基于FPGA的片上网络自适应路由算法的设计与实现

学生：江岑倩

指导教师：蔡旻

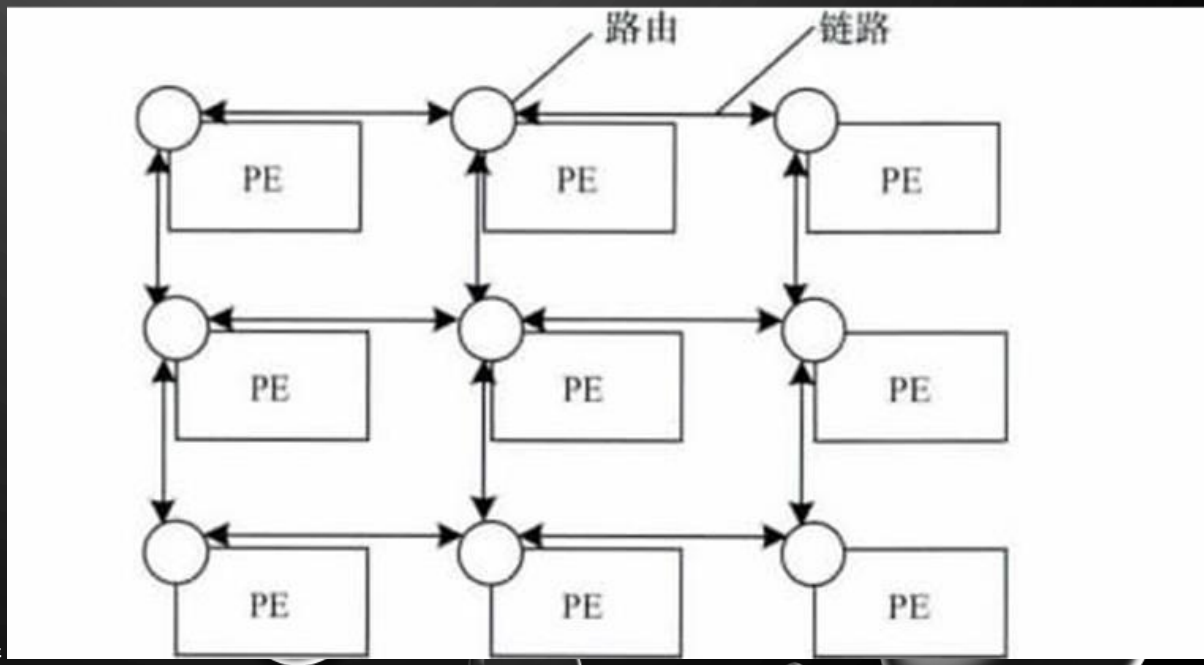
本科生毕业设计答辩

北京工业大学 计算机学院

目录

背景：片上网络

- 多核芯片上集成的核心数越来越多
- 片上网络
  - 比传统的总线互连可扩展性更好、功耗更低



背景：路由算法

- 每个核心上运行的应用其数据访问特征不一样
  - 数据密集型 vs. 计算密集型
- 自适应路由算法
  - 比确定性路由算法更能适应不均匀的片上网络流量分布

