

# **Logos2 系列 FPGA 可配置逻辑模块(CLM)**

## **用户指南**

(UG040001, V1.4)

(2023.07.14)

深圳市紫光同创电子有限公司

版权所有 侵权必究

---

## 文档版本修订记录

| 版本号  | 发布日期       | 修订记录  |
|------|------------|---|
| V0.1 | 2020/02/14 | 初始版本  |
| V1.0 | 2020/10/19 | <ol style="list-style-type: none"> <li>1.统一刷新格式；</li> <li>2.更新<b>总体介绍</b>章节中<b>CLM</b>介绍和功能特性说明；</li> <li>3.更新<b>CLM 的使用</b>小节中<b>CLM</b>使用方式介绍的说明；</li> <li>4.更新<b>CLM 结构简介</b>小节中<b>CLM</b>结构的说明；</li> <li>5.删除<b>多路选择器模式</b>小节多路选择器实现示意图及其说明。</li> </ol>   |
| V1.1 | 2021/10/14 | <ol style="list-style-type: none"> <li>1.总体介绍章节新增<b>资源说明</b>；</li> <li>2.<b>CLM 结构简介</b>小节新增<b>图 2-1</b>、<b>图 2-2</b>中部分端口的说明及共用资源描述；</li> <li>3.更新<b>图 2-1</b>、<b>图 2-2</b>，附加寄存器布线资源输入新增<b>AD/BD/CD/DD</b>端口，同时新增<b>表 2-1</b>及其描述；</li> <li>4.分布式<b>RAM 模式</b>小节中<b>表 2-4</b>以及分布式<b>RAM GTP</b>小节中<b>表 3-9</b>新增<b>GTP_RAM32X2SP</b>、<b>GTP_RAM32X2DP</b>；</li> <li>5.分布式<b>RAM 模式</b>小节新增读写冲突描述；</li> <li>6.分布式<b>RAM 模式</b>小节新增分布式<b>RAM</b>使用说明；</li> <li>7.更新分布式<b>RAM 模式</b>小节<b>图 2-4</b>、<b>图 2-5</b>、<b>图 2-6</b>、<b>图 2-7</b>、<b>图 2-8</b>、<b>图 2-9</b>、<b>图 2-10</b>及其说明；</li> <li>8.新增<b>RAM32X2SP</b>小节、<b>RAM32X2DP</b>小节；</li> <li>9.附加寄存器小节新增<b>PG2L100H</b>以外的<b>Logos2</b>器件布线资源输入描述；</li> <li>10.新增<b>CLM 时序参数</b>小节，新增<b>CLM</b>的典型时序参数说明和时序图；</li> <li>11.修正分布式<b>RAM GTP</b>小节中<b>图 3-5</b>和<b>图 3-6</b>；</li> <li>12.分布式<b>RAM GTP</b>小节新增<b>触发器 GTP</b>小节、<b>锁存器 GTP</b>小节；</li> <li>13.删除<b>CLM 的使用</b>章节中表格存储资源约束方式，新增参考文档的说明。</li> </ol> |
| V1.2 | 2021/12/28 | <ol style="list-style-type: none"> <li>1.分布式<b>RAM 模式</b>小节中<b>表 2-4</b>以及分布式<b>RAM GTP</b>小节中<b>表 3-9</b>新增<b>GTP_RAM32X2X4</b>、<b>GTP_RAM64X1X4</b>；</li> </ol>   |
| V1.3 | 2022/09/23 | <ol style="list-style-type: none"> <li>1.更新分布式<b>RAM 模式</b>小节中分布式<b>RAM</b>的<b>推荐地址深度</b>描述；</li> <li>2.<b>CLM 结构简介</b>小节中新增不同器件<b>CLM</b>结构差异描述及<b>表 2-1</b>；</li> <li>3.<b>CLM 结构简介</b>小节中新增注释说明<b>CLMA</b>的<b>AD</b>端口使用限制；</li> <li>4.分布式<b>RAM 模式</b>小节中新增<b>RAM32X2X4</b>小节、<b>RAM64X1X4</b>小节，修正<b>图 2-11</b>、<b>图 2-12</b>，修正<b>图 2-6</b>、<b>图 2-7</b>及其描述；</li> <li>5.寄存器控制信号小节新增<b>PG2L200H</b>时钟使能端口差异描述及<b>图 2-17</b>；</li> </ol>   |
| V1.4 | 2023/07/14 | <ol style="list-style-type: none"> <li>1.分布式<b>RAM GTP</b>小节中修正<b>图 3-5</b>，增加数据写入之前读输出随地址的跳变；</li> <li>2.分布式<b>RAM GTP</b>小节中更新<b>图 3-8</b>中信号名称；</li> <li>3.删除<b>表 3-4</b>、<b>表 3-10</b>、<b>表 3-13</b>、<b>表 3-17</b>、<b>表 3-20</b>对<b>GTP</b>参数默认值的描述；</li> <li>4.更新<b>图 3-5</b>、<b>图 3-6</b>的读写时序说明；</li> <li>5.进位链<b>GTP</b>小节新增<b>GTP_LUT6CARRY</b>内部结构说明、<b>图 3-3</b>、<b>表 3-6</b>、<b>表 3-7</b>和<b>表 3-8</b>；</li> <li>6.更新<b>表 2-5</b>、<b>图 2-18</b>中时序参数的名称；</li> <li>7.更新<b>表 3-16</b>、<b>表 3-19</b>中部分端口的详细说明；</li> <li>8.更新<b>表 3-4</b>中参数具体有效值的功能描述；</li> <li>9.更新<b>CLM 的使用</b>小节中引用<b>IP</b>文档的名称，增加文档编号。</li> </ol>  |

