AIX Frontend Compiler Reference Manual

ver. 1.0

Written by Youngsun Han (youngsun@pknu.ac.kr)

High Performance Computing Lab.
Department of Computer Engineering
Pukyong National University

Contents

Chapter 1

SKT AIX Frontend Compiler

This README describes the organization and usage of the SKT AIX Frontend Compiler.

1. Source Organization

Common

- AxfcFrontendCompiler
- AxfcIRBuilder
- AxfcIRTranslator
- AxfcMachineDesc
- AxfcGraphWriter
- AxfcIRGraph
- AxfcIRBlock
- AxfcIRNode
- AxfcError
- AxfcMain

Tensorflow

- AxfcTFIRBuilder
- AxfcTFIRTranslator

SKT-AIX

- aixh_pb2

2. Usage

```
$$ python3 AxfcMain.py [-m] [-i] [-o] [-l] [-g]

-m: Path to a machine description file
-i: Path to the protocol buffer of a frozen model
-o: Path to output the generated AIXGraph
-l: Path to log out (optional)
-g: Path to dump out an IR graph (optional)
```

3. Contact

- Youngsun Han (youngsun@pknu.ac.kr)
- Associate Professor
- Department of Computer Engineering, Pukyong National University

Chapter 2

aix_tf

{ "AIX_MODEL_TYPE": "TENSORFLOW", "AIX_PROFIT_THRESHOLD": 500, "AIX_LAYER": { "Conv2D": { "layer": "AIX_LAYER_CONVOLUTION", "activation": "AIX_ACTIVATION_IDENTITY", "is_group": false, "is_conv": true, "profit": 100 }, "DepthwiseConv2dNative": { "layer": "AIX_LAYER_GROUP_CONV", "activation": "AIX_ACTIVATION_IDENTITY", "is_group": true, "is_conv": true, "profit": 100 }, "FusedBatchNorm": { "layer": "AIX_LAYER_BATCHNORM", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "BatchNorm": { "layer": "AIX_LAYER_BATCHNORM", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "AvgPool": { "layer": "AIX_LAYER_AVGPOOL", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "MaxPool": { "layer": "AIX_LAYER_MAXPOOL", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "Softmax": { "layer": "AIX_LAYER_SOFTMAX", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "Add": { "layer": "AIX_LAYER_EWADD", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "Relu": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_RELU", "is_group": false, "is_conv": false, "profit": 100 }, "Relu6": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_LEAKY_RELU", "is_group": false, "is_conv": false, "profit": 100 }, "BiasAdd": { "layer": "AIX_LAYER_BIASADD", "activation": null, "is_group": false, "is_conv": false, "profit": 100 }, "Sigmoid": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_SIGMOID", "is_group": false, "is_conv": false, "profit": 100 }, "Prelu": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_PRELU", "is_group": false, "is_conv": false, "profit": 100 }, "Tanh": { "layer": "AIX_LAYER_ACTIVATION", "activation": "AIX_ACTIVATION_TANH", "is_group": false, "is_conv": false, "profit": 100 } } }

Chapter 3

Namespace Index

3.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

Chapter 7

Namespace Documentation

7.1 aixh_pb2 Namespace Reference

Variables

- int _b = sys.version_info[0] < 3 and (lambda x: x) or (lambda x: x.encode('latin1'))
- _sym_db = _symbol_database.Default()
- DESCRIPTOR
- _AIXLAYER_AIXLAYERTYPE
- _AIXLAYER_AIXACTIVATIONMODE
- _AIXLAYER_AIXSAMPLINGMODE
- _AIXLAYER_AIXDATATYPE
- _AIXLAYER_AIXTENSORFORMAT
- _AIXLAYER_AIXCONVOLUTIONDESC
- _AIXLAYER_AIXSAMPLINGDESC
- _AIXLAYER_AIXEWADDDESC
- _AIXLAYER_AIXTENSOR
- _AIXLAYER
- _AIXGRAPH
- enum_type
- containing_type
- message_type
- AIXLayer
- AIXGraph
- has_options
- _options

7.1.1 Variable Documentation

7.1.1.1 _AIXGRAPH

```
aixh_pb2._AIXGRAPH  [private]
```

Definition at line 678 of file aixh_pb2.py.

7.1.1.2 _AIXLAYER

aixh_pb2._AIXLAYER [private]

Definition at line 509 of file aixh_pb2.py.

7.1.1.3 _AIXLAYER_AIXACTIVATIONMODE

aixh_pb2._AIXLAYER_AIXACTIVATIONMODE [private]

Definition at line 158 of file aixh_pb2.py.

7.1.1.4 _AIXLAYER_AIXCONVOLUTIONDESC

aixh_pb2._AIXLAYER_AIXCONVOLUTIONDESC [private]

Definition at line 298 of file aixh_pb2.py.

7.1.1.5 _AIXLAYER_AIXDATATYPE

aixh_pb2._AIXLAYER_AIXDATATYPE [private]

Definition at line 230 of file aixh_pb2.py.

7.1.1.6 _AIXLAYER_AIXEWADDDESC

aixh_pb2._AIXLAYER_AIXEWADDDESC [private]

Initial value:

```
=  _descriptor.Descriptor(
    name='AIXEWAddDesc',
    full_name='aixh.AIXLayer.AIXEWAddDesc',
    filename=None,
    file=DESCRIPTOR,
    containing_type=None,
    fields=[
        _descriptor.FieldDescriptor(
            name='scale', full_name='aixh.AIXLayer.AIXEWAddDesc.scale', index=0,
             number=1, type=2, cpp_type=6, label=3,
             has_default_value=False, default_value=[],
             message_type=None, enum_type=None, containing_type=None,
             is_extension=False, extension_scope=None,
             options=None),
     ],
     extensions=[
     ],
     nested_types=[],
     enum_types=[
     ],
     options=None,
     is_extendable=False,
     syntax='proto2',
     extension_ranges=[],
     oneofs=[
     ],
     serialized_start=972,
     serialized_end=1001,
 )
```

Definition at line 407 of file aixh_pb2.py.

7.1.1.7 _AIXLAYER_AIXLAYERTYPE

```
aixh_pb2._AIXLAYER_AIXLAYERTYPE  [private]
```

Definition at line 72 of file aixh_pb2.py.

7.1.1.8 _AIXLAYER_AIXSAMPLINGDESC

```
aixh_pb2._AIXLAYER_AIXSAMPLINGDESC  [private]
```

Definition at line 356 of file aixh_pb2.py.

7.1.1.9 _AIXLAYER_AIXSAMPLINGMODE

```
aixh_pb2._AIXLAYER_AIXSAMPLINGMODE  [private]
```

Definition at line 196 of file aixh_pb2.py.

7.1.1.10 _AIXLAYER_AIXTENSOR

```
aixh_pb2._AIXLAYER_AIXTENSOR  [private]
```

Definition at line 437 of file aixh_pb2.py.

7.1.1.11 _AIXLAYER_AIXTENSORFORMAT

```
aixh_pb2._AIXLAYER_AIXTENSORFORMAT  [private]
```

Initial value:

```
=  _descriptor.EnumDescriptor(
    name=’AIXTensorFormat’,
    full_name=’aixh.AIXLayer.AIXTensorFormat’,
    filename=None,
    file=DESCRIPTOR,
    values=[
        _descriptor.EnumValueDescriptor(
            name=’AIX_FORMAT_NCHW’, index=0, number=0,
            options=None,
             type=None),
         _descriptor.EnumValueDescriptor(
             name=’AIX_FORMAT_NHWC’, index=1, number=1,
             options=None,
             type=None),
         _descriptor.EnumValueDescriptor(
             name=’AIX_FORMAT_NWHC’, index=2, number=2,
             options=None,
             type=None),
         _descriptor.EnumValueDescriptor(
             name=’AIX_FORMAT_VECTOR’, index=3, number=3,
             options=None,
             type=None),
     ],
     containing_type=None,
     options=None,
     serialized_start=2094,
     serialized_end=2197,
 )
```

Definition at line 268 of file aixh_pb2.py.

7.1.1.12 _b

```
int aixh_pb2._b = sys.version_info[0] < 3 and (lambda x:  x) or (lambda x:  x.encode('latin1'))
[private]
```

Definition at line 6 of file aixh_pb2.py.

7.1.1.13 _options

```
aixh_pb2._options  [private]
```

Definition at line 798 of file aixh_pb2.py.

7.1.1.14 _sym_db

```
aixh_pb2._sym_db = _symbol_database.Default()  [private]
```

Definition at line 15 of file aixh_pb2.py.

7.1.1.15 AIXGraph

```
aixh_pb2.AIXGraph
```

Initial value:

```
=  _reflection.GeneratedProtocolMessageType('AIXGraph', (_message.Message,), dict(
    DESCRIPTOR=_AIXGRAPH,
    __module__='aixh_pb2'
    # @@protoc_insertion_point(class_scope:aixh.AIXGraph)
))
```

Definition at line 790 of file aixh_pb2.py.

7.1.1.16 AIXLayer

```
aixh_pb2.AIXLayer
```

Definition at line 751 of file aixh_pb2.py.

7.1.1.17 containing_type

```
aixh_pb2.containing_type
```

Definition at line 723 of file aixh_pb2.py.

7.1.1.18 DESCRIPTOR

```
aixh_pb2.DESCRIPTOR
```

Definition at line 17 of file aixh_pb2.py.

7.1.1.19 enum_type

```
aixh_pb2.enum_type
```

Definition at line 722 of file aixh_pb2.py.

7.1.1.20 has_options

```
aixh_pb2.has_options
```

Definition at line 797 of file aixh_pb2.py.

7.1.1.21 message_type

```
aixh_pb2.message_type
```

Definition at line 731 of file aixh_pb2.py.

7.2 AxfcError Namespace Reference

Classes

- class AxfcError

  AxfcError enum class.

7.3 AxfcFrontendCompiler Namespace Reference

Classes

- class AxfcFrontendCompiler

  AxfcFrontendCompiler.

7.4 AxfcGraphWriter Namespace Reference

Classes

- class AxfcGraphWriter

  AxfcGraphWriter class.

7.5 AxfcIRBlock Namespace Reference

Classes

- class AxfcIRBlock

  AxfcIRBlock class.

7.6 AxfcIRBuilder Namespace Reference

Classes

- class AxfcIRBuilder

  AxfcIRBuilder class.

7.7 AxfcIRGraph Namespace Reference

Classes

- class AxfcIRGraph

  AxfcIRGraph class.

