

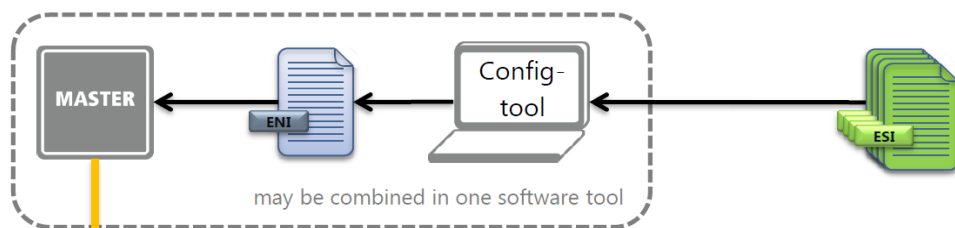
TwinCAT控制系统中 EtherCAT 的结构和拓扑



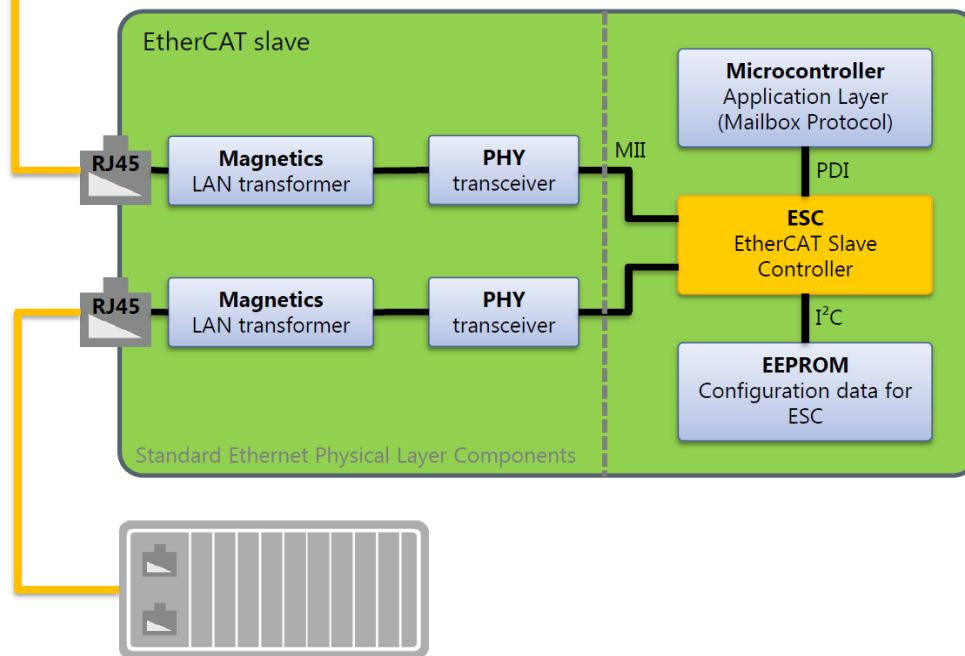
EtherCAT 网络的结构

一个EtherCAT 网络包括：

- 一个 Master 设备：

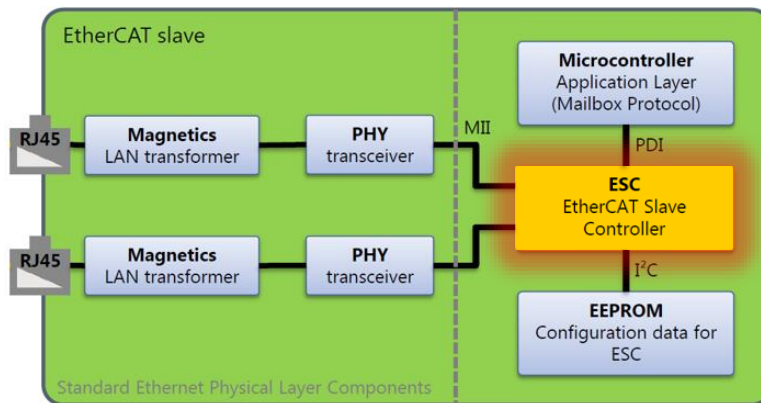


- 多个slave 设备：
(最多. 65535)