相关工作：自适应路由算法

路由算法：从所有输出端口列表中选出可用的输出端口列表

- Odd Even Turn Model（奇偶转弯）路由算法

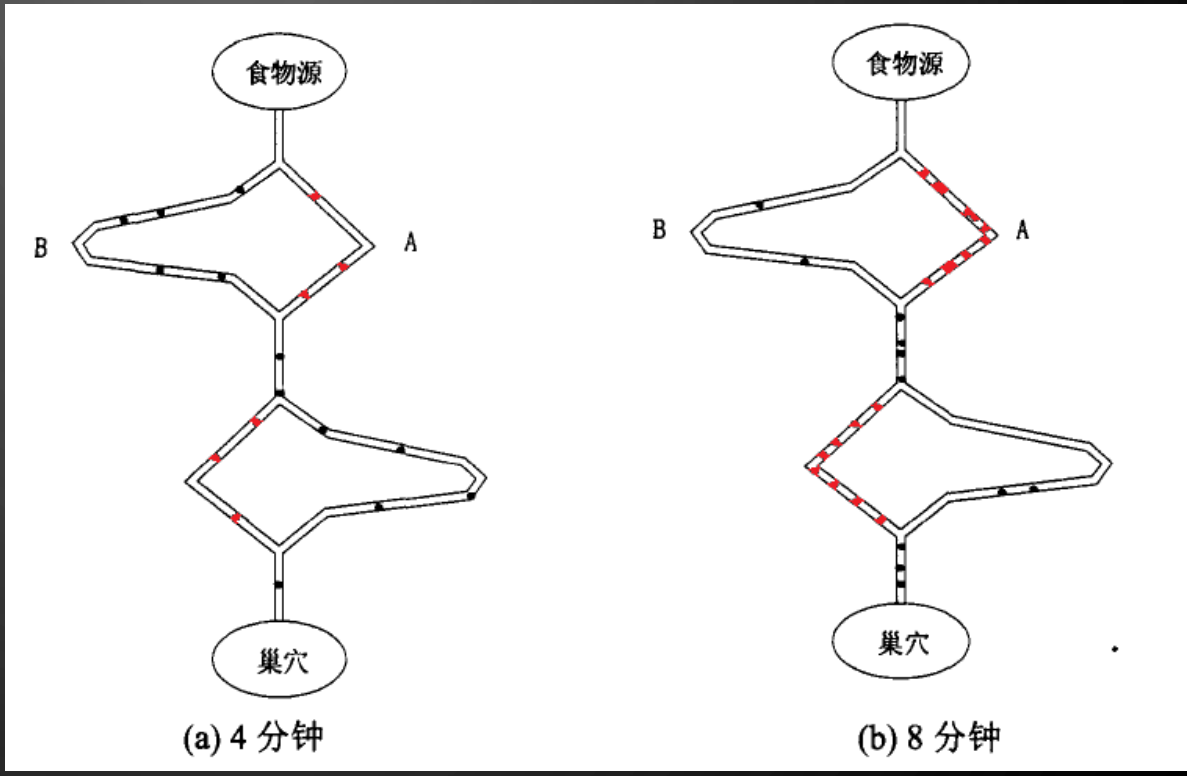
选择算法：从可用的输出端口列表中选出最合适的输出端口

- Buffer Level （缓冲水平）选择算法
- Neighbor on Path （NoP，路径邻居）选择算法
- 上述两种算法都未考虑网络状态的历史信息

相关工作：蚁群优化

蚁群优化（Ant Colony Optimization, ACO）

- 源自自然界中的蚁群觅食行为
- 模拟基于信息素的信息交换与反馈机制
- 算法呈现出并行与分布特性



非对称“双桥”实验

相关工作： ANTNET 算法

- 基于蚁群优化的自适应路由选择算法
- 基本思想：结合了基于信息素的路由表设计以及基于缓冲水平的本地模型
- 选择概率计算

$$P'(j,d)=\frac{P(j,d)+\alpha L_i}{1+\alpha(|N_k|-1)}$$

P（j，d）为表中（j，d）位置的信息素浓度值；Li代表了邻居结点的缓冲水平；N为当前结点的邻居结点数；alpha为路由选择因子；P’（j，d）即为发往目的地结点d的消息包、会从j输出端口输出概率。