7.8 AxfcIRNode Namespace Reference

Classes

- class AxfcIRNode

  AxfcIRNode.

7.9 AxfcIRTranslator Namespace Reference

Classes

- class AIXTensorType

  AIXInputType enum class.
- class AxfcIRTranslator

7.10 AxfcLauncherWriter Namespace Reference

Classes

- class AxfcLauncherWriter

  AxfcLauncherWriter class.

7.11 AxfcMachineDesc Namespace Reference

Classes

- class AxfcMachineDesc

  AxfcMachineDesc class.

7.12 AxfcMain Namespace Reference

Functions

- def __main (vargs)

  main function

Variables

- parser
- metavar
- type
- str
- required
- help
- args = parser.parse_args()

7.12.1 Function Documentation

7.12.1.1 __main()

```
def AxfcMain.__main (
              vargs )  [private]
```

main function

Definition at line 20 of file AxfcMain.py.

7.12.2 Variable Documentation

7.12.2.1 args

```
AxfcMain.args = parser.parse_args()
```

Definition at line 100 of file AxfcMain.py.

7.12.2.2 help

```
AxfcMain.help
```

Definition at line 90 of file AxfcMain.py.

7.12.2.3 metavar

```
AxfcMain.metavar
```

Definition at line 89 of file AxfcMain.py.

7.12.2.4 parser

```
AxfcMain.parser
```

Initial value:

```
=  argparse.ArgumentParser(
        description=’SKT AIX Frontend Compiler’,
        usage=’use "%(prog)s -h/--help" for more information’)
```

Definition at line 85 of file AxfcMain.py.

7.12.2.5 required

```
AxfcMain.required
```

Definition at line 89 of file AxfcMain.py.

7.12.2.6 str

```
AxfcMain.str
```

Definition at line 89 of file AxfcMain.py.

7.12.2.7 type

```
AxfcMain.type
```

Definition at line 89 of file AxfcMain.py.

7.13 AxfcTFIRBuilder Namespace Reference

Classes

- class AxfcTFIRBuilder

  AxfcTFIRBuilder class.

7.14 AxfcTFIRTranslator Namespace Reference

Classes

- class AxfcTFIRTranslator

  AxfcTFIRTranslator class.

Variables

- dictionary aix_data_type_tbl

  Global tables for AIXDataType and AIXTensorFormat.
- dictionary aix_tensor_format_tbl

  AIXTensorFormat table.

7.14.1 Variable Documentation

7.14.1.1 aix_data_type_tbl

`dictionary AxfcTFIRTranslator.aix_data_type_tbl`

Initial value:

```
=  {
    tf.float16: AIXLayer.AIXDataType.AIX_DATA_HALF,
    tf.float32: AIXLayer.AIXDataType.AIX_DATA_FLOAT,
    tf.float64: AIXLayer.AIXDataType.AIX_DATA_DOUBLE,
    tf.uint8: AIXLayer.AIXDataType.AIX_DATA_UINT8,
    tf.int8: AIXLayer.AIXDataType.AIX_DATA_SINT8,
    tf.int16: AIXLayer.AIXDataType.AIX_DATA_SINT16
}
```

Global tables for AIXDataType and AIXTensorFormat.

AIXDataType table

Definition at line 23 of file AxfcTFIRTranslator.py.

7.14.1.2 aix_tensor_format_tbl

`dictionary AxfcTFIRTranslator.aix_tensor_format_tbl`

Initial value:

```
=  {
    b"NCHW": AIXLayer.AIXTensorFormat.AIX_FORMAT_NCHW,
    b"NHWC": AIXLayer.AIXTensorFormat.AIX_FORMAT_NHWC,
    b"NWHC": AIXLayer.AIXTensorFormat.AIX_FORMAT_NWHC,
    b"VECTOR": AIXLayer.AIXTensorFormat.AIX_FORMAT_VECTOR
}
```

AIXTensorFormat table.

Definition at line 33 of file AxfcTFIRTranslator.py.

Chapter 8

Class Documentation

8.1 AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo Class Reference

AIXLayerInfo inner class.

Public Member Functions

- def __init__ (self, op)

  The constructor.
- def __str__ (self)

  For debugging.

Public Attributes

- op

  layer operation name of the layer info
- layer

  AIX layer ID of the layer info.
- activation

  AIX activation ID of the layer info.
- is_group

  indicate whether this layer is group layer or not
- is_conv

  indicate whether this layer is convolution layer or not
- profit

  the profit that can be obtained by accelerating this layer using AIXH

8.1.1 Detailed Description

AIXLayerInfo inner class.

Definition at line 148 of file AxfcMachineDesc.py.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 __init__()

```
def AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.__init__ (
              self,
              op )
```

The constructor.

Definition at line 168 of file AxfcMachineDesc.py.

8.1.3 Member Function Documentation

8.1.3.1 __str__()

```
def AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.__str__ (
              self )
```

For debugging.

Definition at line 177 of file AxfcMachineDesc.py.

8.1.4 Member Data Documentation

8.1.4.1 activation

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.activation
```

AIX activation ID of the layer info.

Definition at line 171 of file AxfcMachineDesc.py.

8.1.4.2 is_conv

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.is_conv
```

indicate whether this layer is convolution layer or not

Definition at line 173 of file AxfcMachineDesc.py.

8.1.4.3 is_group

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.is_group
```

indicate whether this layer is group layer or not

Definition at line 172 of file AxfcMachineDesc.py.

8.1.4.4 layer

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.layer
```

AIX layer ID of the layer info.

Definition at line 170 of file AxfcMachineDesc.py.

8.1.4.5 op

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.op
```

layer operation name of the layer info

Definition at line 169 of file AxfcMachineDesc.py.

8.1.4.6 profit

```
AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo.profit
```

the profit that can be obtained by accelerating this layer using AIXH

Definition at line 174 of file AxfcMachineDesc.py.

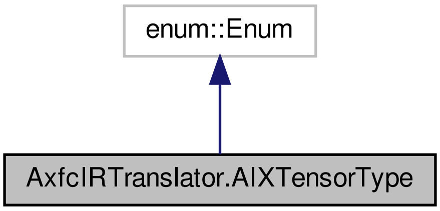
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMachineDesc.py

8.2 AxfcIRTranslator.AIXTensorType Class Reference

AIXInputType enum class.

Inheritance diagram for AxfcIRTranslator.AIXTensorType:



Static Public Attributes

- int AIX_TENSOR_INPUT = 0
- int AIX_TENSOR_FILTER = 1
- int AIX_TENSOR_BIAS = 2
- int AIX_TENSOR_SCALE = 3
- int AIX_TENSOR_MEAN = 4
- int AIX_TENSOR_VARIANCE = 5
- int AIX_TENSOR_OUTPUT = 6
- int AIX_TENSOR_UNKNOWN = 7

8.2.1 Detailed Description

AIXInputType enum class.

Definition at line 22 of file AxfcIRTranslator.py.

8.2.2 Member Data Documentation

8.2.2.1 AIX_TENSOR_BIAS

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_BIAS = 2  [static]
```

Definition at line 25 of file AxfcIRTranslator.py.

8.2.2.2 AIX_TENSOR_FILTER

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_FILTER = 1  [static]
```

Definition at line 24 of file AxfcIRTranslator.py.

8.2.2.3 AIX_TENSOR_INPUT

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_INPUT = 0  [static]
```

Definition at line 23 of file AxfcIRTranslator.py.

8.2.2.4 AIX_TENSOR_MEAN

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_MEAN = 4  [static]
```

Definition at line 27 of file AxfcIRTranslator.py.

8.2.2.5 AIX_TENSOR_OUTPUT

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_OUTPUT = 6  [static]
```

Definition at line 29 of file AxfcIRTranslator.py.

8.2.2.6 AIX_TENSOR_SCALE

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_SCALE = 3  [static]
```

Definition at line 26 of file AxfcIRTranslator.py.

8.2.2.7 AIX_TENSOR_UNKNOWN

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_UNKNOWN = 7  [static]
```

Definition at line 30 of file AxfcIRTranslator.py.

8.2.2.8 AIX_TENSOR_VARIANCE

```
int AxfcIRTranslator.AIXTensorType.AIX_TENSOR_VARIANCE = 5  [static]
```

Definition at line 28 of file AxfcIRTranslator.py.

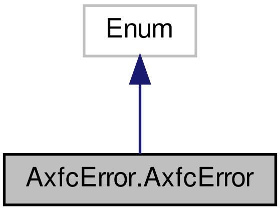
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRTranslator.py

8.3 AxfcError.AxfcError Class Reference

AxfcError enum class.

Inheritance diagram for AxfcError.AxfcError:



Static Public Attributes

- int SUCCESS = 0
- int INVALID_PARAMETER = 1
- int INVALID_FILE_PATH = 2
- int INVALID_INPUT_TYPE = 3
- int INVALID_TF_GRAPH = 4
- int INVALID_IR_GRAPH = 5
- int EMPTY_IR_BLOCK = 6
- int PRED_NODE_NOT_FOUND = 7
- int INVALID_MD_FORMAT = 8
- int NOT_AIXH_SUPPORT = 9
- int NOT_IMPLEMENTED = 10
- int UNKNOWN_TENSOR_TYPE = 11
- int UNSUPPORTED_AIX_LAYER_EMIT = 12
- int INVALID_AIX_LAYER_TYPE = 13
- int INVALID_AIX_TENSOR_FORMAT = 14
- int INVALID_CONVOLUTION_LAYER = 15
- int INVALID_GROUP_CONV_LAYER = 16
- int INVALID_BATCHNORM_LAYER = 17
- int INVALID_ACTIVATION_LAYER = 18

- int INVALID_IDENTITY_LAYER = 19
- int INVALID_PAD_LAYER = 20
- int INVALID_AIX_TENSOR_INPUT = 21
- int DUMP_IR_GRAPH_ERROR = 22
- int INVALID_AIX_GRAPH = 23
- int INVALID_MAXPOOL_LAYER = 24
- int INVALID_EWADD_LAYER = 25
- int IVNALID_BIASADD_LAYER = 26
- int UNREMOVED_IDENTITY = 27

8.3.1 Detailed Description

AxfcError enum class.

Definition at line 20 of file AxfcError.py.

8.3.2 Member Data Documentation

8.3.2.1 DUMP_IR_GRAPH_ERROR

```
int AxfcError.AxfcError.DUMP_IR_GRAPH_ERROR = 22  [static]
```

Definition at line 49 of file AxfcError.py.

