the last mem block over
all stages.
```

3.5.2.2 def mapper.src.flexpipe.flexpipe_preprocess.FlexpipePreprocess.setUseMemory (*self*)

```
Based on table's match type e.g., exact/ ternary etc., determine
which switch memory types it can use e.g., SRAM and TCAM/ TCAM etc.
```

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

- src/flexpipe/flexpipe_preprocess.py

3.6 mapper.src.flexpipe.flexpipe_switch.FlexpipeSwitch Class Reference

Public Member Functions

- def __init__

Public Attributes

- **logger**
- **memoryTypes**
- **numBlocks**
- **order**
- **numStages**
- **depth**
- **width**
- **maxTablesPerBlock**
- **matchType**

3.6.1 Constructor & Destructor Documentation

3.6.1.1 def mapper.src.flexpipe.flexpipe_switch.FlexpipeSwitch.__init__ (*self*, *numBlocks* = numBlocks, *depth* = depth, *width* = width, *order* = order, *maxTablesPerBlock* = maxTablesPerBlock)

```
initialize with resources available in each stage
The number of stages is implicit in the length of
 the numBlocks[mem] and order lists.
 The FlexPipe chip as described here has 4 stages.
 Note we could also have represented it as having
  5 stages with the third and fourth stage
  executing simultaneously, see note below.

Stages in FlexPipe are not uniform- different stages
have different memory types, so for each memory
type we specify the number of blocks available in
each stage in @numBlocks.

@depth and @width are mappings from memory types to the
depth and width of inidividual blocks where-
 depth is the number of match entries the block can store
 width is the max. width of a match entry that can fit

 @order is a list indexed by stage number where @order[i]
 is the topological order of stage i in the
 execution graph (where nodes correspond to stages
 and there is an edge from stage x to y if y executes
 after x). E.g., in a pipeline where stage 3 and 4 execute simultaneously
 order[3] = order[4] etc.

 @maxTablesPerBlock is the maximum number of logical tables
 that can share a memory block
```

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

- src/flexpipe/flexpipe_switch.py

3.7 mapper.src.rmt.rmt_configuration.RmtConfiguration Class Reference

Public Member Functions

- def **__init__** (self, program, switch, preprocess, layout, version)
- def getWordsPerTable (self)
- def checkWordsPerTable (self)
- def **configure** (self, layout)
- def getNumActiveSrams (self)
- def getNumActiveTcams (self)
- def **getPowerForRamsAndTcams** (self)
- def **getPipelineLatency** (self)
- def **getStartTimeOfStage** (self)
- def getColorsForTables (self)
- def **showDict** (self, d)
- def **showList** (sef, l)
- def showPic (self, filename, prefix)
- def getEarliestStageFromAssign (self, assignInfo)
- def display
- def getPerLogAssignInfo (self)
- def **displayFractionUsed** (self)
- def **displayInitialConditions** (self)
- def **showIntList**

- def **showInt** (self, num)
- def **showIntArray**
- def makeArrayToList (self, arr)

Public Attributes

- **logger**
- **program**
- **da**
- **switch**
- **preprocess**
- **version**
- **stMax**
- **logMax**
- **typeColor**
- **pfMax**
- **packingUnits**
- **wordsPerLog**
- **totalUnassignedWords**
- **layout**
- **blocks**
- **totalBlocks**
- **logColors**

3.7.1 Detailed Description

```
Module to describe and log an RMT switch configuration
for a given program
```

3.7.2 Member Function Documentation

3.7.2.1 def mapper.src.rmt.rmt_configuration.RmtConfiguration.checkWordsPerTable (*self*)

```
Check if enough match entries have been
assigned for each table
```

3.7.2.2 def mapper.src.rmt.rmt_configuration.RmtConfiguration.display (*self, paths =* [])

```
Display configuration, in many different ways- fraction of TCAMs/ SRAMs used, power, pipeline latency
start time of stages, tables in each stage, memories used by tables in each stage, dependencies
that force tables to start in a particular stage (useful for debugging program manually)
```

3.7.2.3 def mapper.src.rmt.rmt_configuration.RmtConfiguration.getColorsForTables (*self*)

```
Fills in self.logColors, a map from table index to color info (HEX, tuple)
```

3.7.2.4 def mapper.src.rmt.rmt_configuration.RmtConfiguration.getEarliestStageFromAssign (*self, assignInfo*)