ESC 的内部结构

EtherCAT Slave Controller (ESC) 是集成在每一个EtherCAT从站中的芯片组件，可以支持绝大部分时序要求严格的功能：



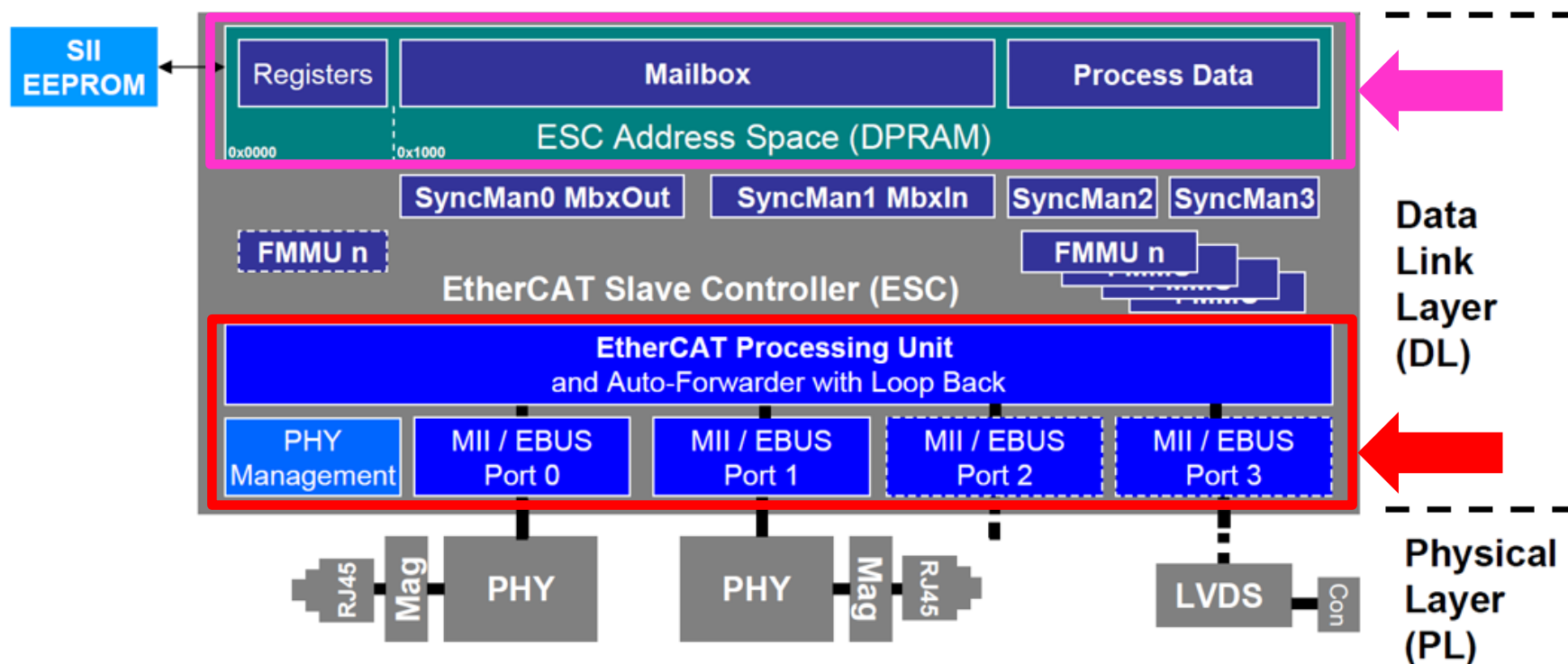
- ESC 实现EtherCAT网络数据帧的硬件路由，无须软件寻址，不用MAC-Address 或者IP地址
- 在数据帧传输过程中处理数据，不用存储和转发数据帧，经过的延迟时间可提前预计
- 通过硬件实现EtherCAT网络中各个设备的本地时间（ Local times ）的精确同步

所有 ESCs 的基本功能完全相同，不论它们来自哪个厂商



ESC 的内部结构

所有ESC都是由同样构件组成的系统，包含 2 - 4个端口和1个用于数据交换的双端口存储器。

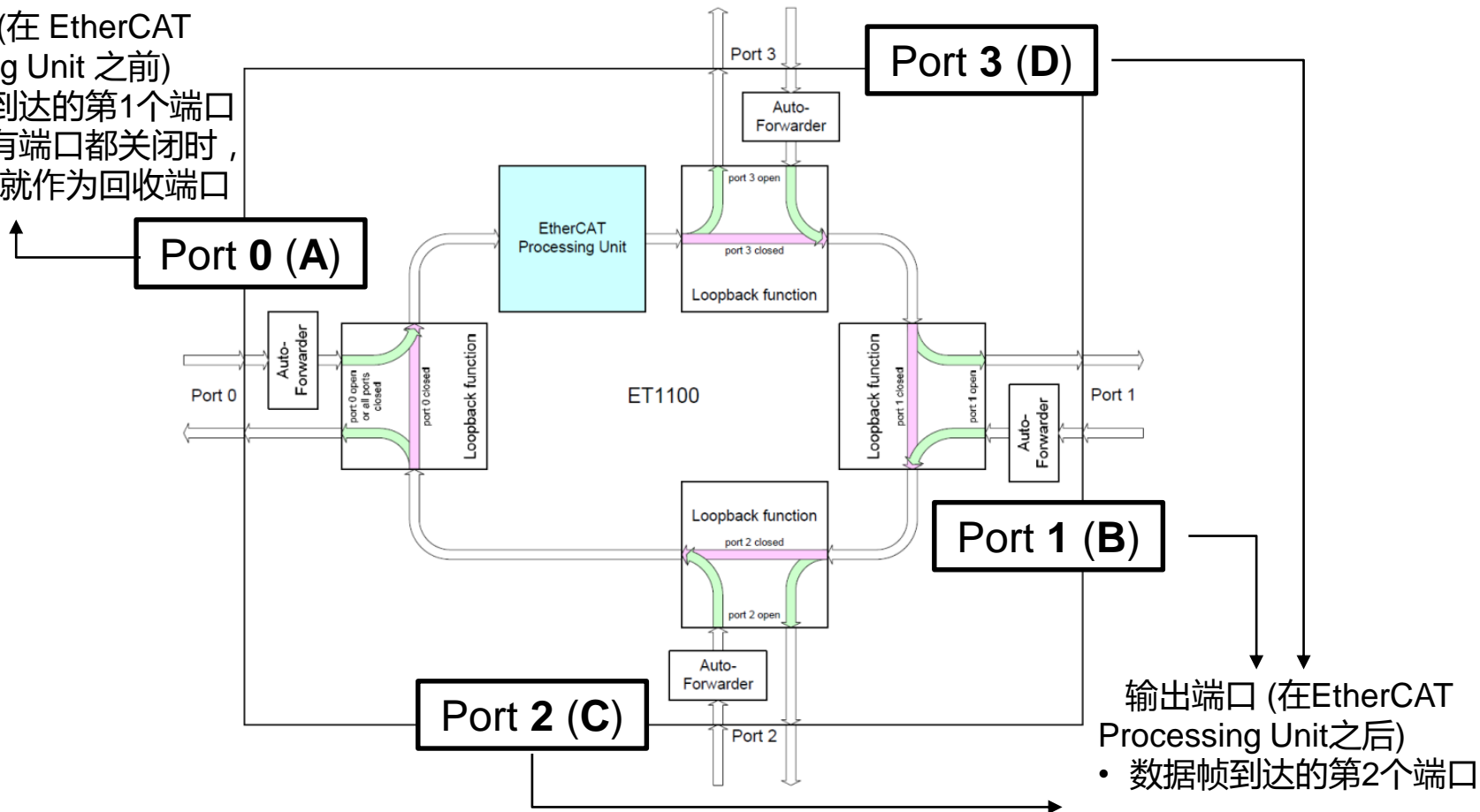


Ports (端口)