8.3.2.2 EMPTY_IR_BLOCK

```
int AxfcError.AxfcError.EMPTY_IR_BLOCK = 6  [static]
```

Definition at line 29 of file AxfcError.py.

8.3.2.3 INVALID_ACTIVATION_LAYER

```
int AxfcError.AxfcError.INVALID_ACTIVATION_LAYER = 18  [static]
```

Definition at line 45 of file AxfcError.py.

8.3.2.4 INVALID_AIX_GRAPH

```
int AxfcError.AxfcError.INVALID_AIX_GRAPH = 23  [static]
```

Definition at line 50 of file AxfcError.py.

8.3.2.5 INVALID_AIX_LAYER_TYPE

```
int AxfcError.AxfcError.INVALID_AIX_LAYER_TYPE = 13  [static]
```

Definition at line 40 of file AxfcError.py.

8.3.2.6 INVALID_AIX_TENSOR_FORMAT

```
int AxfcError.AxfcError.INVALID_AIX_TENSOR_FORMAT = 14  [static]
```

Definition at line 41 of file AxfcError.py.

8.3.2.7 INVALID_AIX_TENSOR_INPUT

```
int AxfcError.AxfcError.INVALID_AIX_TENSOR_INPUT = 21  [static]
```

Definition at line 48 of file AxfcError.py.

8.3.2.8 INVALID_BATCHNORM_LAYER

```
int AxfcError.AxfcError.INVALID_BATCHNORM_LAYER = 17  [static]
```

Definition at line 44 of file AxfcError.py.

8.3.2.9 INVALID_CONVOLUTION_LAYER

```
int AxfcError.AxfcError.INVALID_CONVOLUTION_LAYER = 15  [static]
```

Definition at line 42 of file AxfcError.py.

8.3.2.10 INVALID_EWADD_LAYER

```
int AxfcError.AxfcError.INVALID_EWADD_LAYER = 25  [static]
```

Definition at line 52 of file AxfcError.py.

8.3.2.11 INVALID_FILE_PATH

```
int AxfcError.AxfcError.INVALID_FILE_PATH = 2  [static]
```

Definition at line 23 of file AxfcError.py.

8.3.2.12 INVALID_GROUP_CONV_LAYER

```
int AxfcError.AxfcError.INVALID_GROUP_CONV_LAYER = 16  [static]
```

Definition at line 43 of file AxfcError.py.

8.3.2.13 INVALID_IDENTITY_LAYER

```
int AxfcError.AxfcError.INVALID_IDENTITY_LAYER = 19  [static]
```

Definition at line 46 of file AxfcError.py.

8.3.2.14 INVALID_INPUT_TYPE

```
int AxfcError.AxfcError.INVALID_INPUT_TYPE = 3  [static]
```

Definition at line 26 of file AxfcError.py.

8.3.2.15 INVALID_IR_GRAPH

```
int AxfcError.AxfcError.INVALID_IR_GRAPH = 5  [static]
```

Definition at line 28 of file AxfcError.py.

8.3.2.16 INVALID_MAXPOOL_LAYER

```
int AxfcError.AxfcError.INVALID_MAXPOOL_LAYER = 24  [static]
```

Definition at line 51 of file AxfcError.py.

8.3.2.17 INVALID_MD_FORMAT

```
int AxfcError.AxfcError.INVALID_MD_FORMAT = 8  [static]
```

Definition at line 33 of file AxfcError.py.

8.3.2.18 INVALID_PAD_LAYER

```
int AxfcError.AxfcError.INVALID_PAD_LAYER = 20  [static]
```

Definition at line 47 of file AxfcError.py.

8.3.2.19 INVALID_PARAMETER

```
int AxfcError.AxfcError.INVALID_PARAMETER = 1  [static]
```

Definition at line 22 of file AxfcError.py.

8.3.2.20 INVALID_TF_GRAPH

```
int AxfcError.AxfcError.INVALID_TF_GRAPH = 4  [static]
```

Definition at line 27 of file AxfcError.py.

8.3.2.21 IVNALID_BIASADD_LAYER

```
int AxfcError.AxfcError.IVNALID_BIASADD_LAYER = 26  [static]
```

Definition at line 53 of file AxfcError.py.

8.3.2.22 NOT_AIXH_SUPPORT

```
int AxfcError.AxfcError.NOT_AIXH_SUPPORT = 9  [static]
```

Definition at line 34 of file AxfcError.py.

8.3.2.23 NOT_IMPLEMENTED

```
int AxfcError.AxfcError.NOT_IMPLEMENTED = 10  [static]
```

Definition at line 35 of file AxfcError.py.

8.3.2.24 PRED_NODE_NOT_FOUND

```
int AxfcError.AxfcError.PRED_NODE_NOT_FOUND = 7  [static]
```

Definition at line 30 of file AxfcError.py.

8.3.2.25 SUCCESS

```
int AxfcError.AxfcError.SUCCESS = 0  [static]
```

Definition at line 21 of file AxfcError.py.

8.3.2.26 UNKNOWN_TENSOR_TYPE

```
int AxfcError.AxfcError.UNKNOWN_TENSOR_TYPE = 11  [static]
```

Definition at line 36 of file AxfcError.py.

8.3.2.27 UNREMOVED_IDENTITY

```
int AxfcError.AxfcError.UNREMOVED_IDENTITY = 27  [static]
```

Definition at line 54 of file AxfcError.py.

8.3.2.28 UNSUPPORTED_AIX_LAYER_EMIT

```
int AxfcError.AxfcError.UNSUPPORTED_AIX_LAYER_EMIT = 12  [static]
```

Definition at line 39 of file AxfcError.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcError.py

8.4 AxfcFrontendCompiler.AxfcFrontendCompiler Class Reference

AxfcFrontendCompiler.

Public Member Functions

- def __init__ (self)

  The constructor.
- def get_ir_graph (self)

  This method returns the IR graph.
- def read_md_file

  This method is used to read a machine description in the given path.
- def compile

  This method is used to compile an input AI network model into an AIXGraph object.
- def dump_aix_graphs

  This method is used to dump out the generated AIXGraphs.
- def dump_launcher
- def __str__ (self)

  For debugging.

Private Attributes

- __md

  machine description object
- __ir_builder

  AIXIR builder.
- __ir_translator

  AIXIR-to-AIXGraph translator.

8.4.1 Detailed Description

AxfcFrontendCompiler.

Definition at line 21 of file AxfcFrontendCompiler.py.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 __init__()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.__init__ (
              self )
```

The constructor.

Definition at line 33 of file AxfcFrontendCompiler.py.

8.4.3 Member Function Documentation

8.4.3.1 __str__()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.__str__ (
              self )
```

For debugging.

Definition at line 120 of file AxfcFrontendCompiler.py.

8.4.3.2 compile()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.compile (
              self,
              path )
```

This method is used to compile an input AI network model into an AIXGraph object.

Parameters

self	this object
path	file path of an input AI network model

Returns

error info and an AXIGraph objects

Definition at line 61 of file AxfcFrontendCompiler.py.

8.4.3.3 dump_aix_graphs()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.dump_aix_graphs (
              self,
              out_path )
```

This method is used to dump out the generated AIXGraphs.

Parameters

self	this object
out_path	a file path to output the AIXGraphs
aix_graphs	a list of AIXGraphs to be dumped out

Returns

error info

Definition at line 96 of file AxfcFrontendCompiler.py.

8.4.3.4 dump_launcher()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.dump_launcher (
              self,
              path )
```

Definition at line 116 of file AxfcFrontendCompiler.py.

8.4.3.5 get_ir_graph()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.get_ir_graph (
              self )
```

This method returns the IR graph.

Parameters

self	this object

Returns

the IR graph

Definition at line 42 of file AxfcFrontendCompiler.py.

8.4.3.6 read_md_file()

```
def AxfcFrontendCompiler.AxfcFrontendCompiler.read_md_file (
              self,
              path )
```

This method is used to read a machine description in the given path.

Parameters

self	this object
path	file path of AIXH machine description

Returns

an AXIGraph object

Definition at line 50 of file AxfcFrontendCompiler.py.

8.4.4 Member Data Documentation

8.4.4.1 __ir_builder

```
AxfcFrontendCompiler.AxfcFrontendCompiler.__ir_builder  [private]
```

AIXIR builder.

Definition at line 35 of file AxfcFrontendCompiler.py.

8.4.4.2 __ir_translator

```
AxfcFrontendCompiler.AxfcFrontendCompiler.__ir_translator  [private]
```

AIXIR-to-AIXGraph translator.

Definition at line 36 of file AxfcFrontendCompiler.py.

8.4.4.3 __md

```
AxfcFrontendCompiler.AxfcFrontendCompiler.__md  [private]
```

machine description object

Definition at line 34 of file AxfcFrontendCompiler.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcFrontendCompiler.py

8.5 AxfcGraphWriter.AxfcGraphWriter Class Reference

AxfcGraphWriter class.

Public Member Functions

- def __init__ (self)

  The constructor.
- def add_edge (self, source_node_id, target_node_id)

  This method is used insert the edges of node.
- def add_node

  This method is used insert the node.
- def write_file

  This method is used to write the edges and nodes to Sigma js json format.

Private Attributes

- __edge_id

  Edge's ID (auto increase)
- __graph

  dictionary of edges and nodes
- __nodes

  set of nodes
- __x_axis

  x axis of edges
- __y_axis

  y axis of edges

8.5.1 Detailed Description

AxfcGraphWriter class.

Definition at line 23 of file AxfcGraphWriter.py.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 __init__()

```
def AxfcGraphWriter.AxfcGraphWriter.__init__ (
              self )
```

The constructor.

Definition at line 41 of file AxfcGraphWriter.py.

8.5.3 Member Function Documentation

8.5.3.1 add_edge()

```
def AxfcGraphWriter.AxfcGraphWriter.add_edge (
              self,
              source_node_id,
              target_node_id )
```

This method is used insert the edges of node.

Parameters

self	this object
source_node←↩ _id	node's id for source
target_node_id	node's id for target

Definition at line 53 of file AxfcGraphWriter.py.

8.5.3.2 add_node()

```
def AxfcGraphWriter.AxfcGraphWriter.add_node (
              self,
              ir_node )
```

This method is used insert the node.

Parameters

self	this object
ir_node	AxfcIRNode node

Definition at line 68 of file AxfcGraphWriter.py.