- 信息素浓度的更新

$$P(i)=P(i)+r\times(1-P(i))$$

P(i)为信息素表浓度值，r为激励因子。

，结点在蚂蚁包的memory中时

$$P(i)=P(i)-r\times P(i)$$

，结点不在蚂蚁包的memory中时

我的工作：基于蚁群优化的自适应路由选择算法的FPGA设计与实现

基本思想

- 用信息素来表达网络状态的历史信息
- 根据网络状态的历史信息和当前信息进行路由选择

具体实现

- 新增两种消息包：正向（forward）蚂蚁包和逆向（backward）蚂蚁包
- 采用了基于信息素的路由表结构
- 采用了结合了信息素浓度和邻居结点缓冲水平的路由选择机制
- 正向蚂蚁包和普通包采用路由表进行路由，正向蚂蚁包到达目的结点后转换为逆向蚂蚁包，并在原路返回的过程中更新所经结点信息表中的浓度

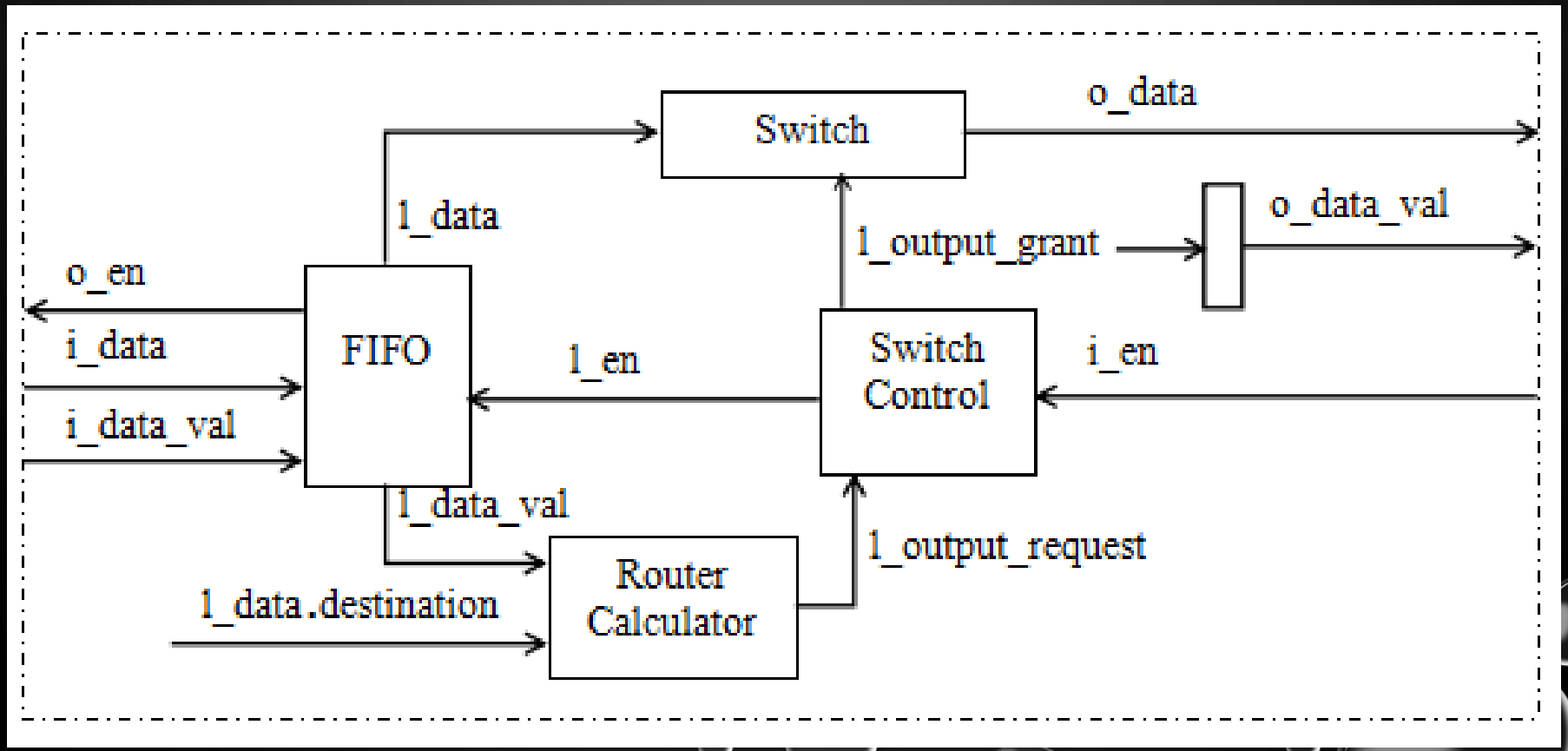
消息包结构

类型	名称	描述
logic	ant	标记消息包的类型。 1: ant packet; 0: normal packet
logic [$clog2(`X_NODES)-1:0]	src_x,src_y	该消息的源结点坐标
logic [$clog2(`X_NODES)-1:0]	dest_x,dest_y	该消息的目的地结点坐标
logic	backward	标记蚂蚁包的类型（仅为蚂蚁包时有效）。 1: backward packet; 0: forward packet
logic [$clog2(`NODES)-1:0]	num_memories	memory中记录的路径的结点数目 （仅为蚂蚁包时有效）
logic [0:`NODES-1][$clog2(`X_NODES)-1:0]	memory_x,memory_y	蚂蚁包走过的路径队列 （仅为蚂蚁包时有效）