## 术语与缩略语

| Terms and Abbreviations<br>术语与缩略语 | Full Spelling<br>英文全拼        | Chinese Explanation<br>中文解释 |
|-----------------------------------|------------------------------|-----------------------------|
| CLM                               | Configurable Logic Module    | 可配置逻辑模块                     |
| LUT                               | Look Up Table                | 查找表                         |
| SRB                               | Singal Relay Block           | 信号互连模块                      |
| GTP                               | Generic Technology Primitive | 通用技术原语                      |
|                                   |                              |                             |

## 目录

|                               |    |
|-------------------------------|----|
| 文档版本修订记录.....                 | 1  |
| 术语与缩略语.....                   | 2  |
| 目录.....                       | 3  |
| 表目录.....                      | 5  |
| 图目录.....                      | 6  |
| 1 总体介绍.....                   | 8  |
| 2 功能描述.....                   | 10 |
| 2.1 CLM 结构简介 .....            | 10 |
| 2.2 LUT6A 和 LUT6S 的工作模式 ..... | 12 |
| 2.2.1 逻辑功能模式 .....            | 12 |
| 2.2.2 多路选择器模式 .....           | 12 |
| 2.2.3 算数功能模式 .....            | 12 |
| 2.2.4 ROM 模式 .....            | 13 |
| 2.2.5 分布式 RAM 模式 .....        | 14 |
| 2.3 CLM 寄存器的工作模式 .....        | 24 |
| 2.3.1 主寄存器 .....              | 24 |
| 2.3.2 附加寄存器 .....             | 25 |
| 2.3.3 寄存器控制信号 .....           | 25 |
| 2.4 CLM 时序参数 .....            | 26 |
| 3 应用示例.....                   | 28 |
| 3.1 常用 GTP .....              | 28 |
| 3.1.1 多路选择器 GTP.....          | 28 |
| 3.1.2 进位链 GTP.....            | 29 |
| 3.1.3 分布式 RAM GTP.....        | 32 |
| 3.1.4 ROM GTP.....            | 35 |
| 3.1.5 触发器 GTP.....            | 37 |
| 3.1.6 锁存器 GTP.....            | 38 |

---

|                   |    |
|-------------------|----|
| 3.2 CLM 的使用 ..... | 39 |
| 免责声明.....         | 41 |