8.5.3.3 write_file()

```
def AxfcGraphWriter.AxfcGraphWriter.write_file (
              self,
              file_path )
```

This method is used to write the edges and nodes to Sigma js json format.

Parameters

self	this object
file_path	file path for dumping the IR graph

Returns

error info

Definition at line 95 of file AxfcGraphWriter.py.

8.5.4 Member Data Documentation

8.5.4.1 __edge_id

```
AxfcGraphWriter.AxfcGraphWriter.__edge_id  [private]
```

Edge's ID (auto increase)

Definition at line 42 of file AxfcGraphWriter.py.

8.5.4.2 __graph

```
AxfcGraphWriter.AxfcGraphWriter.__graph  [private]
```

dictionary of edges and nodes

Definition at line 43 of file AxfcGraphWriter.py.

8.5.4.3 __nodes

```
AxfcGraphWriter.AxfcGraphWriter.__nodes  [private]
```

set of nodes

Definition at line 44 of file AxfcGraphWriter.py.

8.5.4.4 __x_axis

```
AxfcGraphWriter.AxfcGraphWriter.__x_axis  [private]
```

x axis of edges

Definition at line 45 of file AxfcGraphWriter.py.

8.5.4.5 __y_axis

```
AxfcGraphWriter.AxfcGraphWriter.__y_axis  [private]
```

y axis of edges

Definition at line 46 of file AxfcGraphWriter.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcGraphWriter.py

8.6 AxfcIRBlock.AxfcIRBlock Class Reference

AxfcIRBlock class.

Public Member Functions

- def __init__ (self)

  The constructor.
- def analyse_liveness (self)

  This method is used to perform the local liveness analysis in the scope of an IR block.
- def analyze_profit (self)

  This method is used to calculate the profit that we can achieve by accelerating this block in hardware-manner.
- def __str__ (self)

  For debugging.

Public Attributes

- id

  block ID
- nodes

  a list of nodes that make up this block
- live_in

  a list of live-in node IDs
- live_out

  a list of live-out node IDs
- is_aixh_support

  indicate whether this node can be executed in hardware-manner
- aixh_profit

  specify the profit to be obtained by using AIXH
- aix_graph

  an AIX graph emitted from this IR block

Private Member Functions

- def __analyse_inout (self)

  This method is used to find the input and output nodes of this block.

8.6.1 Detailed Description

AxfcIRBlock class.

Definition at line 20 of file AxfcIRBlock.py.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 __init__()

```
def AxfcIRBlock.AxfcIRBlock.__init__ (
              self )
```

The constructor.

Definition at line 44 of file AxfcIRBlock.py.

8.6.3 Member Function Documentation

8.6.3.1 __analyse_inout()

```
def AxfcIRBlock.AxfcIRBlock.__analyse_inout (
              self,
              AxfcError )  [private]
```

This method is used to find the input and output nodes of this block.

Parameters

self	this object

Returns

error info.

Definition at line 96 of file AxfcIRBlock.py.

8.6.3.2 __str__()

```
def AxfcIRBlock.AxfcIRBlock.__str__ (
              self )
```

For debugging.

Definition at line 162 of file AxfcIRBlock.py.

8.6.3.3 analyse_liveness()

```
def AxfcIRBlock.AxfcIRBlock.analyse_liveness (
              self,
              AxfcError )
```

This method is used to perform the local liveness analysis in the scope of an IR block.

We employ a simple heuristic scheme to find live-ins and live-outs of a block without global liveness analysis on the entire IR graph.

Parameters

self	this object

Returns

error info

Definition at line 59 of file AxfcIRBlock.py.

8.6.3.4 analyze_profit()

```
def AxfcIRBlock.AxfcIRBlock.analyze_profit (
              self,
              AxfcError )
```

This method is used to calculate the profit that we can achieve by accelerating this block in hardware-manner.

Parameters

self	this object

Returns

error info

Definition at line 140 of file AxfcIRBlock.py.

8.6.4 Member Data Documentation

8.6.4.1 aix_graph

`AxfcIRBlock.AxfcIRBlock.aix_graph`

an AIX graph emitted from this IR block

Definition at line 51 of file AxfcIRBlock.py.

8.6.4.2 aixh_profit

`AxfcIRBlock.AxfcIRBlock.aixh_profit`

specify the profit to be obtained by using AIXH

Definition at line 50 of file AxfcIRBlock.py.

8.6.4.3 id

`AxfcIRBlock.AxfcIRBlock.id`

block ID

Definition at line 45 of file AxfcIRBlock.py.

8.6.4.4 is_aixh_support

`AxfcIRBlock.AxfcIRBlock.is_aixh_support`

indicate whether this node can be executed in hardware-manner

Definition at line 49 of file AxfcIRBlock.py.

8.6.4.5 live_in

`AxfcIRBlock.AxfcIRBlock.live_in`

a list of live-in node IDs

Definition at line 47 of file AxfcIRBlock.py.

8.6.4.6 live_out

`AxfcIRBlock.AxfcIRBlock.live_out`

a list of live-out node IDs

Definition at line 48 of file AxfcIRBlock.py.

8.6.4.7 nodes

`AxfcIRBlock.AxfcIRBlock.nodes`

a list of nodes that make up this block

Definition at line 46 of file AxfcIRBlock.py.

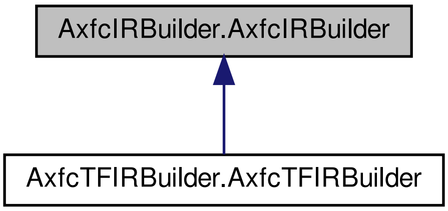
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBlock.py

8.7 AxfcIRBuilder.AxfcIRBuilder Class Reference

AxfcIRBuilder class.

Inheritance diagram for AxfcIRBuilder.AxfcIRBuilder:



Public Member Functions

- def __init__ (self, md)
  The constructor.
- def build_ir
  This method is used to build AXI IR.
- def __str__ (self)
  For debugging.

Private Member Functions

- def __find_aixh_blocks (self)

  This method is used to find AIXH blocks comprised of AIXH-supported nodes.
- def __perform_maximal_munch

  This method performs maximal munch algorithm to recursively find the longest successive AIXH-supported nodes.
- def _read_model_graph

  Abstract methods.
- def _build_naive_ir

  This method is used to construct a naive AIXIR using a tensorflow graph.

Private Attributes

- _md

  AIX machine description.
- _tf_graph

  input Tensorflow graph
- _ir_graph

  output AIXIR graph
- _ir_symtab

  symbol table for IR graph

8.7.1 Detailed Description

AxfcIRBuilder class.

Definition at line 21 of file AxfcIRBuilder.py.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 __init__()

```
def AxfcIRBuilder.AxfcIRBuilder.__init__ (
              self,
              md )
```

The constructor.

Definition at line 36 of file AxfcIRBuilder.py.

8.7.3 Member Function Documentation

8.7.3.1 __find_aixh_blocks()

```
def AxfcIRBuilder.AxfcIRBuilder.__find_aixh_blocks (
              self,
              AxfcError )  [private]
```

This method is used to find AIXH blocks comprised of AIXH-supported nodes.

We employ a maximal munching scheme to find the longest successive AIXH-supported nodes and build up a block with the nodes.

Parameters

self	this object

Returns

error info

Definition at line 95 of file AxfcIRBuilder.py.

8.7.3.2 __perform_maximal_munch()

```
def AxfcIRBuilder.AxfcIRBuilder.__perform_maximal_munch (
              self,
              ir_node )  [private]
```

This method performs maximal munch algorithm to recursively find the longest successive AIXH-supported nodes.

Parameters

self	this object
ir_node	a start node to perform maximal munching
an	IR block of the successive IR nodes supported by the AIX hardware

Returns

error info

Definition at line 141 of file AxfcIRBuilder.py.

8.7.3.3 __str__()

```
def AxfcIRBuilder.AxfcIRBuilder.__str__ (
              self )
```

For debugging.

Definition at line 172 of file AxfcIRBuilder.py.

8.7.3.4 _build_naive_ir()

```
def AxfcIRBuilder.AxfcIRBuilder._build_naive_ir (
              self,
              path )  [private]
```

This method is used to construct a naive AIXIR using a tensorflow graph.

Parameters

self	this object
path	file path of input network model

Returns

error info

Definition at line 192 of file AxfcIRBuilder.py.

8.7.3.5 _read_model_graph()

```
def AxfcIRBuilder.AxfcIRBuilder._read_model_graph (
              self,
              path )  [private]
```

Abstract methods.

This method is used to read a tensorflow graph from an input file in the given path.

Parameters

self	this object
path	file path of input network model

Returns

error info

Definition at line 184 of file AxfcIRBuilder.py.

8.7.3.6 build_ir()

```
def AxfcIRBuilder.AxfcIRBuilder.build_ir (
              self,
              path )
```

This method is used to build AXI IR.

1) it builds a naive IR using the given input model. 2) it checks the IR nodes to be executed in hardware-manner. 3) it finds AIXH IR blocks. each block consist of several AIXH IR nodes. 4) it performs the liveness analysis for resolving the input and output of the blocks.

Parameters

self	this object
path	input path of a frozen model

Returns

error info and an AxfcIRGraph object

Definition at line 51 of file AxfcIRBuilder.py.

8.7.4 Member Data Documentation

8.7.4.1 _ir_graph

```
AxfcIRBuilder.AxfcIRBuilder._ir_graph  [private]
```

output AIXIR graph

Definition at line 39 of file AxfcIRBuilder.py.

8.7.4.2 _ir_symtab

```
AxfcIRBuilder.AxfcIRBuilder._ir_symtab  [private]
```

symbol table for IR graph

Definition at line 40 of file AxfcIRBuilder.py.

8.7.4.3 _md

```
AxfcIRBuilder.AxfcIRBuilder._md  [private]
```

AIX machine description.

Definition at line 37 of file AxfcIRBuilder.py.

8.7.4.4 _tf_graph

```
AxfcIRBuilder.AxfcIRBuilder._tf_graph  [private]
```

input Tensorflow graph

Definition at line 38 of file AxfcIRBuilder.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBuilder.py

8.8 AxfcIRGraph.AxfcIRGraph Class Reference

AxfcIRGraph class.