EtherCAT Slave Controllers (ESC) 支持 2 至 4 个端口 :

输入端口 (在 EtherCAT Processing Unit 之前)

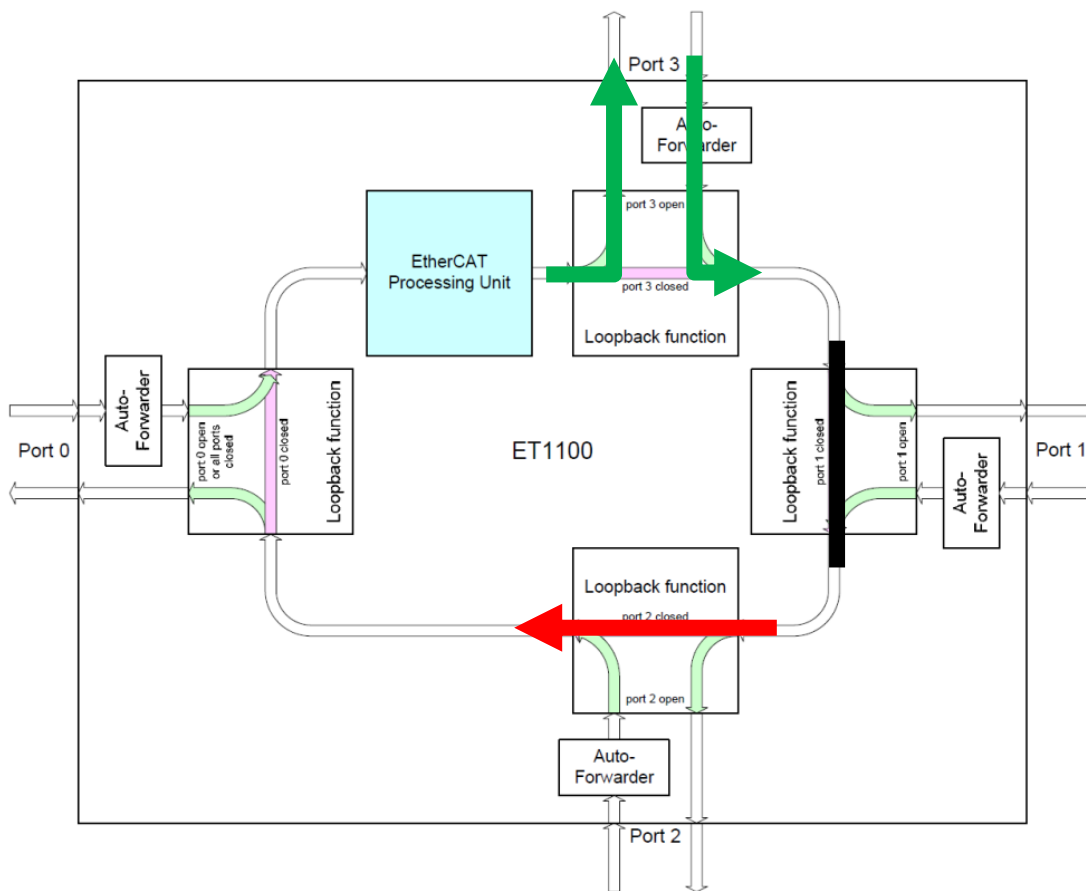
- 数据帧到达的第1个端口
- 其它所有端口都关闭时, 该端口就作为回收端口



Ports (端口)

每个端口总是处于以下两种状态之一：

- **Port open**：数据帧从这个端口出去，然后从这个端口回来
- **Port closed**：数据帧从内部传到本站的下一个端口



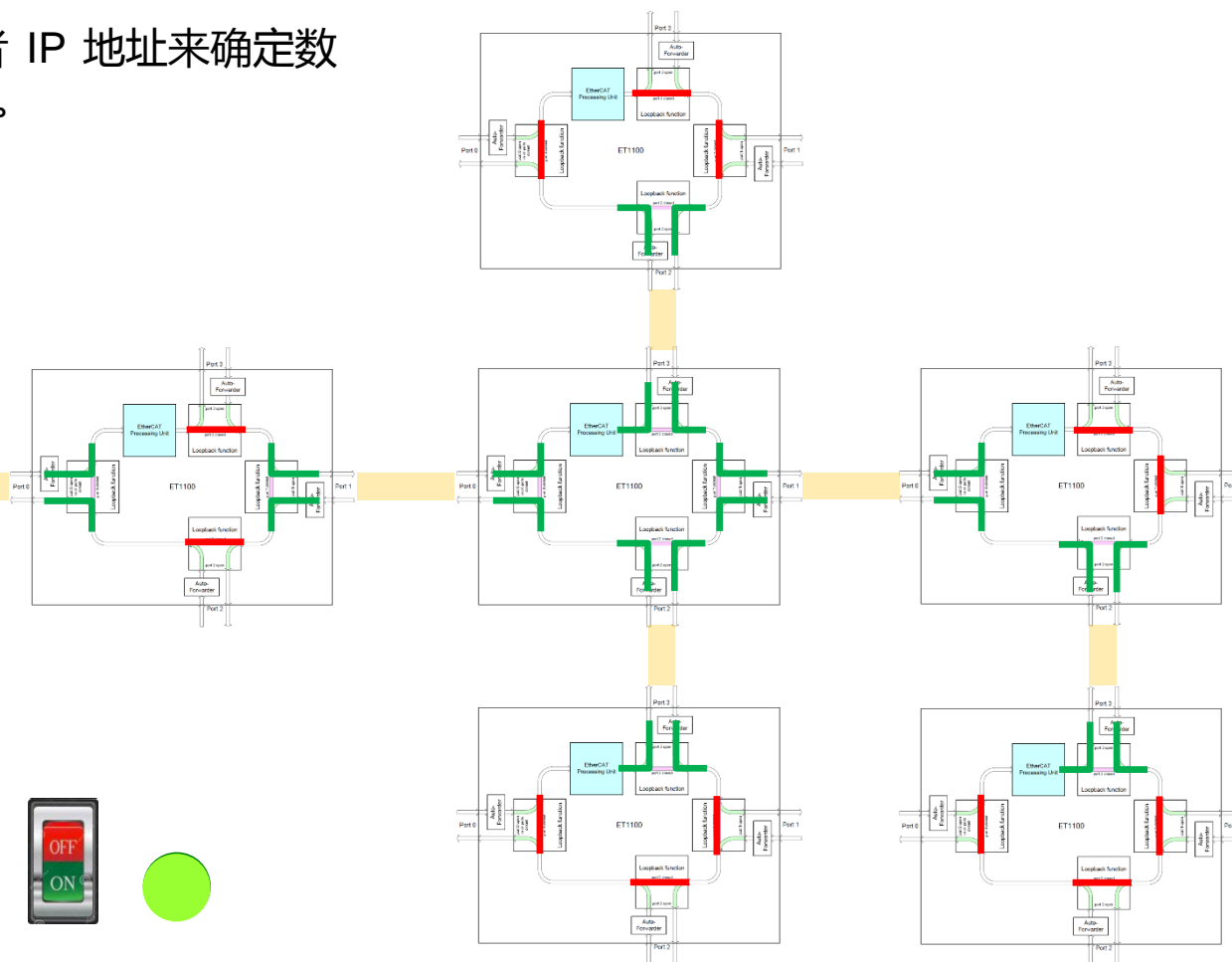
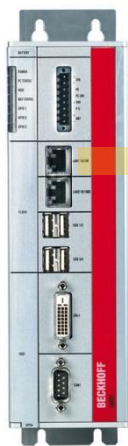
端口的默认设置是**自动**模式：
如果有物理连接，端口自动打开，如果没有检测到物理连接就自动关闭。