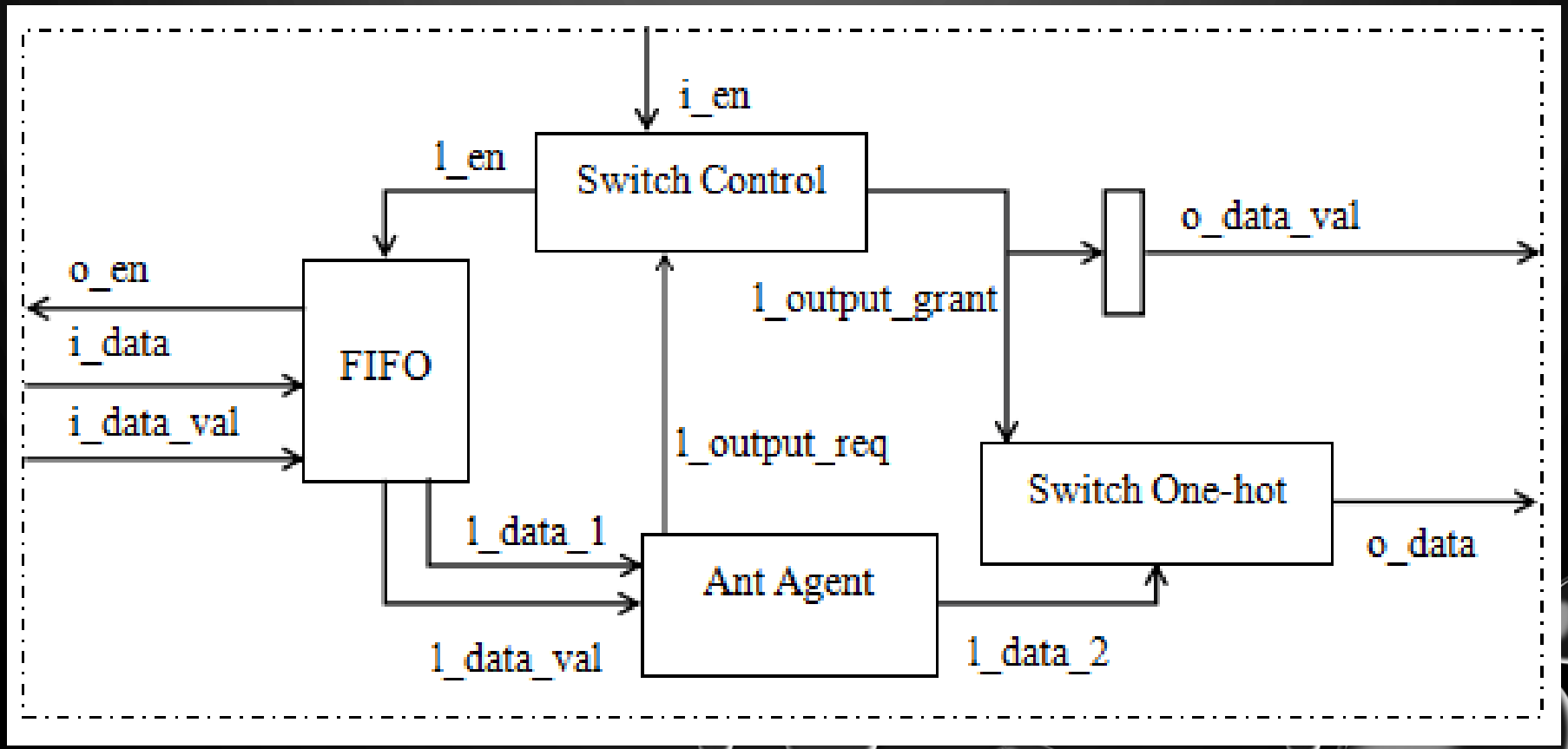
信息素表结构

Destination\Neighbor	北	东	南	西
0				
1				
:				
d	ph[d][北]	ph[d][东]	ph[d][南]	ph[d][西]
:				
结点数-2				
结点数-1				

路由器架构（原）


Switch
o_data
o_data_val
l_data
l_output_grant
o_en
i_data
i_data_val
FIFO
l_en
Switch
Control
i_en
l_data_val
l_output_request
Router
Calculator
l_data.destination

路由器架构（现）


i_en
l_en
Switch Control
o_data_val
o_en
l_output_grant
i_data
FIFO
i_data_val
l_output_req
Switch One-hot
l_data_1
o_data
Ant Agent
l_data_val
l_data_2

实验环境

- 开发环境：ALTERA QUARTUS PRIME 15.1
- 硬件描述语言：SYSTEMVERILOG
- 开发流程：FPGA设计收入->仿真->输出实验结果->实验结果统计
  - 采用TESTBENCH对单个结点和整个网络进行测试
  - 采用$DISPLAY语句输出片上网络的性能指标

实验配置

- 网格（MESH）拓扑结构，4X4=16个结点；缓冲长度：4 PACKETS
- 三种SYNTHETIC TRAFFICS：UNIFORM，TRANSPOSE和HOTSPOT
- 包注入率（PACKET INJECTION RATE）：0.01 ~ 0.6 PACKETS/NODE/CYCLE
- MAX CYCLES
  - WARMUP: 1000, MEASURE: 10000, NO MEASURE: 3000, DRAIN: 3000
- ACO路由选择算法
  - 路由选择因子ALPHA=0；激励因子RF=1