Public Member Functions

- def __init__

  The constructor.
- def append_node

  This method is used to append the given IR node into the graph.
- def append_block

  This method is used to append the given IR block into the graph.
- def analyse_liveness (self)

  This method is used to perform the liveness analysis of this graph.
- def dump_to_file

  This method is used to visualize the IR graph using Sigma js.
- def __str__ (self)

  For debugging.

Public Attributes

- root_node

  output root node of this graph
- nodes

  a list of nodes consisting this graph
- blocks

  a list of blocks that are contained this graph
- symtab

  a reference to an IR symbol table

8.8.1 Detailed Description

AxfcIRGraph class.

Definition at line 21 of file AxfcIRGraph.py.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 __init__()

```
def AxfcIRGraph.AxfcIRGraph.__init__ (
              self,
              symtab )
```

The constructor.

Parameters

self	this object
symtab	a symbol table for referring to an IR node using its name

Definition at line 38 of file AxfcIRGraph.py.

8.8.3 Member Function Documentation

8.8.3.1 __str__()

```
def AxfcIRGraph.AxfcIRGraph.__str__ (
              self )
```

For debugging.

Definition at line 120 of file AxfcIRGraph.py.

8.8.3.2 analyse_liveness()

```
def AxfcIRGraph.AxfcIRGraph.analyse_liveness (
              self,
              AxfcError )
```

This method is used to perform the liveness analysis of this graph.

Parameters

self	this object

Returns

error info

Definition at line 75 of file AxfcIRGraph.py.

8.8.3.3 append_block()

```
def AxfcIRGraph.AxfcIRGraph.append_block (
              self,
              ir_block )
```

This method is used to append the given IR block into the graph.

Parameters

self	this object
ir_block	an IR block to be appended

Definition at line 65 of file AxfcIRGraph.py.

8.8.3.4 append_node()

```
def AxfcIRGraph.AxfcIRGraph.append_node (
              self,
              ir_node )
```

This method is used to append the given IR node into the graph.

Parameters

self	this object
ir_node	an IR node to be appended

Definition at line 50 of file AxfcIRGraph.py.

8.8.3.5 dump_to_file()

```
def AxfcIRGraph.AxfcIRGraph.dump_to_file (
              self,
              file_path )
```

This method is used to visualize the IR graph using Sigma js.

Parameters

self	this object
file_path	a file path to dump out the IR graph
ignore_ops	a list of operations to be ignored

Returns

error info

Definition at line 90 of file AxfcIRGraph.py.

8.8.4 Member Data Documentation

8.8.4.1 blocks

`AxfcIRGraph.AxfcIRGraph.blocks`

a list of blocks that are contained this graph

Definition at line 43 of file AxfcIRGraph.py.

8.8.4.2 nodes

`AxfcIRGraph.AxfcIRGraph.nodes`

a list of nodes consisting this graph

Definition at line 42 of file AxfcIRGraph.py.

8.8.4.3 root_node

`AxfcIRGraph.AxfcIRGraph.root_node`

output root node of this graph

Definition at line 39 of file AxfcIRGraph.py.

8.8.4.4 symtab

`AxfcIRGraph.AxfcIRGraph.symtab`

a reference to an IR symbol table

Definition at line 44 of file AxfcIRGraph.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRGraph.py

8.9 AxfcIRNode.AxfcIRNode Class Reference

AxfcIRNode.

Public Member Functions

- def __init__ (self, node_def)

  The constructor.
- def analyze_profit (self)

  This method is used to calculate and return the profit that we can get by accelerating the operation of this node in hardware-manner.
- def __eq__ (self, other)

  This methods is used to compare id with equal (==) for using Set.
- def __hash__ (self)

  This methods make this object become hasable by id.
- def __del__ (self)

  Destructor.
- def __str__ (self)

  For debugging.

Public Attributes

- id

  node ID
- name

  node name
- layer_id

  node layer ID
- succs

  node a list of successor nodes
- preds

  node a list of predecessor nodes
- node_def

  node a reference to an input node object
- block_ref

  reference to the IR block that contains this node

- aixh_profit

  specify the profit to be obtained by using AIXH
- is_aixh_support

  indicate whether this node can be executed in hardware-manner
- eval_flag

  indicate whether this node has been already evaluated or not for maximal munching
- is_input

  indicate whether this node is an input node or not
- is_output

  indicate whether this node is an output node or not
- aix_layer

  reference to the AIX layer derived from this node
- op

  operation of this node

8.9.1 Detailed Description

AxfcIRNode.

Created: 2020. 08. 03

Authors: Youngsun Han (youngsun@pknu.ac.kr) Heng Sengthai (sengthai37@gmail.com)

High Performance Computing Laboratory (hpcl.pknu.ac.kr) AxfcIRNode class

Definition at line 17 of file AxfcIRNode.py.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 __init__()

```
def AxfcIRNode.AxfcIRNode.__init__ (
              self,
              node_def )
```

The constructor.

Definition at line 62 of file AxfcIRNode.py.

8.9.2.2 __del__()

```
def AxfcIRNode.AxfcIRNode.__del__ (
              self )
```

Destructor.

Definition at line 115 of file AxfcIRNode.py.

8.9.3 Member Function Documentation

8.9.3.1 __eq__()

```
def AxfcIRNode.AxfcIRNode.__eq__ (
              self,
              other )
```

This methods is used to compare id with equal (==) for using Set.

Parameters

self	this object
other	another AxfcIRNode object

Definition at line 100 of file AxfcIRNode.py.

8.9.3.2 __hash__()

```
def AxfcIRNode.AxfcIRNode.__hash__ (
              self )
```

This methods make this object become hasable by id.

Parameters

self	this object

Definition at line 111 of file AxfcIRNode.py.

8.9.3.3 __str__()

```
def AxfcIRNode.AxfcIRNode.__str__ (
              self )
```

For debugging.

Definition at line 124 of file AxfcIRNode.py.

8.9.3.4 analyze_profit()

```
def AxfcIRNode.AxfcIRNode.analyze_profit (
              self,
              int )
```

This method is used to calculate and return the profit that we can get by accelerating the operation of this node in hardware-manner.

Parameters

self	this object

Returns

the calculated profit

Definition at line 85 of file AxfcIRNode.py.

8.9.4 Member Data Documentation

8.9.4.1 aix_layer

```
AxfcIRNode.AxfcIRNode.aix_layer
```

reference to the AIX layer derived from this node

Definition at line 78 of file AxfcIRNode.py.

8.9.4.2 aixh_profit

```
AxfcIRNode.AxfcIRNode.aixh_profit
```

specify the profit to be obtained by using AIXH

Definition at line 71 of file AxfcIRNode.py.

8.9.4.3 block_ref

```
AxfcIRNode.AxfcIRNode.block_ref
```

reference to the IR block that contains this node

Definition at line 69 of file AxfcIRNode.py.

8.9.4.4 eval_flag

`AxfcIRNode.AxfcIRNode.eval_flag`

indicate whether this node has been already evaluated or not for maximal munching

Definition at line 73 of file AxfcIRNode.py.

8.9.4.5 id

`AxfcIRNode.AxfcIRNode.id`

node ID

Definition at line 63 of file AxfcIRNode.py.

8.9.4.6 is_aixh_support

`AxfcIRNode.AxfcIRNode.is_aixh_support`

indicate whether this node can be executed in hardware-manner

Definition at line 72 of file AxfcIRNode.py.

8.9.4.7 is_input

`AxfcIRNode.AxfcIRNode.is_input`

indicate whether this node is an input node or not

Definition at line 75 of file AxfcIRNode.py.

8.9.4.8 is_output

`AxfcIRNode.AxfcIRNode.is_output`

indicate whether this node is an output node or not

Definition at line 76 of file AxfcIRNode.py.

8.9.4.9 layer_id

```
AxfcIRNode.AxfcIRNode.layer_id
```

node layer ID

Definition at line 65 of file AxfcIRNode.py.

8.9.4.10 name

```
AxfcIRNode.AxfcIRNode.name
```

node name

Definition at line 64 of file AxfcIRNode.py.

8.9.4.11 node_def

```
AxfcIRNode.AxfcIRNode.node_def
```

node a reference to an input node object

Definition at line 68 of file AxfcIRNode.py.

8.9.4.12 op

```
AxfcIRNode.AxfcIRNode.op
```

operation of this node

Definition at line 117 of file AxfcIRNode.py.

8.9.4.13 preds

```
AxfcIRNode.AxfcIRNode.preds
```

node a list of predecessor nodes

Definition at line 67 of file AxfcIRNode.py.

8.9.4.14 succs

```
AxfcIRNode.AxfcIRNode.succs
```

node a list of successor nodes

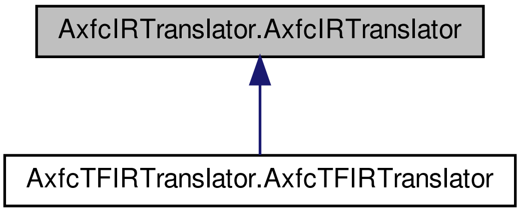
Definition at line 66 of file AxfcIRNode.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRNode.py

8.10 AxfcIRTranslator.AxfcIRTranslator Class Reference

Inheritance diagram for AxfcIRTranslator.AxfcIRTranslator:


AxfcIRTranslator.AxfcIRTranslator
AxfcTFIRTranslator.AxfcTFIRTranslator

Public Member Functions

- def __init__ (self, md)

  The constructor.
- def emit_aixh_graphs

  This method translates IR blocks of the given IR graph into AIXGraphs and return them.
- def __str__ (self)

  For debugging.

Public Attributes

- aix_graphs

  a list of AIXGraphs translated from an input model

Private Member Functions

- def __emit_aixh_block

  This method is used to translate an IR block into an AIXGraph.
- def __emit_aixh_node

  This method is used to translate an IR node into an AIXLayer object.
- def _get_emitted_input_nodes

  This method is used to return a list of already emitted input nodes.
- def _emit_aix_layer_convolution

  Abstract methods.
- def _emit_aix_layer_group_conv
- def _emit_aix_layer_batchnorm
- def _emit_aix_layer_maxpool
- def _emit_aix_layer_ewadd
- def _emit_aix_layer_avgpool
- def _emit_aix_layer_biasadd
- def _emit_aix_layer_softmax
- def _emit_aix_layer_activation
- def _emit_aix_tensor_input

  emission methods for AIX tensors
- def _emit_aix_tensor_filter
- def _emit_aix_tensor_bias
- def _emit_aix_tensor_scale
- def _emit_aix_tensor_mean
- def _emit_aix_tensor_variance
- def _emit_aix_tensor_output
- def _emit_aix_convolution_desc

  emission methods for AIX convolution dec
- def _emit_aix_sampling_desc

  emission methods for AIX sampling dec

Private Attributes

- _md

  AIX machine description.
- _ir_symtab

  a symbol table of pairs of an IR node's name and itself
- __emit_aix_layer_tbl

  a dictionary of pairs of AIXLayerType and its AIX layer emission method
- _aix_graph

  the current AIX graph being translated

8.10.1 Detailed Description

Definition at line 37 of file AxfcIRTranslator.py.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 __init__()

```
def AxfcIRTranslator.AxfcIRTranslator.__init__ (
              self,
              md )
```

The constructor.