Ports (端口)

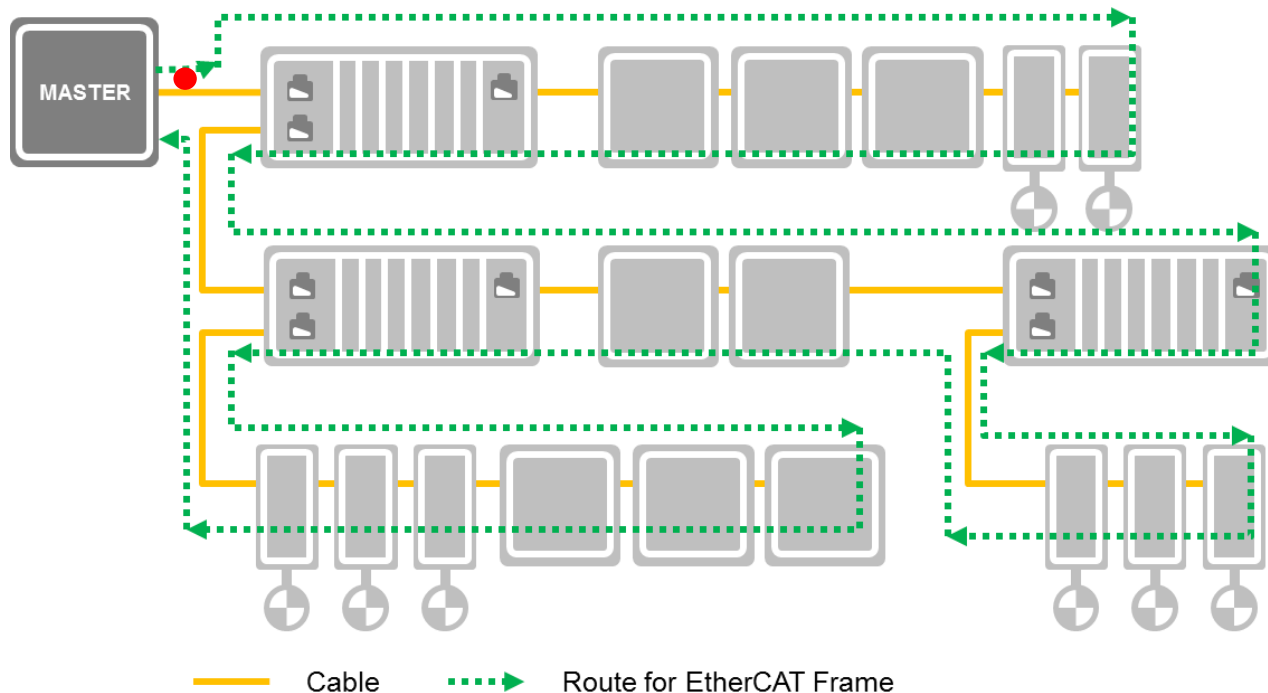
基于默认的自动模式，EtherCAT网络在上电时就会**自动**构建自身的架构：