```
Returns a mapping from tables
to information about the first stage they could have started
in- a table must start after all the tables it depends on
have been completely assigned. The informations is a list of
 lists [@st, @dep, @prev, @prevEnd] which should be read as
Table xx can't start before stage @st because of @dependency
on table @prev which ends in stage @prevEnd, one for each
@prev that the table depends on.
```

3.7.2.5 def mapper.src.rmt.rmt_configuration.RmtConfiguration.getNumActiveSrams (*self*)

```
For each table in each stage,
- number of RAMs for a match packing units (max if many types)
- + number of RAMs for an action packing unit
```

3.7.2.6 def mapper.src.rmt.rmt_configuration.RmtConfiguration.getNumActiveTcams (*self*)

```
If match data width is less than TCAM width, only
a fraction of TCAM is active/ consumes power.
```

3.7.2.7 def mapper.src.rmt.rmt_configuration.RmtConfiguration.getPerLogAssignInfo (*self*)

```
 Return mapping from table to assignment info
  'start': start stage (or -1 if not valid)
  'end': end stage (or -1 if not valid)
```

3.7.2.8 def mapper.src.rmt.rmt_configuration.RmtConfiguration.getWordsPerTable (*self*)

```
Get match entries per table per stage.
```

3.7.2.9 def mapper.src.rmt.rmt_configuration.RmtConfiguration.makeArrayToList (*self, arr*)

```
Makes an encapsulated array [[2], [5], [4]] into a list [2, 5, 4].
```

3.7.2.10 def mapper.src.rmt.rmt_configuration.RmtConfiguration.showPic (*self, filename, prefix*)

```
Output a picture of the configuration to a file called @prefix@filename
```

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

- src/rmt/rmt_configuration.py

3.8 mapper.src.rmt.rmt_dependency_analysis.RmtDependencyAnalysis Class Reference

Public Member Functions

- def **__init__** (self, program)
- def **getDigraph** (self)
- def **addWeights** (self, originalGr, tableWeights)

- def **reverseEdges** (self, originalGr)
- def **makePath** (self, nodes, gr)
- def **showPath**
- def **getPerLogProgramInfo**
- def **flipEdgeSign** (self, originalGr)
- def showCriticalPath

Public Attributes

- **logger**
- **program**
- **logMax**

3.8.1 Member Function Documentation

3.8.1.1 def mapper.src.rmt.rmt_dependency_analysis.RmtDependencyAnalysis.showCriticalPath (*self, gr* = None)

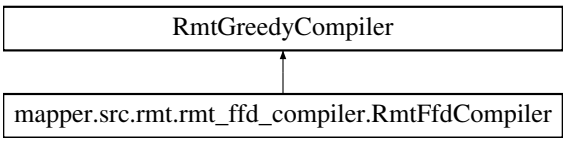
```
show critical path- i.e., longest path of dependencies, where weight of
dependency edge is 1 if it forces a table into a new stage.
```

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

- src/rmt/rmt_dependency_analysis.py

3.9 mapper.src.rmt.rmt_ffd_compiler.RmtFfdCompiler Class Reference

Inheritance diagram for mapper.src.rmt.rmt_ffd_compiler.RmtFfdCompiler:



Public Member Functions

- def **__init__**
- def **getLimitingResourceUse** (self, table)
- def getOrderedTables (self)

Public Attributes

- **logger**
- **orderedTables**

3.9.1 Member Function Documentation

3.9.1.1 def mapper.src.rmt.rmt_ffd_compiler.RmtFfdCompiler.getOrderedTables (*self*)

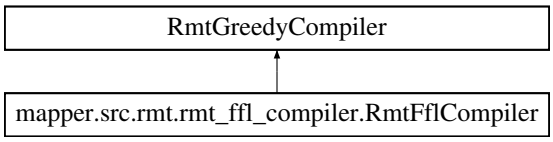
```
Return tables in order of limiting resource- if a table needs 2 out 8 available input xbar units
and 512 of say 1024 available bits of the action data xbar (in every stage), then action data xbar
is the limiting resources, since it need 50% of the available resource per stage
(vs 25% for input xbar).
```

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

- src/rmt/rmt_ffd_compiler.py

3.10 mapper.src.rmt.rmt_ffl_compiler.RmtFflCompiler Class Reference

Inheritance diagram for mapper.src.rmt.rmt_ffl_compiler.RmtFflCompiler:



Public Member Functions

- def **__init__**
- def getOrderedTables (self)

Public Attributes

- **logger**
- **orderedTables**

3.10.1 Member Function Documentation

3.10.1.1 def mapper.src.rmt.rmt_ffl_compiler.RmtFflCompiler.getOrderedTables (*self*)

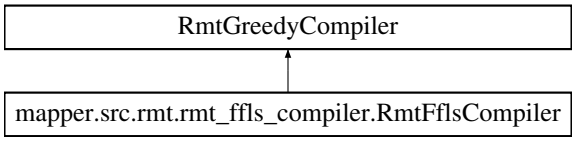
```
Return tables in order of level- number of edges in the longest dependency
path to the end of the program, where an edge/ dependency has weight 1 if it forces
the second table to a new stage
```

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

- src/rmt/rmt_ffl_compiler.py

3.11 mapper.src.rmt.rmt_ffls_compiler.RmtFflsCompiler Class Reference

Inheritance diagram for mapper.src.rmt.rmt_ffls_compiler.RmtFflsCompiler:



Public Member Functions

- def **__init__**
- def **getNumStages** (self, tableIndex)
- def **getOrderedTables** (self)

Public Attributes

- **logger**
- **orderedTables**

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

- src/rmt/rmt_ffls_compiler.py

3.12 mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler Class Reference

Public Member Functions

- def **__init__**
- def getIndex (self, table, names)
- def getBestFitForWords (self, tableIndex, mem, lastBlock, numWordsLeft)
- def getPerPf (self, mem, tableIndex)
- def getBestFitForBlocks (self, tableIndex, mem, lastBlock, numSramBlocksReserved)
- def getNextTable (self)
- def **getMem** (self, tableIndex)
- def **getInputCrossbarSubunits** (self, tableIndex, mem)
- def **getMaximumInputCrossbarSubunits** (self, tableIndex, mem)
- def **getAvailableInputCrossbarSubunits** (self, tableIndex, mem)
- def **getWidthActionData** (self, tableIndex)
- def **getMaximumWidthActionData** (self, tableIndex, mem)
- def **getAvailableWidthActionData** (self, tableIndex, mem)
- def **updateLastBlockAndAssign**
- def getNumSramBlocksReserved (self)
- def **logCompilerAttempt** (self, compilerAttempt)
- def solve (self, program, switch, preprocess)

Public Attributes

- **numSramBlocksReserved**
- **version**
- **logger**
- **program**
- **switch**
- **preprocess**
- **results**
- **gr**
- **blocksPerSt**
- **numBlocks**
- **blocksPerPf**
- **wordsPerPf**
- **layout**
- **block**
- **word**
- **inputCrossbar**
- **actionCrossbar**
- **lastBlock**
- **assigned**
- **currentStage**
- **table**
- **numWordsLeft**

3.12.1 Detailed Description