## 表目录

|                                 |    |
|---------------------------------|----|
| 表 2-1 布线资源到附加寄存器的输入.....        | 12 |
| 表 2-2 1 位全加器真值表.....            | 12 |
| 表 2-3 CLM 支持的 ROM 模式.....       | 13 |
| 表 2-4 CLMS 支持的 RAM 模式.....      | 14 |
| 表 2-5 CLM 典型时序参数 .....          | 26 |
| 表 3-1 多路选择器 GTP .....           | 28 |
| 表 3-2 多路选择器的端口描述.....           | 28 |
| 表 3-3 多路选择器的真值表.....            | 29 |
| 表 3-4 进位链 GTP 参数说明 .....        | 29 |
| 表 3-5 进位链 GTP 端口描述 .....        | 29 |
| 表 3-6 内部信号 LUT5A、LUT5B 真值表..... | 30 |
| 表 3-7 输出信号 COUT 真值表.....        | 31 |
| 表 3-8 输出信号 Z 真值表 .....          | 32 |
| 表 3-9 RAM GTP .....             | 32 |
| 表 3-10 分布式 RAM GTP 参数说明 .....   | 33 |
| 表 3-11 分布式 RAM GTP 端口描述 .....   | 33 |
| 表 3-12 ROM GTP .....            | 35 |
| 表 3-13 分布式 ROM GTP 参数说明 .....   | 35 |
| 表 3-14 分布式 ROM GTP 端口说明 .....   | 36 |
| 表 3-15 触发器 GTP .....            | 37 |
| 表 3-16 触发器 GTP 端口描述.....        | 37 |
| 表 3-17 GTP_DFF_CE 参数描述.....     | 38 |
| 表 3-18 锁存器 GTP .....            | 38 |
| 表 3-19 锁存器 GTP 端口描述.....        | 39 |
| 表 3-20 GTP_DLATCH_CE 参数描述 ..... | 39 |

## 图目录

|   |    |
|---|----|
| 图 1-1 CLM 分布图 .....                               | 8  |
| 图 2-1 CLMA 逻辑框图 .....                             | 10 |
| 图 2-2 CLMS 逻辑框图 .....                             | 11 |
| 图 2-3 加法器实现示意图 .....                              | 13 |
| 图 2-4 RAM32X1SP 实现方式逻辑示意图 .....                   | 15 |
| 图 2-5 RAM32X1DP 实现方式逻辑示意图 .....                   | 16 |
| 图 2-6 RAM64X1SP 实现方式逻辑示意图 .....                   | 16 |
| 图 2-7 RAM64X1DP 实现方式逻辑示意图 .....                   | 17 |
| 图 2-8 RAM128X1SP 实现方式逻辑示意图 .....                  | 18 |
| 图 2-9 RAM128X1DP 实现方式逻辑示意图 .....                  | 19 |
| 图 2-10 RAM256X1SP 实现方式逻辑示意图 .....                 | 20 |
| 图 2-11 RAM32X2SP 实现方式逻辑示意图 .....                  | 21 |
| 图 2-12 RAM32X2DP 实现方式逻辑示意图 .....                  | 21 |
| 图 2-13 RAM32X2X4 实现方式逻辑示意图 .....                  | 22 |
| 图 2-14 RAM64X1X4 实现方式逻辑示意图 .....                  | 23 |
| 图 2-15 主寄存器示意图 .....                              | 24 |
| 图 2-16 附加寄存器示意图 .....                             | 25 |
| 图 2-17 PG2L200H 和 Logos2 系列其他器件的 CE 信号差异示意图 ..... | 26 |
| 图 2-18 CLM 时序图 .....                              | 27 |
| 图 3-1 多路选择器 GTP 端口示意图 .....                       | 28 |
| 图 3-2 进位链 GTP 端口示意图 .....                         | 29 |
| 图 3-3 GTP_LUT6CARRY 内部结构示意图 .....                 | 30 |
| 图 3-4 RAM GTP 端口示意图 .....                         | 33 |
| 图 3-5 分布式单端口 RAM 时序图 .....                        | 34 |
| 图 3-6 分布式简单双口 RAM 时序图 .....                       | 35 |
| 图 3-7 ROM GTP 端口示意图 .....                         | 36 |
| 图 3-8 GTP_ROM256X1 波形示意图 .....                    | 36 |

---

|                                  |    |
|----------------------------------|----|
| 图 3-9 GTP_DFF_CE 端口示意图.....      | 37 |
| 图 3-10 GTP_DLATCH_CE 端口示意图 ..... | 38 |



