LLM for NTT

1.文件构成

NTT模块由多个verilog文件构成，其中NTT.v是顶层文件，NTT_tb.v为仿真测试文件：

- verilog/AddressGenerator.v
- verilog/BRAM.v
- verilog/ButterFly.v
- verilog/defines.v
- verilog/intMult.v
- verilog/ModMult.v
- verilog/ModRed.v
- verilog/ShiftReg.v

2.仿真测试

- **NTT测试**：执行下面的命令将生成NTT测试数据于test/文件夹中

```
make ntt-gen
```

- **INTT测试**：执行下面的命令将生成INTT测试数据于test/文件夹中

```
make intt-gen
```

3.LLM目标

3.1 动态参数

A.Params

> **Note**
>
> Baseline源码的参数配置：

参数名称	值	释义
N	1024	多项式维度
K	16	数据位宽=$log_2(Q)$
P	8	PE数目
Pr	8	同一Stage的计算并行度
Pc	1	跨Stage的计算并行度

B.生成目标

- 多项式维度N：生成不同N下的代码$(N \in \{2^8, \ldots, 2^{16}\})$
- 数据位宽K：生成不同K下的代码$(K \in [8, 128])$
- 运算单元数P：生成不同P下的代码$(P \in 1, 2, 4, 8, 16)$

C.高阶目标

在$P = 8$时，实现对不同并行度的支持：

- $(Pr, Pc) = (8, 1)$（Baseline）
- $(Pr, Pc) = (4, 2)$
- $(Pr, Pc) = (2, 4)$
- $(Pr, Pc) = (1, 8)$

> **Note**
>
> 可以尝试生成全参数化的代码，看效果如何！

3.2 功能优化

A.运算单元优化

- 乘法器：阵列乘法器、Wallce、Booth等；
- 模乘：Montgomery、Barrett；

B.位宽动态支持

- 固定运算器位宽，如8/16bits，如何动态支持其它位宽；

C.访存优化

- 如何设计无Bank冲突访存，并减少数据交互等；

3.3 功能扩展

- 生成仅支持INTT的代码；
- 生成支持NTT+INTT的代码；
- 生成支持NTT→点乘→INTT的代码