```
Greedy heuristic compiler, the specific greedy heuristic
is determined by the getOrderedTables function defined in
child classes RmtFFlCompiler and RmtFfdCompiler etc.
```

3.12.2 Member Function Documentation

3.12.2.1 def mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler.getBestFitForBlocks (*self, tableIndex, mem, lastBlock, numSramBlocksReserved*)

```
Given number of blocks available, find a config. (i.e., pick a
packing unit) to fit the maximum number of entries possible
Based solely on memory resources needed for match and action data,
not other resources like crossbar units etc.
```

3.12.2.2 def mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler.getBestFitForWords (*self, tableIndex, mem, lastBlock, numWordsLeft*)

```
Given the number of match entries left, find minimum number of
blocks needed to fit all of them
Based solely on memory resources needed for match and action data,
not other resources like crossbar units etc.
```

3.12.2.3 def mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler.getIndex (*self, table, names*)

```
Returns index of table in names
```

3.12.2.4 def mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler.getNextTable (*self*)

```
Get the next table from the heuristic order that hasn't been
assigned yet.
```

3.12.2.5 def mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler.getNumSramBlocksReserved (*self*)

```
Returns number SRAM blocks reserved for the action data of tables
that will go in TCAM
```

3.12.2.6 def mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler.getPerPf (*self, mem, tableIndex*)

```
Get packing unit configurations for given table in given memory type.
```

3.12.2.7 def mapper.src.rmt.rmt_greedy_compiler.RmtGreedyCompiler.solve (*self, program, switch, preprocess*)