Definition at line 55 of file AxfcIRTranslator.py.

8.10.3 Member Function Documentation

8.10.3.1 __emit_aixh_block()

```
def AxfcIRTranslator.AxfcIRTranslator.__emit_aixh_block (
              self,
              ir_block )  [private]
```

This method is used to translate an IR block into an AIXGraph.

Parameters

self	this object
ir_block	input IR block

Returns

error info and an output AIXGraph

Definition at line 107 of file AxfcIRTranslator.py.

8.10.3.2 __emit_aixh_node()

```
def AxfcIRTranslator.AxfcIRTranslator.__emit_aixh_node (
              self,
              ir_node )  [private]
```

This method is used to translate an IR node into an AIXLayer object.

Parameters

self	this object
ir_node	input IR node to be translated

Returns

error info and an output AIXLayer object

Definition at line 142 of file AxfcIRTranslator.py.

8.10.3.3 __str__()

```
def AxfcIRTranslator.AxfcIRTranslator.__str__ (
              self )
```

For debugging.

Definition at line 234 of file AxfcIRTranslator.py.

8.10.3.4 _emit_aix_convolution_desc()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_convolution_desc (
              self,
              ir_node )  [private]
```

emission methods for AIX convolution dec

Definition at line 292 of file AxfcIRTranslator.py.

8.10.3.5 _emit_aix_layer_activation()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_activation (
              self,
              ir_node )  [private]
```

Definition at line 266 of file AxfcIRTranslator.py.

8.10.3.6 _emit_aix_layer_avgpool()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_avgpool (
              self,
              ir_node )  [private]
```

Definition at line 257 of file AxfcIRTranslator.py.

8.10.3.7 _emit_aix_layer_batchnorm()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_batchnorm (
              self,
              ir_node )  [private]
```

Definition at line 248 of file AxfcIRTranslator.py.

8.10.3.8 _emit_aix_layer_biasadd()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_biasadd (
              self,
              ir_node )  [private]
```

Definition at line 260 of file AxfcIRTranslator.py.

8.10.3.9 _emit_aix_layer_convolution()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_convolution (
              self,
              ir_node )  [private]
```

Abstract methods.

emission methods for AIX layers

Definition at line 242 of file AxfcIRTranslator.py.

8.10.3.10 _emit_aix_layer_ewadd()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_ewadd (
              self,
              ir_node )  [private]
```

Definition at line 254 of file AxfcIRTranslator.py.

8.10.3.11 _emit_aix_layer_group_conv()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_group_conv (
              self,
              ir_node )  [private]
```

Definition at line 245 of file AxfcIRTranslator.py.

8.10.3.12 _emit_aix_layer_maxpool()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_maxpool (
              self,
              ir_node )  [private]
```

Definition at line 251 of file AxfcIRTranslator.py.

8.10.3.13 _emit_aix_layer_softmax()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_layer_softmax (
              self,
              ir_node )  [private]
```

Definition at line 263 of file AxfcIRTranslator.py.

8.10.3.14 _emit_aix_sampling_desc()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_sampling_desc (
              self,
              ir_node )  [private]
```

emission methods for AIX sampling dec

Definition at line 296 of file AxfcIRTranslator.py.

8.10.3.15 _emit_aix_tensor_bias()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_bias (
              self,
              ir_node )  [private]
```

Definition at line 276 of file AxfcIRTranslator.py.

8.10.3.16 _emit_aix_tensor_filter()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_filter (
              self,
              ir_node )  [private]
```

Definition at line 273 of file AxfcIRTranslator.py.

8.10.3.17 _emit_aix_tensor_input()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_input (
              self,
              ir_node )  [private]
```

emission methods for AIX tensors

Definition at line 270 of file AxfcIRTranslator.py.

8.10.3.18 _emit_aix_tensor_mean()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_mean (
              self,
              ir_node )  [private]
```

Definition at line 282 of file AxfcIRTranslator.py.

8.10.3.19 _emit_aix_tensor_output()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_output (
              self,
              ir_node )  [private]
```

Definition at line 288 of file AxfcIRTranslator.py.

8.10.3.20 _emit_aix_tensor_scale()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_scale (
              self,
              ir_node )  [private]
```

Definition at line 279 of file AxfcIRTranslator.py.

8.10.3.21 _emit_aix_tensor_variance()

```
def AxfcIRTranslator.AxfcIRTranslator._emit_aix_tensor_variance (
              self,
              ir_node )  [private]
```

Definition at line 285 of file AxfcIRTranslator.py.

8.10.3.22 _get_emitted_input_nodes()

```
def AxfcIRTranslator.AxfcIRTranslator._get_emitted_input_nodes (
              self,
              ir_node )  [private]
```

This method is used to return a list of already emitted input nodes.

If there are input nodes that have not translated yet, we perform __emit_aixh_node method repeatedly to emit them all.

Parameters

self	this object
ir_node	current node to emit its input nodes

Returns

a list of emitted input nodes

Definition at line 206 of file AxfcIRTranslator.py.

8.10.3.23 emit_aixh_graphs()

```
def AxfcIRTranslator.AxfcIRTranslator.emit_aixh_graphs (
              self,
              ir_graph )
```

This method translates IR blocks of the given IR graph into AIXGraphs and return them.

Parameters

self	this object
ir_graph	input IR graph

Returns

error info and a list of AIXGraphs

Definition at line 78 of file AxfcIRTranslator.py.

8.10.4 Member Data Documentation

8.10.4.1 __emit_aix_layer_tbl

```
AxfcIRTranslator.AxfcIRTranslator.__emit_aix_layer_tbl  [private]
```

a dictionary of pairs of AIXLayerType and its AIX layer emission method

Definition at line 60 of file AxfcIRTranslator.py.

8.10.4.2 _aix_graph

```
AxfcIRTranslator.AxfcIRTranslator._aix_graph  [private]
```

the current AIX graph being translated

Definition at line 111 of file AxfcIRTranslator.py.

8.10.4.3 _ir_symtab

```
AxfcIRTranslator.AxfcIRTranslator._ir_symtab  [private]
```

a symbol table of pairs of an IR node's name and itself

Definition at line 58 of file AxfcIRTranslator.py.

8.10.4.4 _md

```
AxfcIRTranslator.AxfcIRTranslator._md  [private]
```

AIX machine description.

Definition at line 56 of file AxfcIRTranslator.py.

8.10.4.5 aix_graphs

```
AxfcIRTranslator.AxfcIRTranslator.aix_graphs
```

a list of AIXGraphs translated from an input model

Definition at line 57 of file AxfcIRTranslator.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRTranslator.py

8.11 AxfcLauncherWriter.AxfcLauncherWriter Class Reference

AxfcLauncherWriter class.

Public Member Functions

- def __init__

  The constructor.
- def emit_aixh_launcher (self)

  This method is used to emit a launcher for the generated AIXGraph.
- def __str__ (self)

  For debugging.

Private Attributes

- __md

  AIX machine description.
- __ir_graph

  an AIXIR graph that will be used for writing the launcher

8.11.1 Detailed Description

AxfcLauncherWriter class.

Definition at line 20 of file AxfcLauncherWriter.py.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 __init__()

```
def AxfcLauncherWriter.AxfcLauncherWriter.__init__ (
              self,
              md )
```

The constructor.

Definition at line 29 of file AxfcLauncherWriter.py.

8.11.3 Member Function Documentation

8.11.3.1 __str__()

```
def AxfcLauncherWriter.AxfcLauncherWriter.__str__ (
              self )
```

For debugging.

Definition at line 40 of file AxfcLauncherWriter.py.

8.11.3.2 emit_aixh_launcher()

```
def AxfcLauncherWriter.AxfcLauncherWriter.emit_aixh_launcher (
              self )
```

This method is used to emit a launcher for the generated AIXGraph.

Parameters

self	this object

Definition at line 35 of file AxfcLauncherWriter.py.

8.11.4 Member Data Documentation

8.11.4.1 __ir_graph

```
AxfcLauncherWriter.AxfcLauncherWriter.__ir_graph  [private]
```

an AIXIR graph that will be used for writing the launcher

Definition at line 31 of file AxfcLauncherWriter.py.

8.11.4.2 __md

```
AxfcLauncherWriter.AxfcLauncherWriter.__md  [private]
```

AIX machine description.

Definition at line 30 of file AxfcLauncherWriter.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcLauncherWriter.py

8.12 AxfcMachineDesc.AxfcMachineDesc Class Reference

AxfcMachineDesc class.

Classes

- class AIXLayerInfo

  AIXLayerInfo inner class.

Public Member Functions

- def __init__ (self)

  The constructor.
- def read_file

  This method is used to read a machine description from the given input path.
- def get_layer_info

  This method returns the information of a specific AIX layer.
- def get_aixh_support

  This method indicates whether the given operation is supported by the AIX hardware or not.
- def get_in_type (self)

  This method returns the type of AI framework.
- def get_profit_threshold (self)

  This method returns the input type of the frontend compilation.
- def __str__ (self)

  For debugging.

Static Public Attributes

- int TYPE_TENSORFLOW = 0
- int TYPE_PYTORCH = 1
- int TYPE_MXNET = 2
- int TYPE_UNKNOWN = 3
- int DEFAULT_PROFIT_THRESHOLD = 1000

Private Attributes

- __aix_model_info_tbl

  general info.
- __aix_layer_info_tbl

  general machine description info.

8.12.1 Detailed Description

AxfcMachineDesc class.

Definition at line 23 of file AxfcMachineDesc.py.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 __init__()

```
def AxfcMachineDesc.AxfcMachineDesc.__init__ (
              self )
```

The constructor.

Definition at line 40 of file AxfcMachineDesc.py.

8.12.3 Member Function Documentation

8.12.3.1 __str__()

