**Toolchain Spec**

**Frontend**

**[parser]**

Input: Transformer Model (.onnx)

Output: network raw info (.json)

**[optimizer]**

Input: network raw info (.json), library interface (configuration)

Output: optimized network info (.json)

**Backend**

**[builder]**

Input: network info(internal), library interface (generation)

Output: Module--RTL hardware code(VHDL/Verilog HDL)

**[constructor]**

Input: network info(interconnection), Module--RTL hardware code

Output: Accelerator--RTL hardware code(VHDL/Verilog HDL)

**Template Library**

**[configuration lib]**

Network Structure Representation of Existing Operators

Interface: Operators (topology+attribute) configure (.ini)

**[generation lib]**

Auto-Generation of (high-performance) Operators

Interface: Generators Parameters configure (.ini)

Future Work

**[func template lib] --** Data Movement (PCIE/HBM)

Scheduler/Memory Mapping/DSE…

**Demo 1.0**

**Functional Description:**

Frontend:

1. parse onnx model and get structured information
2. graph optimization and granularity match with configuration

Backend:

1. build operators module with library and operator attribute info
2. construct hardware accelerator with network info and operator codes

Library:

1. configuration library (structure + attribute description)
2. generation library (auto generators of different operators)

**operator requirement: gelu, layernorm, mm, madd, transpose, split, merge**

**Interface Specitication:**

**[network info]: (refer to dag\_match.json)**

json文件类型，网络模型的算子列表，单个算子信息存储格式如下：

-op\_id: 算子编号

-op\_type: 算子类型

-attribute: 算子性质（所需配置参数）

-name: 参数名

-type: 参数类型

-ints/…: 具体参数

-topo: 拓扑结构信息

-name: 算子名

-input: 输入线名

-output: 输出线名

-pre: 前层算子信息(算子编号，算子类型)

-next: 后层算子信息(算子编号，算子类型)

**[library interface]:**

Configuration Interface: **(refer to gelu.ini/layernorm.ini)**

Topology Configure:

-opset: 结构包含的细粒度算子集（命名规则：算子类型\_标号）

-flowset: 结构边集（命名规则：算子-算子）

(-matrix: 按序邻接矩阵表达)

-sign: 关键标志算子

Attribute Configure:

-attribute: 算子参数集（命名规则：参数名-类型）

Generation Interface: (refer to genlib.ini)

规范：生成器名 = 需求参数列表

**coding (refer to nnoptimizer.py/utils.py)**

面向对象编程，通用函数单独成文件，类相关函数实现为内置方法；

文件、函数组织划分明确，避免代码过长，有一定注释。