不需要 MAC-ID 或者 IP 地址来确定数据帧在网络中的走向。



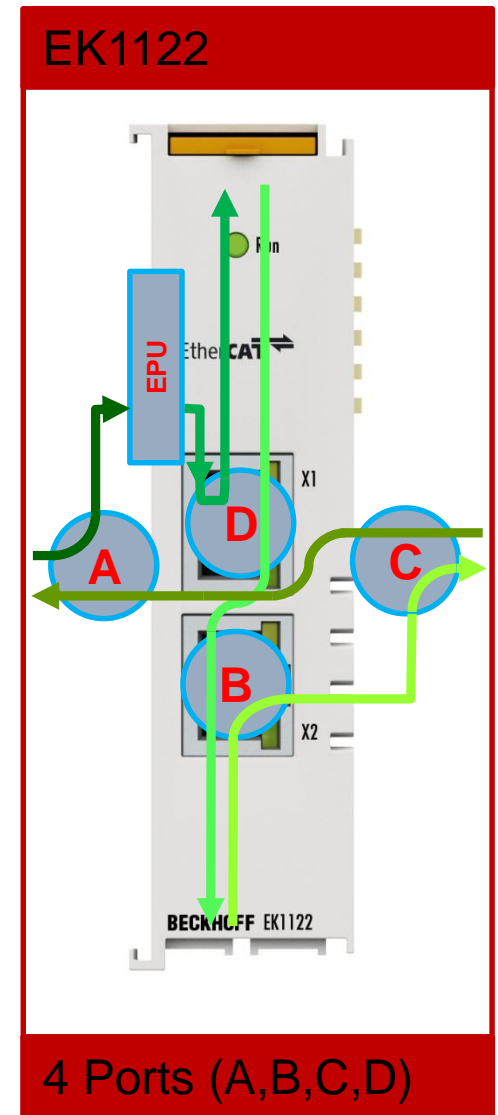
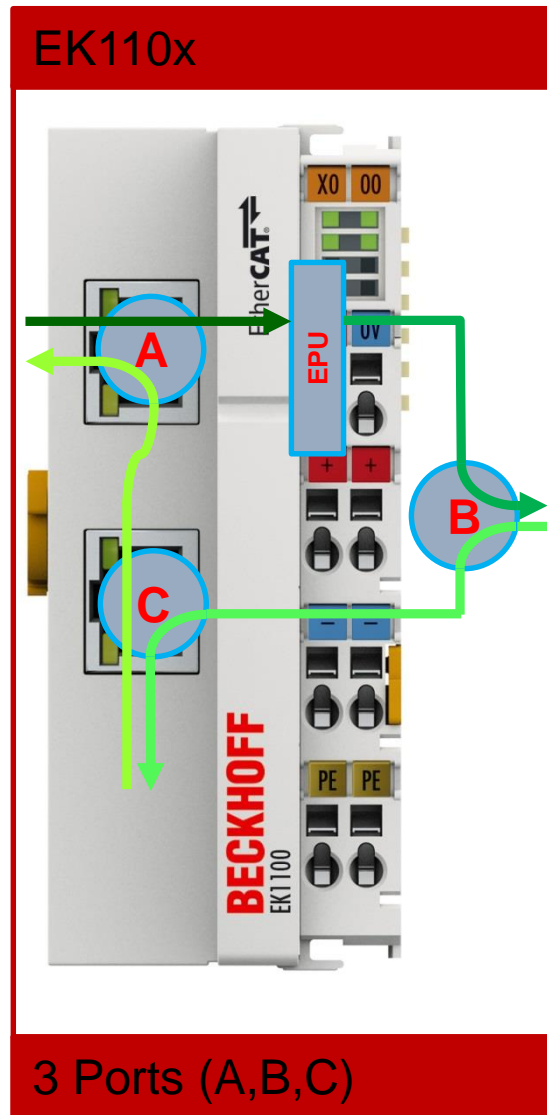
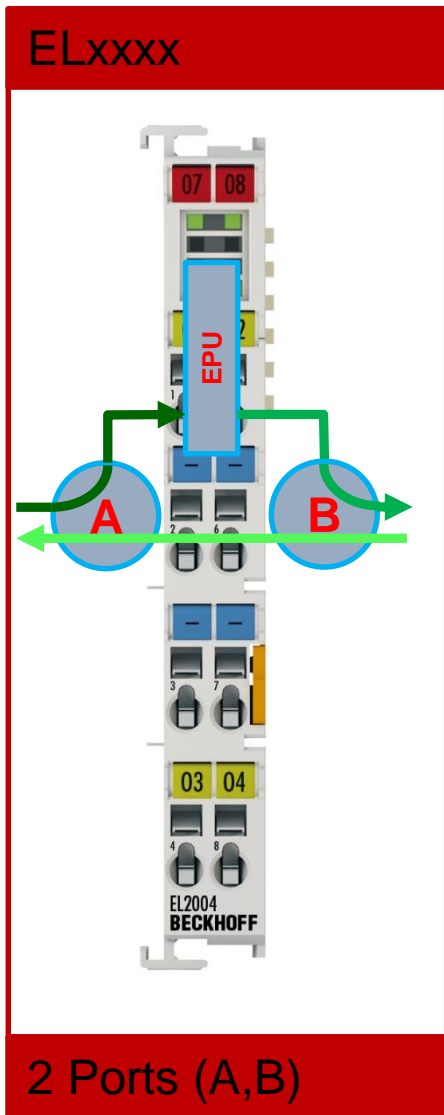
EtherCAT 网络中的数据帧走向

