



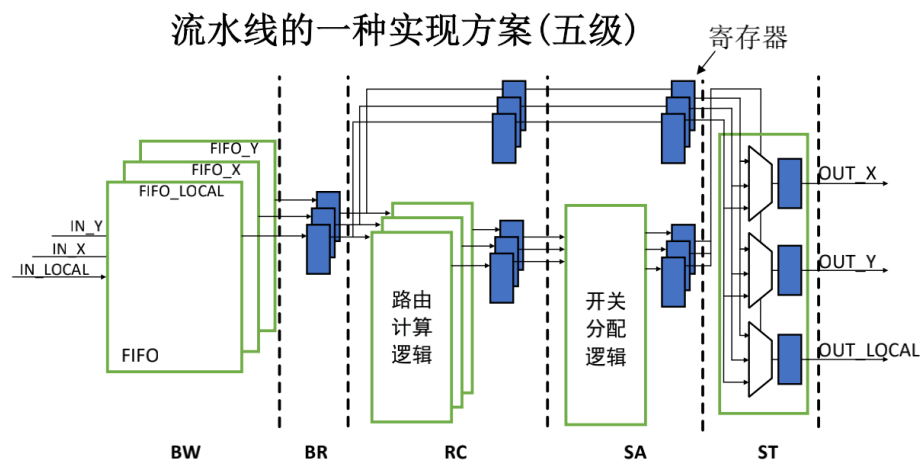
NoC Design Technique Document

Target：设计一个2*2 的mesh结构，每个mesh包括一个 router，NI（不用做），IP（不用做），4个mesh相互连接，实现【存储+传送】网络模型。

Router Instruction

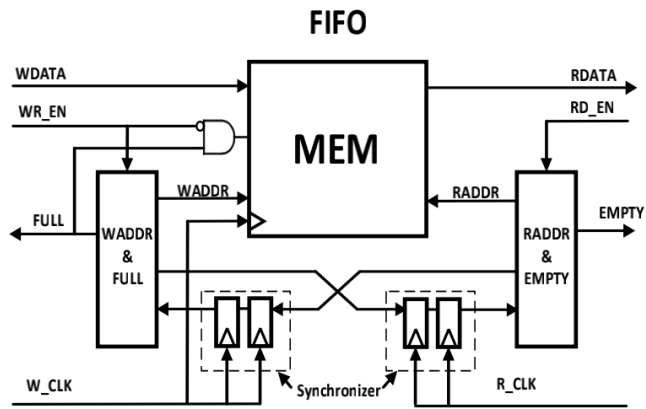
include 3 modules:

- FIFO (used for BUFFER)
- 路由计算
- 开关分配



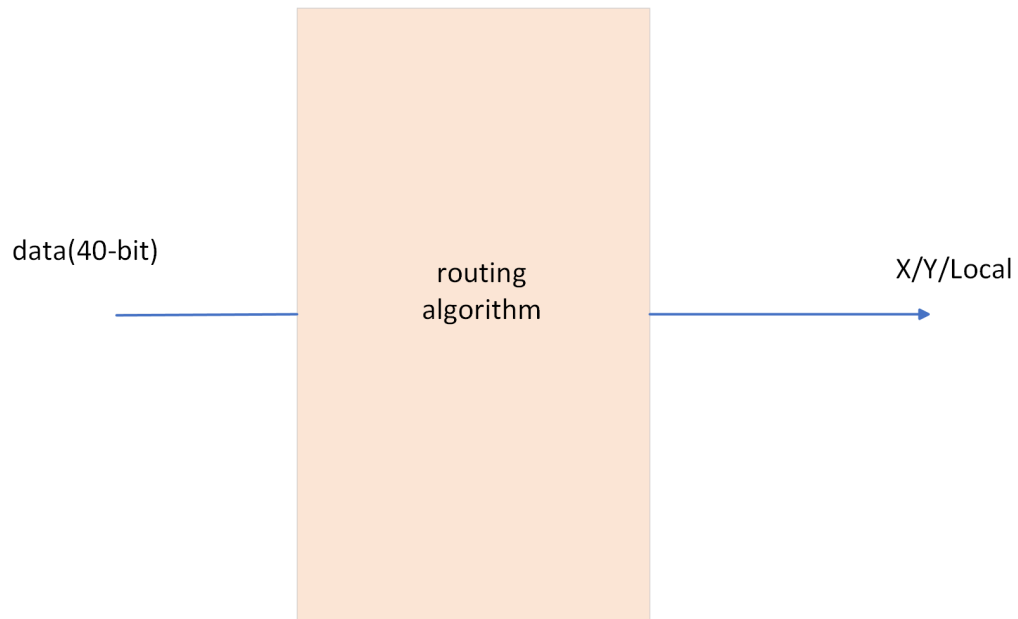
FIFO

- 作为缓存区，实现存储功能
- 一种可能的实现方式如下：



- MEM 是一个 FIFO, $40\text{-bit} \times \text{Length}$ 宽长比 (Length是一个需要trade-off的参数)
- 如何判断full, empty :
 - 当 $WADDR == RADDR \rightarrow \text{empty}$
 - 当 $WADDR + 1 == RADDR \rightarrow \text{full}$
 - 具体实现方法可以用 $\log_2 LENGTH + 1$ 个bit存储指针, 可参考《数字逻辑设计》slides/lab3/page6