## 1 总体介绍

CLM(Configurable Logic Module, 可配置逻辑模块)是 Logos2 系列产品的基本逻辑单元。CLM 在 Logos2 系列产品中按列分布,支持 CLMA 和 CLMS 两种形态。CLMA 和 CLMS 均可实现逻辑、算术、移位寄存器以及 ROM 功能,仅有 CLMS 支持分布式 RAM 功能,CLMA 与 CLMS 数量比例约为3:1。CLM 与 CLM 之间、CLM 与其它片内资源之间通过信号互连模块(SRB)连接,如图 1-1 所示。

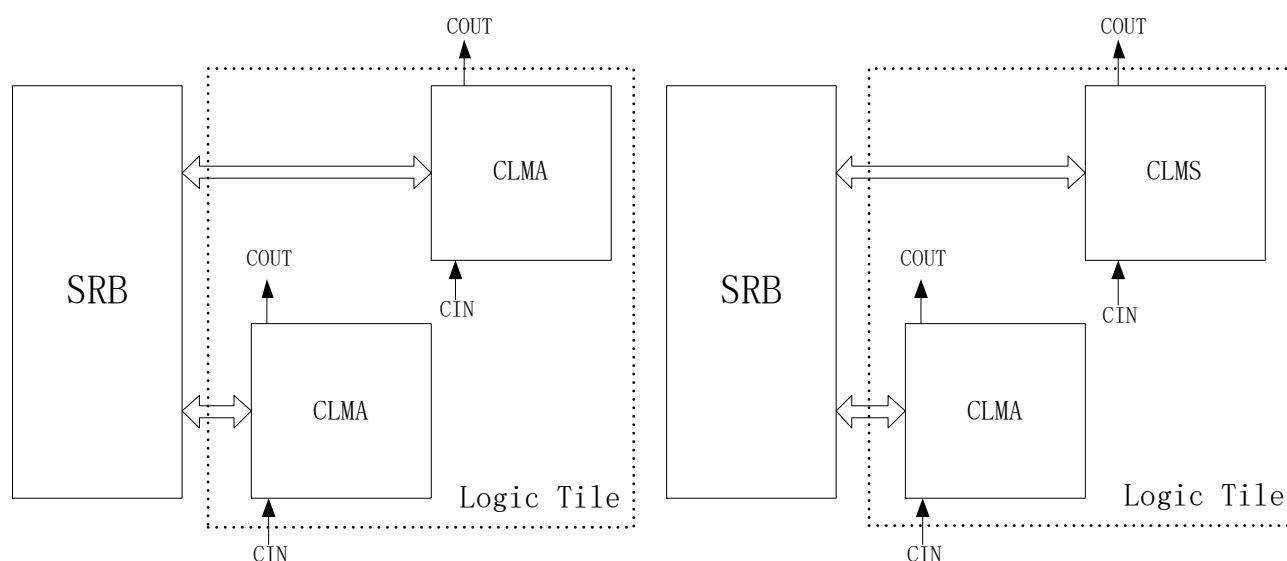


图 1-1 CLM 分布图

Logos2 系列 FPGA 产品的 CLM 主要特性包括:

- 采用创新的 LUT6 逻辑结构
- 每个 CLM 包含 4 个多功能 LUT6
- 每个 CLM 包含 8 个寄存器
- 支持算术功能模式
- 支持快速算术进位逻辑
- 可高效实现多路选择功能
- 可实现 ROM 功能
- 支持移位寄存器级联链
- CLMS 支持分布式 RAM 模式

Logos2 系列的 CLM 采用了 LUT6 结构,并且改进 carrychain 的结构,对于 Flip-Flop,增加了单个 CLM 包含的数量,同时对其内部逻辑进行了优化。保证时序的同时,消耗更小的功耗,整体上对大容量的设计,支持力度更友好。

器件容量通常由一个 LUT4 和一个触发器组成的逻辑单元来衡量，如上文所述，Logos2 系列 FPGA 产品的 CLM 采用创新的 LUT6 逻辑结构，包含丰富的触发器、锁存器、进位逻辑以及能够在片内创建分布式 RAM/ROM 的能力，因此增加了其有效容量。逻辑单元的数目与 LUT6 的比率是 1.6:1, Logos2 系列器件的 CLM 资源规模说明见《*DS04001\_Logos2 系列 FPGA 器件数据手册*》。