所有数据帧在网络中以一种“逻辑闭环”的方式传播，与网络的硬件拓扑无关，无论它是链式、菊花链、星形还是树形拓扑。



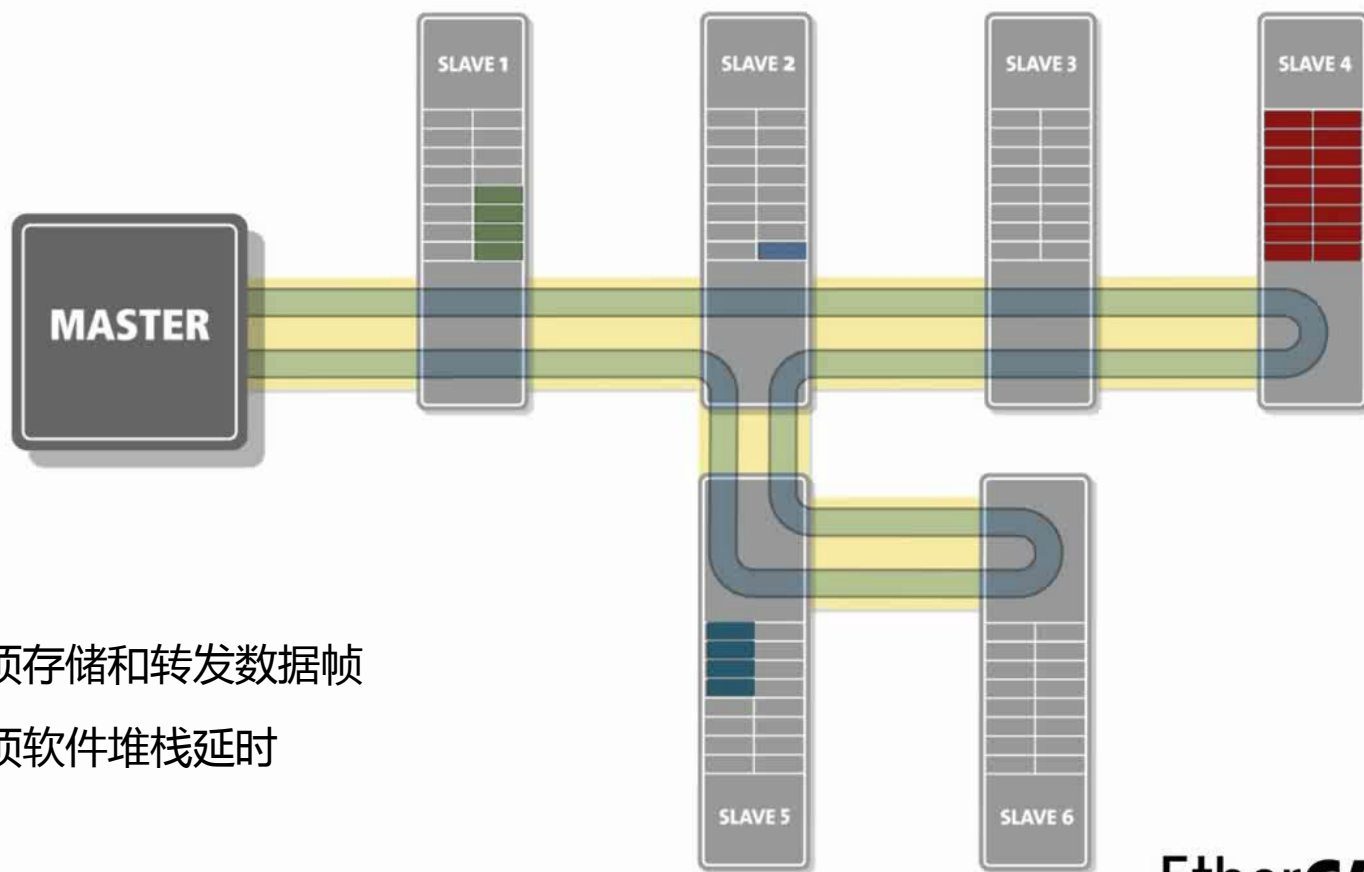
- 所有数据帧都由Master发出，以**事前严格定义**的顺序，依次经过网络上的所有从站，走过一个**完整的闭环**后回到Master。
- 所有数据帧通过从站中的 EtherCAT Processing Unit (EtherCAT处理单元) 只有 **1 次**

Beckhoff 拓扑组件中EtherCAT数据帧的走向



EtherCAT Processing Unit (EtherCAT处理单元)

EtherCAT Processing Unit 总是位于 Port 0 之后其它端口之前，并在数据帧传输的过程中提取和插入数据：



- 无须存储和转发数据帧
- 无须软件堆栈延时

EtherCAT®

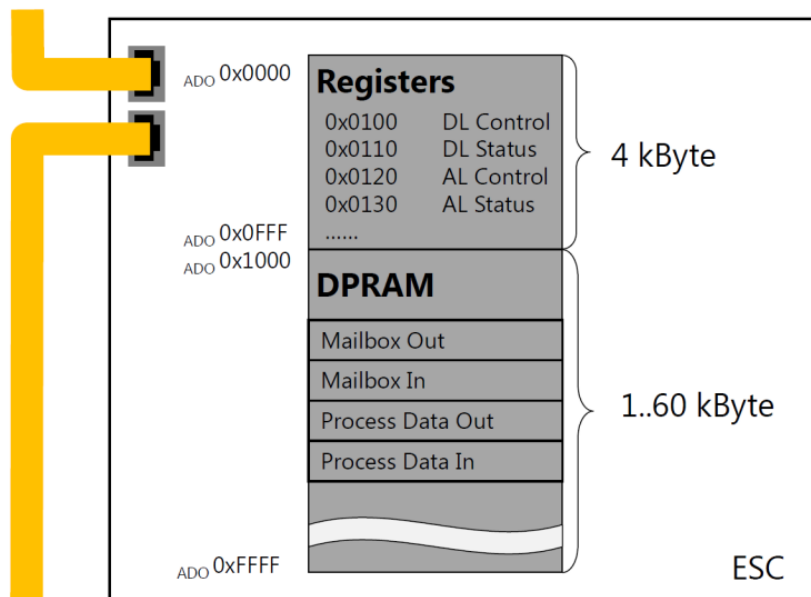
Dual Ported RAM (双端口存储器)

访问 ESC 的 **Dual-Ported RAM** 读出 并/或 写入数据

(双端口存储器 **Dual-Ported RAM** , 可以分别从主站及本地微处理器uC访问):

DPRAM 分为两个区:

- **Registers** [0x0000 - 0x0FFF]: 存储底层的设置和诊断信息
- **Process Data RAM** [\geq 0x1000]: 用于应用层的主从通讯 (Process Data and mailbox).



TwinCAT 中的 Register 访问

Registers 可以在从站的 Advanced Settings 界面中直接监视 (部分可写)