```
Runs a greedy heuristic and returns switch configuration
```

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

- src/rmt/rmt_greedy_compiler.py

3.13 mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler Class Reference

Public Member Functions

- def __init__
- def **setDimensionSizes** (self)
- def **computeSum** (self, dictCount)
- def **P** (self, l, m)
- def actionAssignmentConstraint (self)
- def capacityConstraint (self)
- def wordLayoutConstraint (self)
- def useMemoryConstraint (self)
- def assignmentConstraint (self)
- def pipelineLatencyVariables (self)
- def getStartAllMemTimesStartTimeOfStage (self)
- def getEndAllMemTimesStartTimeOfStage (self)
- def **checkPipelineLatencyConstraint** (self, model)
- def **pipelineLatencyConstraint** (self)
- def **checkStartingAndEndingStagesConstraint** (self, model)
- def displayActiveRams (self, model)
- def **displayStartingAndEndingStages** (self, model)
- def **getStartingAndEndingStages** (self)
- def dependencyConstraint (self)
- def **maximumStageConstraint** (self)
- def **getBlockBinary** (self)
- def **getLayoutBinary** (self)
- def **checkInputCrossbarConstraint** (self, model, mem)
- def **inputCrossbarConstraint** (self)
- def **getBlockAllMemBinary** (self)
- def resolutionLogicConstraint (self)
- def actionCrossbarConstraint (self)
- def onePackingUnitForLogInStage (self)
- def getNumActiveSrams (self)
- def getNumActiveTcams (self)
- def **getPowerForRamsAndTcamsObjective** (self)
- def **startAndEndStagesVariables** (self)
- def **getXxAllMemTimesBlockAllMemBin** (self)
- def **getIlpStartingDictValues** (self, block, layout, word, startTimeOfStage)
- def **displayMaximumStage** (self, model)
- def solve (self, program, switch, preprocess)
- def **setIlpResults** (self, solverTimes, nIterations)
- def checkConstraints (self, model)

Public Attributes

- **logger**
- objectiveStr

  self.logger.debug("Displaying greedy solution") greedyConfig.display()
- **relativeGap**
- **greedyVersion**
- **emphasis**
- **variableSelect**
- **timeLimit**
- **treeLimit**

- **workMem**
- **nodeFileInd**
- **workDir**
- **ignoreConstraint**
- **dictNumVariables**
- **numVariables**
- **dictNumConstraints**
- **numConstraints**
- **dimensionSizes**
- **startTimeOfStage**
- **startAllMemTimesStartTimeOfStage**
- **endAllMemTimesStartTimeOfStage**
- **startTimeOfStartStageOfLog**
- **startTimeOfEndStageOfLog**
- **numActiveSrams**
- **numActiveTcams**
- **powerForRamsAndTcams**
- **startAllMem**
- **startAllMemTimesBlockAllMemBin**
- **endAllMem**
- **endAllMemTimesBlockAllMemBin**
- **program**
- **switch**
- **preprocess**
- **all**
- pfMax

  Constants.
- **stMax**
- **logMax**
- **blockMax**
- **wordMax**
- results

  Variables.
- **m**
- **word**
- **block**
- **blockBin**
- **layout**
- **layoutBin**
- **blockAllMemBin**
- **isMaximumStage**
- **totalBlocksForStBin**
- **isMaximumStageTimesTotalBlocksForStBin**
- **startingDict**

3.13.1 Constructor & Destructor Documentation

3.13.1.1 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.__init__ (*self*, *relativeGap*, *greedyVersion*, *objectiveStr* = `'maximumStage'`, *emphasis* = `0`, *timeLimit* = `None`, *treeLimit* = `None`, *variableSelect* = `None`, *ignoreConstraint* = `None`, *workMem* = `None`, *nodeFileInd* = `None`, *workDir* = `None`)

```
Initialize compiler with CPLEX parameters
@relativeGap is a fraction f so that CPLEX will stop at a solution that
 is within f of the optimal, corresponds to CPX_PARAM_EPGAP in CPLEX
 @greedyVersion is one of 'ffl'/ 'ffd' etc.
 @objectiveStr is one of pipelineLatency, maximumStage, powerForRamsAndTcams
 other parameters are optional .. (see pycpx for more info)
```

3.13.2 Member Function Documentation

3.13.2.1 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.actionAssignmentConstraint (*self*)

```
How many action words we can fit.
```

3.13.2.2 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.actionCrossbarConstraint (*self*)

```
No more than 1280 bits of action from each stage.
```

3.13.2.3 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.assignmentConstraint (*self*)

```
All match entries must be assigned somewhere in the pipeline.
```

3.13.2.4 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.capacityConstraint (*self*)

```
Don't use more memories than available in each stage for action, match
```

3.13.2.5 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.checkConstraints (*self, model*)

```
Given a complete model with all variables filled in,
checks against (some of) the inequality constraints that define the ILP
and warns if anything's violated. Used to check greedy solutions,
and other starting solutions.
```

3.13.2.6 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.dependencyConstraint (*self*)