## 2 功能描述

### 2.1 CLM 结构简介

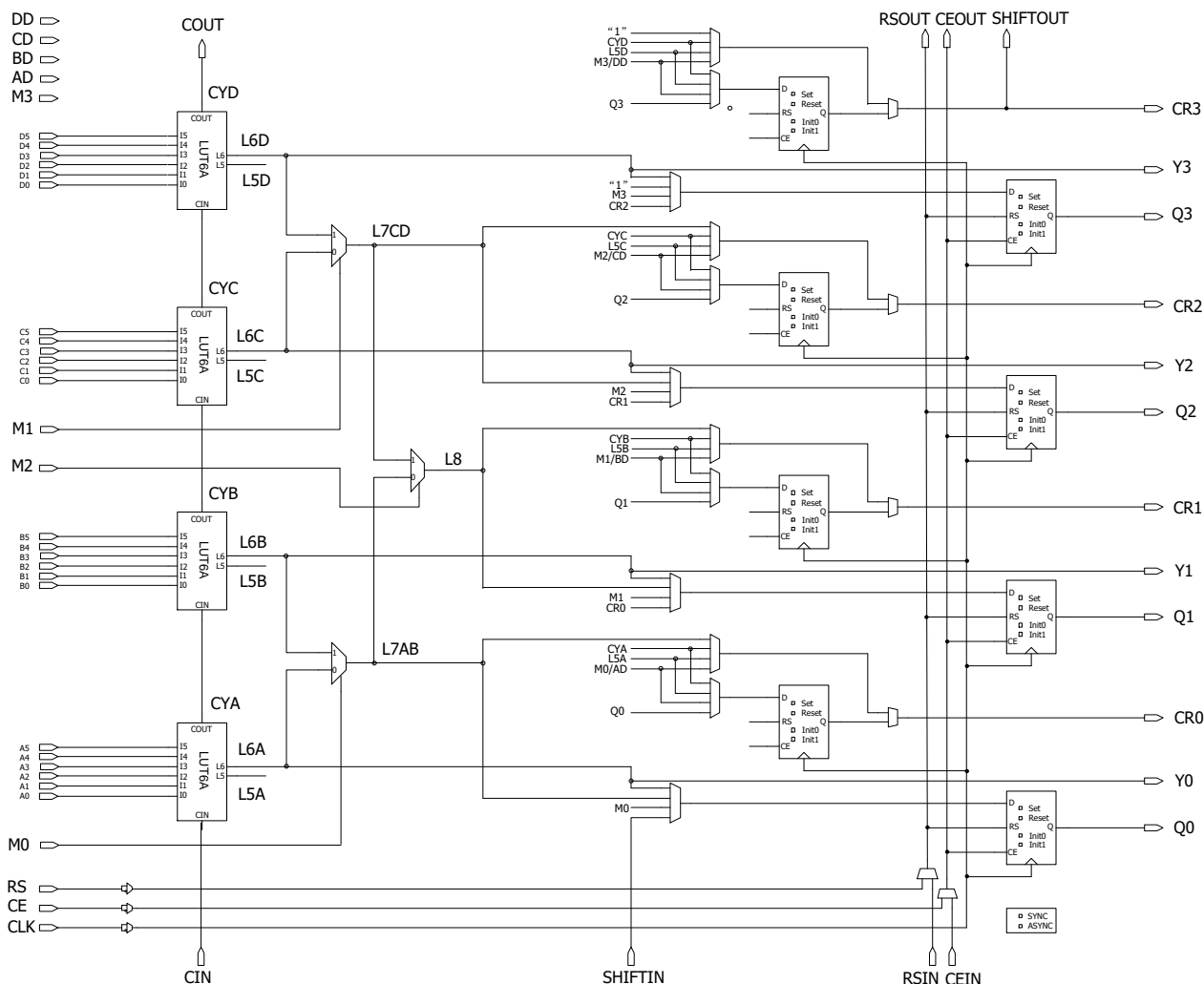


图 2-1 CLMA 逻辑框图

CLMA 的逻辑框图如图 2-1 所示，每个 CLMA 包含 4 个 6 输入 LUTs、8 个寄存器、多个扩展功能选择器、以及 4 条独立的级联链等。其中 LUT6A（为了方便区分，下文 LUT6A 作为 CLMA 中 4 个 LUT6 的统称，把输入分别为 A0~A5、B0~B5、C0~C5、D0~D5 的 LUT6A 分别称作 LUT6A\_A、LUT6A\_B、LUT6A\_C、LUT6A\_D）采用了创新的架构设计，在 6 输入查找表的基础上集成了专用电路以实现 4:1 多路选择器功能和快速算术进位逻辑；扩展功能选择器主要用于实现宽位查找表和输出选择功能；级联链包括算术逻辑进位链（从 CIN 到 COUT），专用移位寄存器链（从 SHIFTIN 到 SHIFTOUT），寄存器复/置位级联链（从 RSIN 到 RSOUT）和寄存器 CE 级联链（从 CEIN 到 CEOUT）；A0~A5/B0~B5/C0~C5/D0~D5 分别为 4 个 LUT6 的 6 个输入端口，M0~M2 为扩展 LUT7/LUT8 的多路选择器的控制端，RS 为寄存器复/置位信

号, CE 为时钟使能, CLK 为时钟输入, Y0~Y3 为 4 个 LUT6 的数据输出, CR0~CR3 为 LUT7/8 和 CR 寄存器等输出的选择输出端口, Q0~Q3 为 LUT6 和寄存器（不同于 CR 寄存器）等输出的选择输出端口。

CLMS 是 CLMA 的扩展, 它在支持 CLMA 所有功能的基础上增加了对分布式 RAM 的支持, 其中的多功能 LUT6 称为 LUT6S（为了方便区分, 下文 LUT6S 作为 CLMS 中 4 个 LUT6 的统称, 把输入分别为 A0~A5、B0~B5、C0~C5、D0~D5 的 LUT6S 分别称作 LUT6S\_A、LUT6S\_B、LUT6S\_C、LUT6S\_D）。CLMS 可配置为大小为 64\*4 的 SP(Single Port, 单口) RAM 或大小为 64\*3 的 SDP(Simple Dual Port, 简单双口) RAM。CLMS 的逻辑框图如图 2-2 所示, 其中 AD~DD 和 M0~M3 为 4 个 LUT6 的分布式 RAM 模式写数据输入, WE 为分布式 RAM 模式写使能。