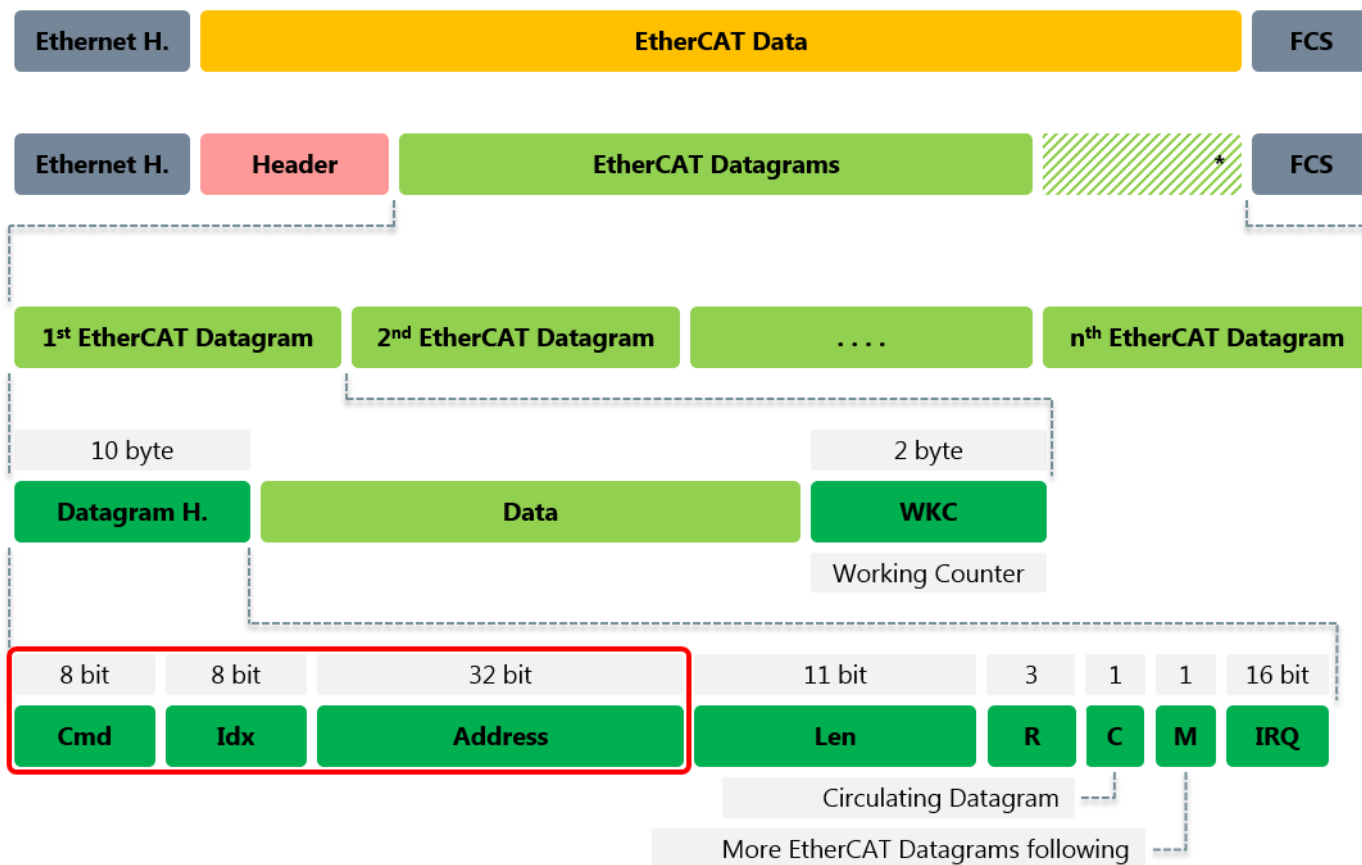
The screenshot shows the TwinCAT configuration interface. On the left, the 'Solution Explorer' displays a project structure with 'Term 7 (EL4102)' selected. The main window shows the 'EtherCAT' configuration for 'Term 6 (EL3102) - B'. The 'Advanced Settings' dialog is open, showing the 'Memory' tab. The 'Memory' tab contains a table of registers with columns for 'Offs', 'Dec', 'Hex', and 'Char'.

Offs	Dec	Hex	Char
0000	ESC Rev/Type	17	0011
0002	ESC Build	3	0003
0004	SM/FMMU Cnt	2056	0808
0006	Ports/DPRAM	2568	0a08
0008	Features	252	00fc
0010	Phys Addr	1007	03ef
0012	Configured Station Alias	0	0000
0020	Register Protect	0	0000
0030	Access Protect	0	0000
0100	ESC Ctrl	61441	f001
0102	ESC CtrlEx	199	00c7
0108	Phys. RW Offset	0	0000



EtherCAT 数据帧和数据报文

在EtherCAT网络中，Ethernet 帧的有效数据叫作“数据报文”