Routing Algorithm



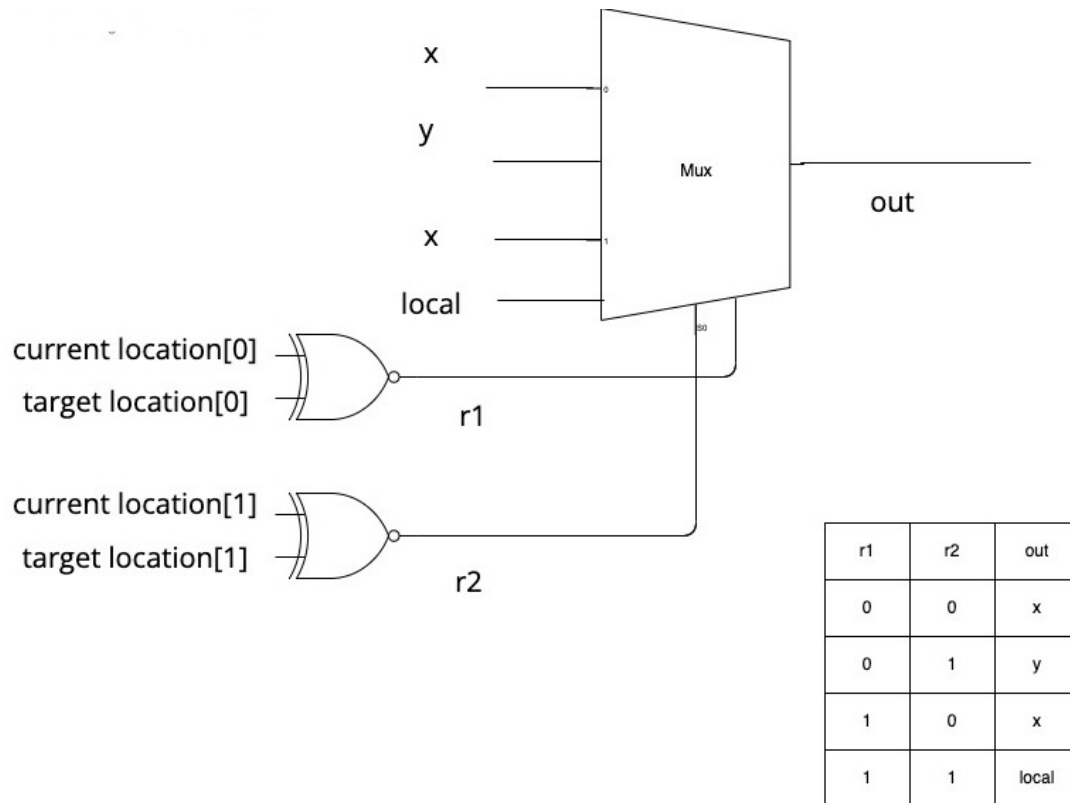
- 选择data从source → target 的路径

- 算法：

- 对X/Y/Local 进行编码

X	01
Y	10
Local	11

- 将data所在当前路由器的编码与目标路由的编码按位XNOR，得到的2-bit作为4输入mux的控制端：



- current location 可以从上一个FIFO中得到信息，初始化信息可以从data中获得
- target location 可以从data中获得
- out 即代表下一个data传输方向（X/Y/Local）

Switch Allocation

- 控制流量；同时对于冲突的情况进行仲裁，避免数据死锁

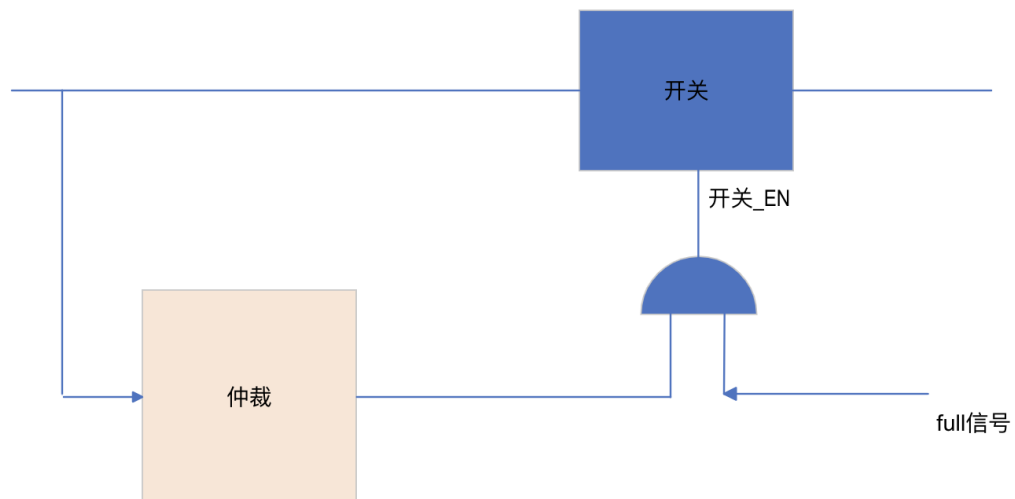
缓冲区	转发端口	仲裁结果
X	Y	成功
Y	LOCAL	成功
LOCAL	X	成功

无转发端口冲突

缓冲区	转发端口	仲裁结果
X	LOCAL	成功
Y	LOCAL	失败
LOCAL	X	成功

转发端口冲突

- 方法（轮询）



- 默认优先级是 $X > Y > Local$
- 假设某一时刻X, Y, Local缓冲区和转发端口已经准备完毕, 两两对下一步转发端口进行冲突检查 (XNOR)
- iff
 - 三组的XNOR结果都是0, 无冲突

- 仲裁结果为‘成功’，传输数据
- iff
 - （至多有一组）XNOR结果有1，发生冲突
 - 较高优先级的仲裁结果为‘成功’，传输数据
 - 较低优先级的仲裁结果为‘失败’，关闭开关
 - 调整被关闭开关的下一次优先级
- full信号也能控制开关的关闭，暂停这一次的数据传输
- 仲裁结果作为feedback链接回上一个fifo的en信号位。