```
def AxfcMachineDesc.AxfcMachineDesc.__str__ (
              self )
```

For debugging.

Definition at line 138 of file AxfcMachineDesc.py.

8.12.3.2 get_aixh_support()

```
def AxfcMachineDesc.AxfcMachineDesc.get_aixh_support (
              self,
              layer_type )
```

This method indicates whether the given operation is supported by the AIX hardware or not.

Parameters

self	this object
layer_type	the name of an AIX layer type to be checked

Returns

the input type of the frontend compilation

Definition at line 100 of file AxfcMachineDesc.py.

8.12.3.3 get_in_type()

```
def AxfcMachineDesc.AxfcMachineDesc.get_in_type (
              self )
```

This method returns the type of AI framework.

Parameters

self	this object

Returns

the input type of the frontend compilation

Definition at line 111 of file AxfcMachineDesc.py.

8.12.3.4 get_layer_info()

```
def AxfcMachineDesc.AxfcMachineDesc.get_layer_info (
              self,
              layer_type )
```

This method returns the information of a specific AIX layer.

Parameters

self	this object
layer_type	the name of an AIX layer type to be returned

Returns

an operation information

Definition at line 88 of file AxfcMachineDesc.py.

8.12.3.5 get_profit_threshold()

```
def AxfcMachineDesc.AxfcMachineDesc.get_profit_threshold (
              self )
```

This method returns the input type of the frontend compilation.

Parameters

self	this object

Returns

the profit threshold to determine whether to use hardware acceleration

Definition at line 128 of file AxfcMachineDesc.py.

8.12.3.6 read_file()

```
def AxfcMachineDesc.AxfcMachineDesc.read_file (
              self,
              path )
```

This method is used to read a machine description from the given input path.

Parameters

self	this object
path	file path of the machine description

Returns

error info

Definition at line 49 of file AxfcMachineDesc.py.

8.12.4 Member Data Documentation

8.12.4.1 __aix_layer_info_tbl

```
AxfcMachineDesc.AxfcMachineDesc.__aix_layer_info_tbl  [private]
```

general machine description info.

of dictionary type

Definition at line 42 of file AxfcMachineDesc.py.

8.12.4.2 __aix_model_info_tbl

```
AxfcMachineDesc.AxfcMachineDesc.__aix_model_info_tbl  [private]
```

general info.

of dictionary type for AIX compiler

Definition at line 41 of file AxfcMachineDesc.py.

8.12.4.3 DEFAULT_PROFIT_THRESHOLD

```
int AxfcMachineDesc.AxfcMachineDesc.DEFAULT_PROFIT_THRESHOLD = 1000  [static]
```

Definition at line 31 of file AxfcMachineDesc.py.

8.12.4.4 TYPE_MXNET

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_MXNET = 2  [static]
```

Definition at line 27 of file AxfcMachineDesc.py.

8.12.4.5 TYPE_PYTORCH

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_PYTORCH = 1  [static]
```

Definition at line 26 of file AxfcMachineDesc.py.

8.12.4.6 TYPE_TENSORFLOW

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_TENSORFLOW = 0  [static]
```

Definition at line 25 of file AxfcMachineDesc.py.

8.12.4.7 TYPE_UNKNOWN

```
int AxfcMachineDesc.AxfcMachineDesc.TYPE_UNKNOWN = 3  [static]
```

Definition at line 28 of file AxfcMachineDesc.py.

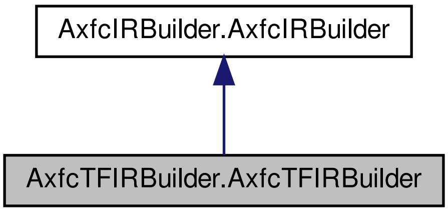
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMachineDesc.py

8.13 AxfcTFIRBuilder.AxfcTFIRBuilder Class Reference

AxfcTFIRBuilder class.

Inheritance diagram for AxfcTFIRBuilder.AxfcTFIRBuilder:



Public Member Functions

- def __init__ (self, md)

  The constructor.
- def __str__ (self)

  For debugging.

Private Member Functions

- def _read_model_graph

  This method is used to read a tensorflow graph from an input file in the given path.
- def _build_naive_ir

  This method is used to construct a naive AIXIR using a tensorflow graph.
- def __prune_ir_nodes (self)

  This method is used to prune unnecessary nodes from the IR graph.
- def __append_node_def

Private Attributes

- _tf_graph

8.13.1 Detailed Description

AxfcTFIRBuilder class.

Definition at line 23 of file AxfcTFIRBuilder.py.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 __init__()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__init__ (
              self,
              md )
```

The constructor.

Definition at line 26 of file AxfcTFIRBuilder.py.

8.13.3 Member Function Documentation

8.13.3.1 __append_node_def()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__append_node_def (
              self,
              tf_node_def )  [private]
```

Definition at line 141 of file AxfcTFIRBuilder.py.

8.13.3.2 __prune_ir_nodes()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__prune_ir_nodes (
              self,
              AxfcError )  [private]
```

This method is used to prune unnecessary nodes from the IR graph.

Currently, we will remove "identity" and "pad" nodes for the IR translation.

Parameters

self	this object

Returns

error info

Definition at line 83 of file AxfcTFIRBuilder.py.

8.13.3.3 __str__()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder.__str__ (
              self )
```

For debugging.

Definition at line 182 of file AxfcTFIRBuilder.py.

8.13.3.4 _build_naive_ir()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder._build_naive_ir (
              self,
              path )  [private]
```

This method is used to construct a naive AIXIR using a tensorflow graph.

Parameters

self	this object
path	file path of input network model

Returns

error info

Definition at line 55 of file AxfcTFIRBuilder.py.

8.13.3.5 _read_model_graph()

```
def AxfcTFIRBuilder.AxfcTFIRBuilder._read_model_graph (
              self,
              path )  [private]
```

This method is used to read a tensorflow graph from an input file in the given path.

Parameters

self	this object
path	file path of input network model

Returns

error info

Definition at line 34 of file AxfcTFIRBuilder.py.

8.13.4 Member Data Documentation

8.13.4.1 _tf_graph

```
AxfcTFIRBuilder.AxfcTFIRBuilder._tf_graph  [private]
```

Definition at line 46 of file AxfcTFIRBuilder.py.

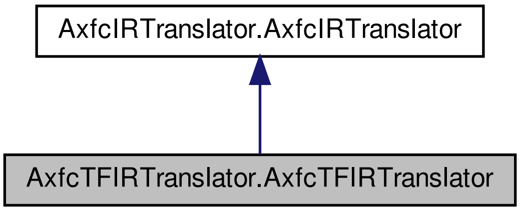
The documentation for this class was generated from the following file:

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRBuilder.py

8.14 AxfcTFIRTranslator.AxfcTFIRTranslator Class Reference

AxfcTFIRTranslator class.

Inheritance diagram for AxfcTFIRTranslator.AxfcTFIRTranslator:


AxfcIRTranslator.AxfcIRTranslator
AxfcTFIRTranslator.AxfcTFIRTranslator

Public Member Functions

- def __init__ (self, md)

  The constructor.

Private Member Functions

- def __get_aix_data_type

  private methods
- def __get_aix_tensor_format

  This method returns the tensor format of the given node_def.
- def __get_aix_tensor_dims

  This method get aixtensor dims from format as dictionary.
- def __get_values_of_format

  This method get data from aix_tensor_format format as dictionary.
- def _emit_aix_layer_convolution

  protected methods
- def _emit_aix_layer_group_conv

  This method emits some tensorflow-specific information of the given IR node into the given AIX group convolution layer object.
- def _emit_aix_layer_batchnorm

  This method emits some tensorflow-specific information of the given IR node into the given AIX batchnorm layer object.
- def _emit_aix_layer_avgpool

  This method emits some tensorflow-specific information of the given IR node into the given AIX avgpool layer object.
- def _emit_aix_layer_maxpool

  This method emits some tensorflow-specific information of the given IR node into the given AIX maxpool layer object.
- def _emit_aix_layer_ewadd

  This method emits some tensorflow-specific information of the given IR node into the given AIX element-wise add (ewadd) layer object.
- def _emit_aix_layer_softmax

  This method emits some tensorflow-specific information of the given IR node into the given AIX softmax layer object.
- def _emit_aix_layer_biasadd

  This method emits some tensorflow-specific information of the given IR node into the given AIX biasadd layer object.
- def _emit_aix_layer_activation

  This method emits some tensorflow-specific information of the given IR node into the given AIX activation layer object.
- def _emit_aix_tensor_input

  This method emits an AIX tensor of an input type from the given IR node.
- def _emit_aix_tensor_filter

  This method emits an AIX tensor of an filter type from the given IR node.
- def _emit_aix_tensor_bias

  This method emits an AIX tensor of an bias type from the given IR node.
- def _emit_aix_tensor_scale

  This method emits an AIX tensor of an scale type from the given IR node.
- def _emit_aix_tensor_mean

  This method emits an AIX tensor of an mean type from the given IR node.
- def _emit_aix_tensor_variance

  This method emits an AIX tensor of an variance type from the given IR node.
- def _emit_aix_tensor_output

  This method emits an AIX tensor of an output type from the given IR node.
- def _emit_aix_convolution_desc

  This method emits the AIX convolution description of the given IR node.
- def _emit_aix_sampling_desc

  This method emits the AIX sampling description of the given IR node.

Additional Inherited Members

8.14.1 Detailed Description

AxfcTFIRTranslator class.

Definition at line 45 of file AxfcTFIRTranslator.py.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 __init__()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__init__ (
              self,
              md )
```

The constructor.

Definition at line 48 of file AxfcTFIRTranslator.py.

8.14.3 Member Function Documentation

8.14.3.1 __get_aix_data_type()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_aix_data_type (
              self,
              tf_node_def )  [private]
```

private methods

This method returns the data type of the given node_def

Parameters

tf_node_def	input node_def

Returns

error info.

Definition at line 59 of file AxfcTFIRTranslator.py.

8.14.3.2 __get_aix_tensor_dims()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_aix_tensor_dims (
              self,
              aix_tensor )  [private]
```

This method get aixtensor dims from format as dictionary.

Parameters

self	this object
AIXTensor	an an AIX tensor data contains dims, data format, dtype, size and ptr

Returns

a dictionary object has key as element of data format. e.g input['H'] = 2

Definition at line 106 of file AxfcTFIRTranslator.py.