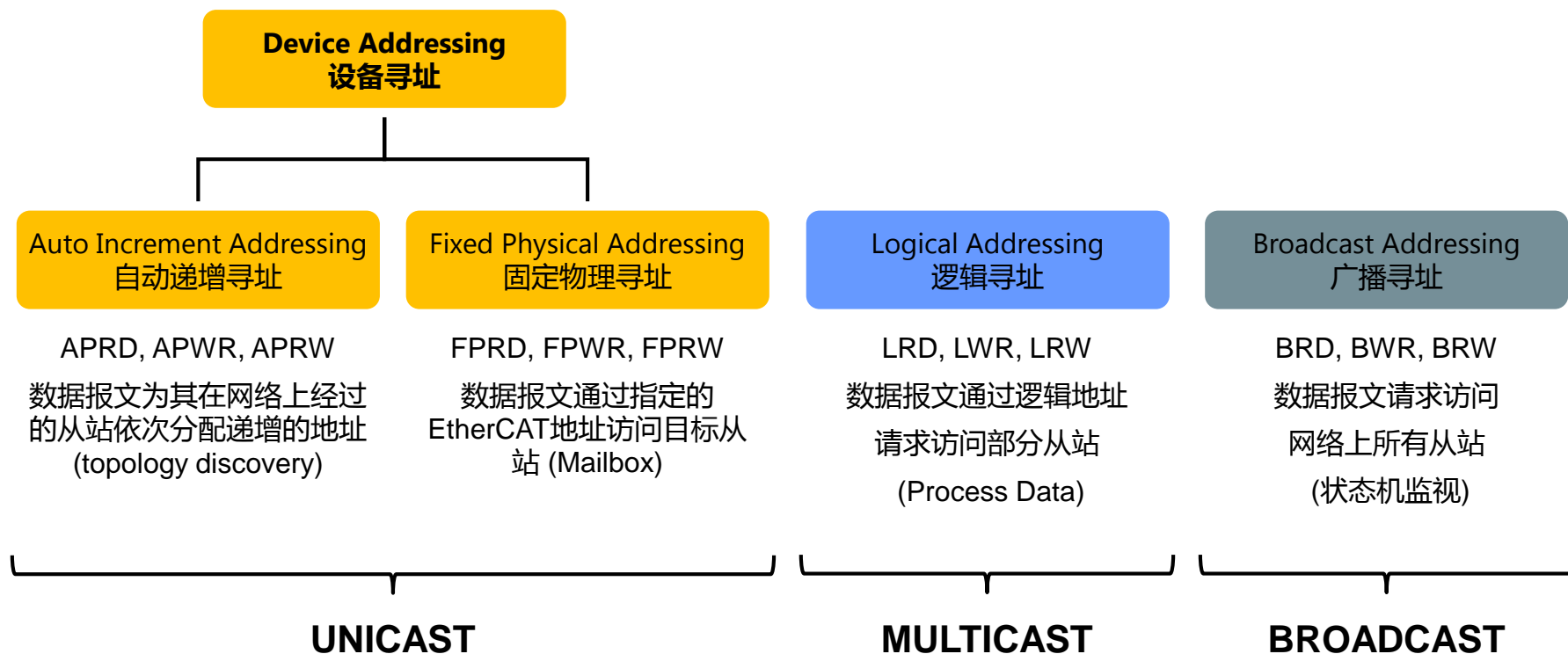


*add 1-32 padding bytes if Ethernet frame is less than 64 bytes



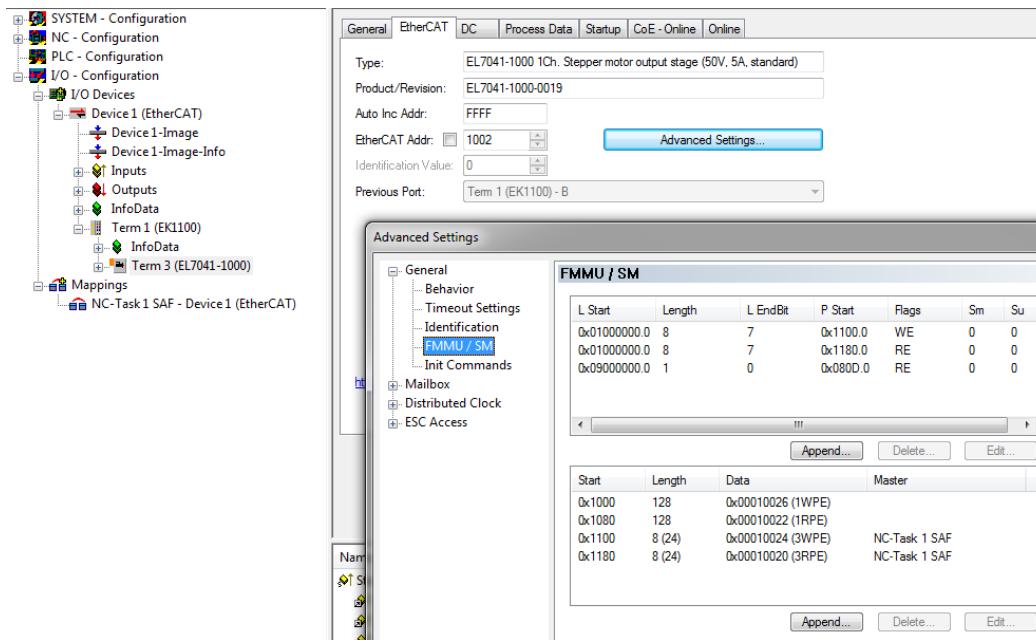
EtherCAT 寻址模式

每个数据报文向一个、多个或者所有从站请求 DPRAM 的内存访问（根据所用的寻址模式）



SyncManagers 和 FMMUs

- **SyncManagers** 阻止主站和从站微处理器 (uC) 同时访问 ESC存储区 , 确保数据的一致性
→ 含周期性数据 (Process Data) 和非周期性数据 (Mailbox)
- **FMMUs** 为Lxx数据报文完成逻辑地址到物理地址的转换
→ 仅对于周期性数据 (Process Data)



从站的 SyncManagers 和 FMMU 是由主站在初始化阶段自动配置的 , 该配置基于每个从站的 XML 文件和整个网络的设置。



SyncManager 和 FMMU 的设置

- Logical Write (逻辑写) 访问：



Edit FMMU

Type: Outputs

Log. Start Address: 16777216

Length: 8

Phys. Start Address: 4352

Read Access

Write Access

Enable

FMMU 把报文头部命令的逻辑地址 翻译对应的物理地址

Edit Sync Manager

Type: Outputs

Start Address: 4352

Length: 8

Buffer: 3

Enable

Watchdog Trigger

SyncManager 往DPRAM 写入连贯一致的数据

Frame	Cmd	Addr	Size	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
0	LRD	0x09000000	2	2.000			
0	LRW	0x01000000	8	2.000			
0	BRD	0x0000 0x0130	2	2.000	0.34	63 / 6.96	1
					0.35		

数据从主站发出的逻辑命令中读出



SyncManager 和 FMMU 的设置

- Logical Read (逻辑读) 访问:



The 'Edit FMMU' dialog box shows the configuration for a memory-mapped I/O. The 'Type' is set to 'Inputs'. The 'Log. Start Address' is 16777216, and the 'Phys. Start Address' is 4480. The 'Length' is 8. The 'Read Access' checkbox is checked. A blue callout box points to the 'Phys. Start Address' field with the text: "FMMU 把命令头部的物理地址转换到对应的逻辑地址".

The 'Edit Sync Manager' dialog box shows the configuration for a sync manager. The 'Type' is 'Inputs'. The 'Start Address' is 4480, and the 'Length' is 8. The 'Buffer' is set to 3. The 'Enable' checkbox is checked. A blue callout box points to the 'Start Address' field with the text: "SyncManager 从 DPRAM 读取连贯一致的数据".

数据写入到主站发出的逻辑命令中

Frame	Cmd	Addr	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
0	LRD	0x09000000	2.000			
0	LRW	0x01000000	2.000			
0	BRD	0x0000 0x0130	2.000	0.34	63 / 6.96	1
				0.35		