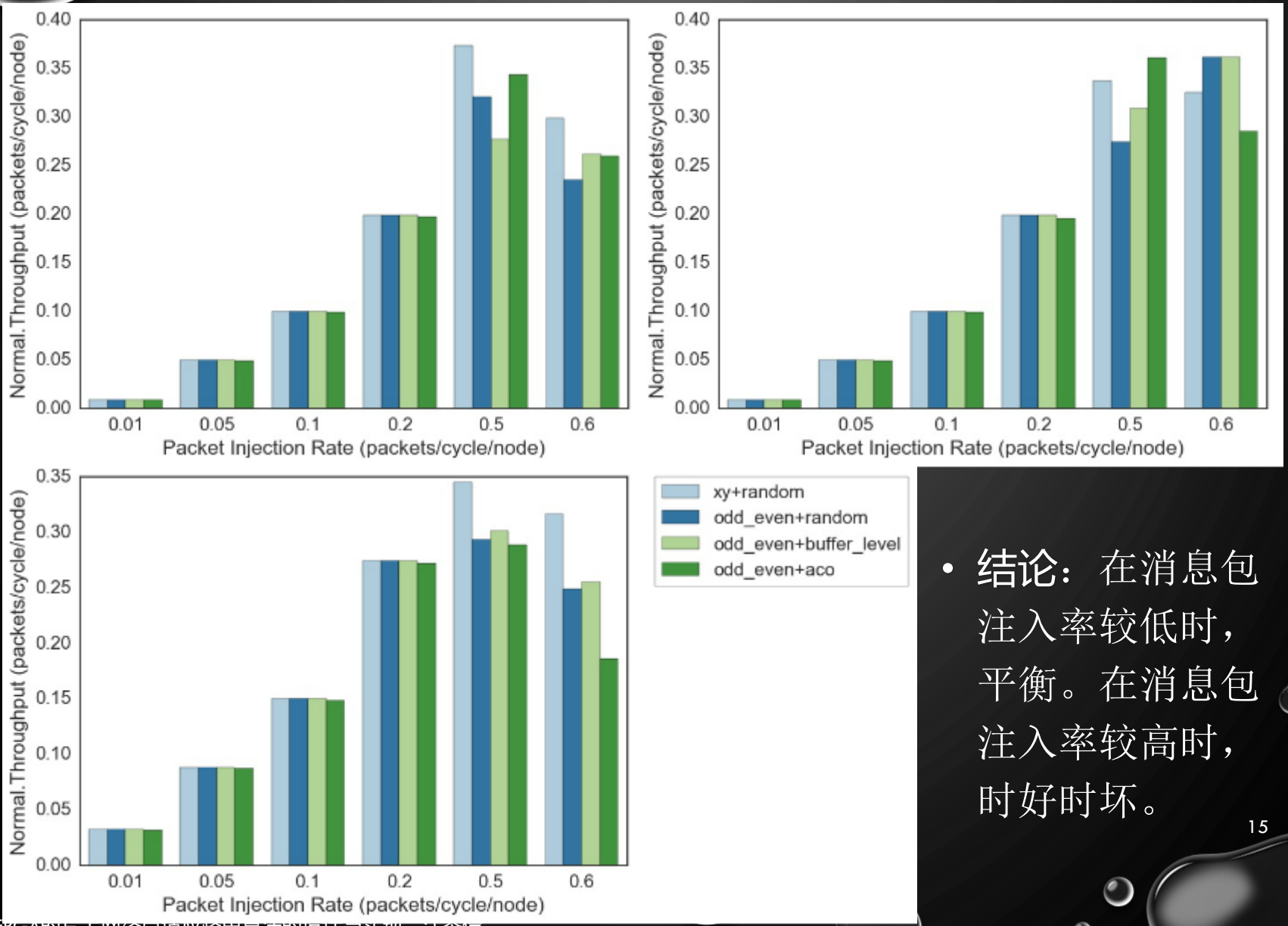
主要性能指标

- 吞吐率（THROUGHPUT）= #PACKETS TRANSMITTED/NODE/CYCLE
- 平均包延时（AVERAGE PACKET DELAY）

消息包结构体中用来帮助测试的字段

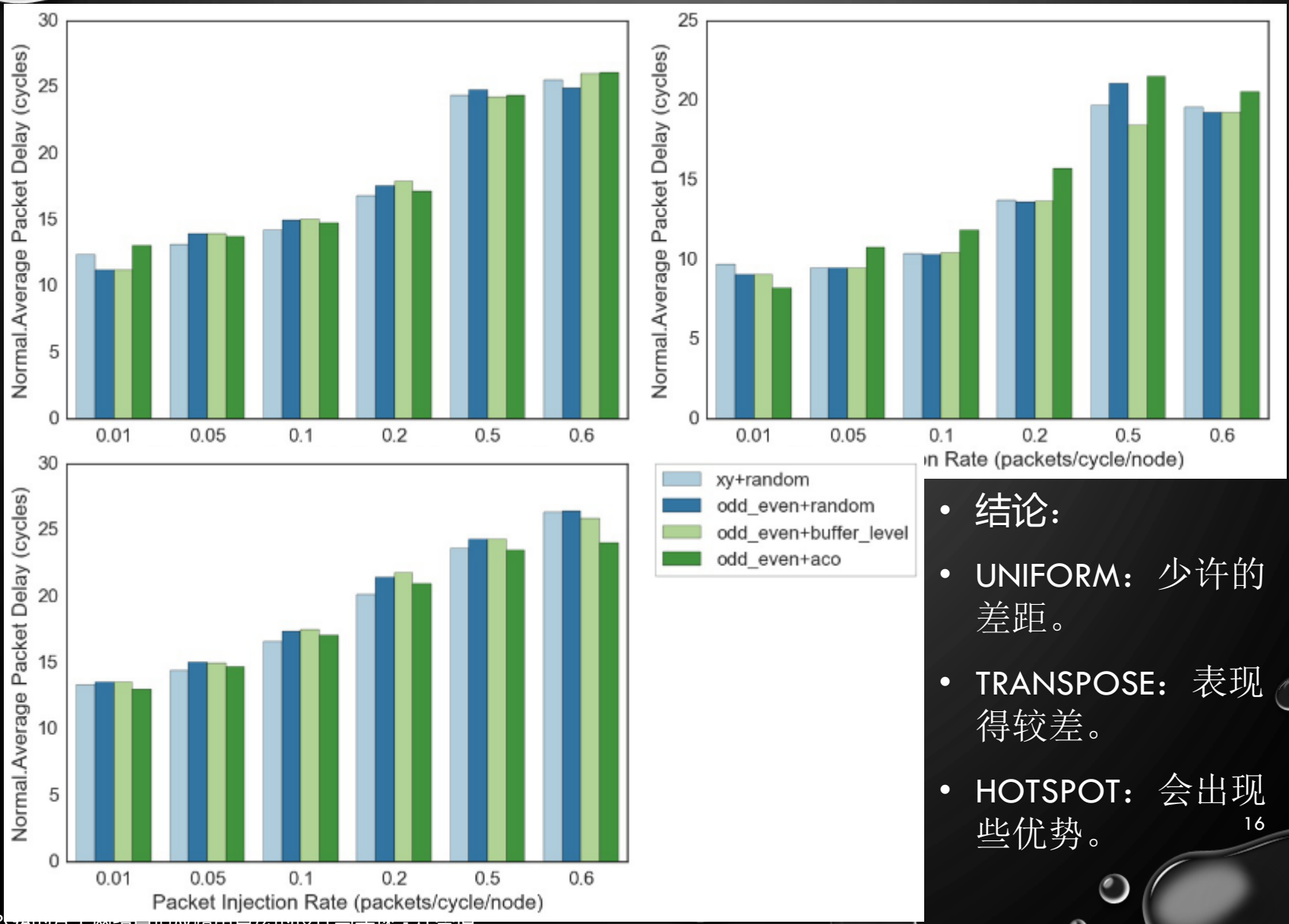
类型	名称	描述
Logic [7:0]	id	仿真时记录消息包的id号
logic	measure	仿真时记录测试状态
logic [`TIME_STAMP_SIZE-1:0]	timestamp	仿真时记录数据包进入网络时的时间

实验结果分析：吞吐率



- **结论：**在消息包注入率较低时，平衡。在消息包注入率较高时，时好时坏。

实验结果分析：平均包时延



- 结论：
- UNIFORM：少许的差距。
- TRANSPOSE：表现得较差。
- HOTSPOT：会出现些优势。

结束语

工作总结

- 前期工作
  - NoC、 ACO和FPGA文献阅读
  - 基于MOJO开发板的FPGA编程练习
- 核心工作
  - Odd Even路由算法的FPGA设计、实现与仿真
  - ACO选择算法的FPGA设计与实现与仿真

不足与可改进之处

- 扩大结点数、完善路由器架构、优化ACO选择算法

谢谢！