8.14.3.3 __get_aix_tensor_format()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_aix_tensor_format (
              self,
              tf_node_def )  [private]
```

This method returns the tensor format of the given node_def.

Parameters

tf_node_def	input node_def

Returns

error info.

Definition at line 83 of file AxfcTFIRTranslator.py.

8.14.3.4 __get_values_of_format()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator.__get_values_of_format (
              self,
              values )  [private]
```

This method get data from aix_tensor_format format as dictionary.

Parameters

self	this object
values	an list of input values
tensor_format	an AIX tensor format

Returns

a dictionary object has key as element of data format. e.g input['H'] = 2

Definition at line 116 of file AxfcTFIRTranslator.py.

8.14.3.5 _emit_aix_convolution_desc()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_convolution_desc (
              self,
              ir_node ) [private]
```

This method emits the AIX convolution description of the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor

Returns

an AIX convolution description

Definition at line 1257 of file AxfcTFIRTranslator.py.

8.14.3.6 _emit_aix_layer_activation()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_activation (
              self,
              ir_node ) [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX activation layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX activation layer

Definition at line 853 of file AxfcTFIRTranslator.py.

8.14.3.7 _emit_aix_layer_avgpool()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_avgpool (
              self,
              ir_node )  [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX avgpool layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX avgpool layer

Definition at line 488 of file AxfcTFIRTranslator.py.

8.14.3.8 _emit_aix_layer_batchnorm()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_batchnorm (
              self,
              ir_node )  [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX batchnorm layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX batchnorm layer

Definition at line 400 of file AxfcTFIRTranslator.py.

8.14.3.9 _emit_aix_layer_biasadd()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_biasadd (
              self,
              ir_node )  [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX biasadd layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX avgpool layer

Definition at line 801 of file AxfcTFIRTranslator.py.

8.14.3.10 _emit_aix_layer_convolution()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_convolution (
              self,
              ir_node )  [private]
```

protected methods

This method emits some tensorflow-specific information of the given IR node into the given AIX convolution layer object. The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX convolution layer

Definition at line 139 of file AxfcTFIRTranslator.py.

8.14.3.11 _emit_aix_layer_ewadd()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_ewadd (
              self,
              ir_node )  [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX element-wise add (ewadd) layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX avgpool layer

Definition at line 658 of file AxfcTFIRTranslator.py.

8.14.3.12 _emit_aix_layer_group_conv()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_group_conv (
              self,
              ir_node )  [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX group convolution layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX convolution layer

Definition at line 268 of file AxfcTFIRTranslator.py.

8.14.3.13 _emit_aix_layer_maxpool()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_maxpool (
              self,
              ir_node )  [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX maxpool layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX avgpool layer

Definition at line 568 of file AxfcTFIRTranslator.py.

8.14.3.14 _emit_aix_layer_softmax()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_layer_softmax (
              self,
              ir_node )  [private]
```

This method emits some tensorflow-specific information of the given IR node into the given AIX softmax layer object.

The information includes layer inputs, layer outputs, and so on.

Parameters

self	this object
ir_node	an IR node to be emitted

Returns

an output AIX softmax layer

Definition at line 733 of file AxfcTFIRTranslator.py.

8.14.3.15 _emit_aix_sampling_desc()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_sampling_desc (
              self,
              ir_node )  [private]
```

This method emits the AIX sampling description of the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor

Returns

an AIX sampling description

Definition at line 1378 of file AxfcTFIRTranslator.py.

8.14.3.16 _emit_aix_tensor_bias()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_bias (
              self,
              ir_node )  [private]
```

This method emits an AIX tensor of an bias type from the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor
is_default	indicates if default values are used to emit

Returns

an AIX tensor of an bias type

Definition at line 1051 of file AxfcTFIRTranslator.py.

8.14.3.17 _emit_aix_tensor_filter()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_filter (
              self,
              ir_node )  [private]
```

This method emits an AIX tensor of an filter type from the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor
is_default	indicates if default values are used to emit

Returns

an AIX tensor of an filter type

Definition at line 990 of file AxfcTFIRTranslator.py.

8.14.3.18 _emit_aix_tensor_input()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_input (
              self,
              ir_node )  [private]
```

This method emits an AIX tensor of an input type from the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor

Returns

an AIX tensor of an input type

Definition at line 928 of file AxfcTFIRTranslator.py.

8.14.3.19 _emit_aix_tensor_mean()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_mean (
              self,
              ir_node )  [private]
```

This method emits an AIX tensor of an mean type from the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor
is_default	indicates if default values are used to emit

Returns

an AIX tensor of an mean type

Definition at line 1139 of file AxfcTFIRTranslator.py.

8.14.3.20 _emit_aix_tensor_output()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_output (
              self,
              ir_node )  [private]
```

This method emits an AIX tensor of an output type from the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor
output_dims	output dimensions

Returns

an AIX tensor of an output type

Definition at line 1229 of file AxfcTFIRTranslator.py.

8.14.3.21 _emit_aix_tensor_scale()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_scale (
              self,
              ir_node )  [private]
```

This method emits an AIX tensor of an scale type from the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor
is_default	indicates if default values are used to emit

Returns

an AIX tensor of an scale type

Definition at line 1094 of file AxfcTFIRTranslator.py.

8.14.3.22 _emit_aix_tensor_variance()

```
def AxfcTFIRTranslator.AxfcTFIRTranslator._emit_aix_tensor_variance (
              self,
              ir_node )  [private]
```

This method emits an AIX tensor of an variance type from the given IR node.

Parameters

self	this object
ir_node	an IR node to be emitted as an AIX tensor
is_default	indicates if default values are used to emit

Returns

an AIX tensor of an variance type

Definition at line 1184 of file AxfcTFIRTranslator.py.

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

- /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRTranslator.py

Chapter 9

File Documentation

9.1 /home/youngsun/Project/SKT-AIX/Development/aixc/README.md File Reference

9.2 /home/youngsun/Project/SKT-AIX/Development/aixc/src/aixh_pb2.py File Reference

Namespaces

- aixh_pb2

Variables

- int aixh_pb2._b = sys.version_info[0] $<$ 3 and (lambda x: x) or (lambda x: x.encode('latin1'))
- aixh_pb2._sym_db = _symbol_database.Default()
- aixh_pb2.DESCRIPTOR
- aixh_pb2._AIXLAYER_AIXLAYERTYPE
- aixh_pb2._AIXLAYER_AIXACTIVATIONMODE
- aixh_pb2._AIXLAYER_AIXSAMPLINGMODE
- aixh_pb2._AIXLAYER_AIXDATATYPE
- aixh_pb2._AIXLAYER_AIXTENSORFORMAT
- aixh_pb2._AIXLAYER_AIXCONVOLUTIONDESC
- aixh_pb2._AIXLAYER_AIXSAMPLINGDESC
- aixh_pb2._AIXLAYER_AIXEWADDDESC
- aixh_pb2._AIXLAYER_AIXTENSOR
- aixh_pb2._AIXLAYER
- aixh_pb2._AIXGRAPH
- aixh_pb2.enum_type
- aixh_pb2.containing_type
- aixh_pb2.message_type
- aixh_pb2.AIXLayer
- aixh_pb2.AIXGraph
- aixh_pb2.has_options
- aixh_pb2._options

9.3 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcError.py File Reference

Classes

- class AxfcError.AxfcError

  AxfcError enum class.

Namespaces

- AxfcError

9.4 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcFrontendCompiler.py File Reference

Classes

- class AxfcFrontendCompiler.AxfcFrontendCompiler

  AxfcFrontendCompiler.

Namespaces

- AxfcFrontendCompiler

9.5 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcGraphWriter.py File Reference

Classes

- class AxfcGraphWriter.AxfcGraphWriter

  AxfcGraphWriter class.

Namespaces

- AxfcGraphWriter

9.6 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBlock.py File Reference

Classes

- class AxfcIRBlock.AxfcIRBlock

  AxfcIRBlock class.

Namespaces

- AxfcIRBlock

9.7 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRBuilder.py File Reference

Classes

- class AxfcIRBuilder.AxfcIRBuilder

  AxfcIRBuilder class.

Namespaces

- AxfcIRBuilder

9.8 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRGraph.py File Reference

Classes

- class AxfcIRGraph.AxfcIRGraph

  AxfcIRGraph class.

Namespaces

- AxfcIRGraph

9.9 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRNode.py File Reference

Classes

- class AxfcIRNode.AxfcIRNode

  AxfcIRNode.

Namespaces

- AxfcIRNode

9.10 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcIRTranslator.py File Reference

Classes

- class AxfcIRTranslator.AIXTensorType

  AIXInputType enum class.
- class AxfcIRTranslator.AxfcIRTranslator

Namespaces

- AxfcIRTranslator

9.11 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcLauncherWriter.py File Reference

Classes

- class AxfcLauncherWriter.AxfcLauncherWriter

  AxfcLauncherWriter class.

Namespaces

- AxfcLauncherWriter

9.12 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMachineDesc.py File Reference

Classes

- class AxfcMachineDesc.AxfcMachineDesc

  AxfcMachineDesc class.
- class AxfcMachineDesc.AxfcMachineDesc.AIXLayerInfo

  AIXLayerInfo inner class.

Namespaces

- AxfcMachineDesc

9.13 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcMain.py File Reference

Namespaces

- AxfcMain

Functions

- def AxfcMain.__main (vargs)

  main function

Variables

- AxfcMain.parser
- AxfcMain.metavar
- AxfcMain.type
- AxfcMain.str
- AxfcMain.required
- AxfcMain.help
- AxfcMain.args = parser.parse_args()

9.14 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRBuilder.py File Reference

Classes

- class AxfcTFIRBuilder.AxfcTFIRBuilder

  AxfcTFIRBuilder class.

Namespaces

- AxfcTFIRBuilder

9.15 /home/youngsun/Project/SKT-AIX/Development/aixc/src/AxfcTFIRTranslator.py File Reference

Classes

- class AxfcTFIRTranslator.AxfcTFIRTranslator

  AxfcTFIRTranslator class.

Namespaces

- AxfcTFIRTranslator

Variables

- dictionary AxfcTFIRTranslator.aix_data_type_tbl

  Global tables for AIXDataType and AIXTensorFormat.
- dictionary AxfcTFIRTranslator.aix_tensor_format_tbl

  AIXTensorFormat table.

9.16 /home/youngsun/Project/SKT-AIX/Development/aixc/tst/aix_tf.md File Reference

Index