```
If log2 action depends on log1, then last stage (any mem)
of log1 is strictly before first stage (any mem) of log2.
```

3.13.2.7 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.displayActiveRams (*self, model*)

```
For each table in each stage,
- number of RAMs for a match packing units (enforce one type per st)
- + number of RAMs for an action packing unit
```

3.13.2.8 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.getEndAllMemTimesStartTimeOfStage (*self*)

```
Defining product variable endAllMem x StartTimeOfStage
```

3.13.2.9 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.getNumActiveSrams (*self*)

```
For each table in each stage,
- number of RAMs for a match packing units (enforce one type per st)
- + number of RAMs for an action packing unit
```

3.13.2.10 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.getNumActiveTcams (*self*)

```
If match data width is less than TCAM width, only
a fraction of TCAM is active/ consumes power.
```

3.13.2.11 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.getStartAllMemTimesStartTimeOfStage (*self*)

```
Defining product variable StartAllMem x StartTimeOfStage
```

3.13.2.12 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.onePackingUnitForLogInStage (*self*)

```
Restrict logical tables to use same packing unit in a stage
instead of combinations of different size packing units in the
same stage. Packing units in different stages can be different though.
```

3.13.2.13 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.pipelineLatencyVariables (*self*)

```
Variables for start and end time of stages
```

3.13.2.14 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.resolutionLogicConstraint (*self*)

```
Limits number of match tables per stage. Since table match resolution logic
can only handle a finite number
```

3.13.2.15 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.solve (*self, program, switch, preprocess*)

```
Returns a configuration for program in switch, given some preprocessed information,
like possible packing units for different logical tables.
```

3.13.2.16 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.useMemoryConstraint (*self*)

```
assign blocks of logical table to mem only if it's allowed
e.g., a ternary table can't use SRAM blocks. Preprocessor
computes what's allowed (self.preprocess.use)
```

3.13.2.17 def mapper.src.rmt.orig_rmt_compiler.RmtIlpCompiler.wordLayoutConstraint (*self*)

```
Relating number of packing units to number of blocks used/
words per row fit
```

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

- src/rmt/orig_rmt_compiler.py

3.14 mapper.src.rmt.rmt_preprocess.RmtPreprocess Class Reference

Public Member Functions

- def **__init__** (self)
- def **gcd** (self, a, b)
- def **lcm** (self, a, b)
- def getMaxPf (self, logicalWidth, blockWidth)
- def setUseMemory (self)
- def **preprocess** (self, program, switch)
- def GetMultipleBefore (self, maxValue, div)

Public Attributes

- **logger**
- **use**
- **program**
- **switch**
- **toposortOrderStages**
- **layout**
- **word**
- **NumPackingFactors**
- **inputCrossbarNumSubunits**
- **actionCrossbarNumBits**
- **actionWidths**

3.14.1 Detailed Description

```
Preprocessor module that precomputes information such as candidate packing units
for compiler to use.
```

3.14.2 Member Function Documentation

3.14.2.1 def mapper.src.rmt.rmt_preprocess.RmtPreprocess.getMaxPf (*self, logicalWidth, blockWidth*)

```
minimum number of memory blocks needed for match entries to
fit evenly in each row without wasting any bits. Any more blocks
and we can view it as a combination of smaller packing units
```

3.14.2.2 def mapper.src.rmt.rmt_preprocess.RmtPreprocess.GetMultipleBefore (*self, maxValue, div*)

```
Get multiple of div just before maxValue
```

3.14.2.3 def mapper.src.rmt.rmt_preprocess.RmtPreprocess.setUseMemory (*self*)

```
Based on table's match type e.g., exact/ ternary etc., determine
which switch memory types it can use e.g., SRAM and TCAM/ TCAM etc.
```

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

- src/rmt/rmt_preprocess.py

3.15 mapper.src.rmt.rmt_switch.RmtSwitch Class Reference

Public Member Functions

- def **__init__**

Public Attributes

- **logger**
- **memoryTypes**
- **matchType**
- **numBlocks**
- **toposortOrderStages**
- **unpackableMemTypes**
- **numStages**
- **depth**
- **width**
- **inputCrossbarNumSubunits**
- **inputCrossbarWidthSubunit**
- **actionCrossbarNumBits**
- **resolutionLogicNumMatchTables**
- **matchDelay**
- **actionDelay**
- **successorDelay**
- **power**
- **typesIn**
- **inMem**
- **allTypes**

3.15.1 Detailed Description

```
RMT switch. Describes resources to be configured in the RMT switch.
```

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

- src/rmt/rmt_switch.py

Index