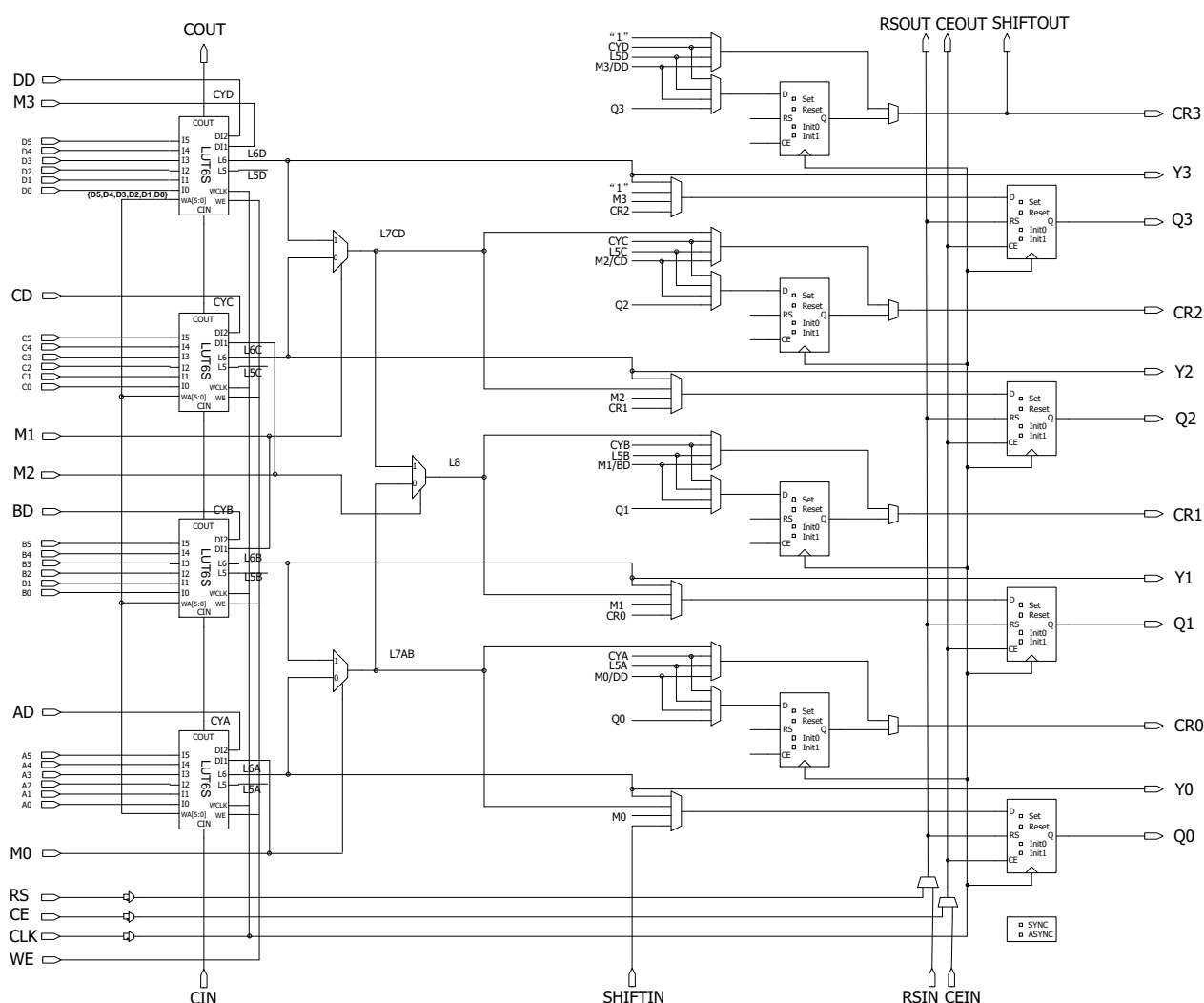


图 2-2 CLMS 逻辑框图

CLMA 和 CLMS 部分资源存在共用, 只使用了内部部分资源时占用了例如 RS、CE、CLK 等共用资源后剩余资源无法独立使用。例如使用了 CLMA 中的一个 LUT6 和 CE 时钟使能, 剩余 LUT6 无法使用与之独立的 CE 时钟使能进行控制。

图 2-1 和图 2-2 中，布线资源到附加寄存器（左边列寄存器）的输入复用 M0/M1/M2/M3 或 AD/BD/CD/DD 端口。其中，PG2L100H 等器件复用 M0/M1/M2/M3 端口，PG2L50H 等器件复用 AD/BD/CD/DD，如下表所示。此外 PG2L200H 的 CE 信号结构和上图图片存在差别，详见图 2-17。

表 2-1 布线资源到附加寄存器的输入

| 器件                | 复用端口        |
|-------------------|-------------|
| PG2L100H/PG2L200H | M0/M1/M2/M3 |
| PG2L25H/PG2L50H   | AD/BD/CD/DD |

注：CLMA 复用的 AD 端口，由于布线资源输出受限，其在硬件上是悬空的。

## 2.2 LUT6A 和 LUT6S 的工作模式

LUT6A 和 LUT6S 可灵活配置，以支持基本逻辑，多路选择，算术逻辑，ROM 功能以及分布式 RAM 功能（仅限于 LUT6S）等不同功能。

### 2.2.1 逻辑功能模式

在逻辑功能模式下，每个 LUT6A（或 LUT6S）可实现 1 个 LUT6，结合扩展功能选择器，每个 CLM 可支持实现 4 个 LUT6、2 个 LUT7 或 1 个 LUT8。

### 2.2.2 多路选择器模式

在多路选择模式下，每个 LUT6A（或 LUT6S）可实现 1 个 4:1 多路选择器，每个 CLM 可支持 4 个 4:1 多路选择器。结合扩展功能选择器，每 2 个 LUT6A（或 LUT6S）可支持 1 个 8:1 多路选择器，每个 CLM 可支持 2 个 8:1 多路选择器；每 4 个 LUT6A（或 LUT6S）可支持 1 个 16:1 多路选择器，每个 CLM 可支持 1 个 16:1 多路选择器。更宽的多路数据选择可通过 CLM 组合产生。

### 2.2.3 算数功能模式

在算术功能模式下，LUT6A(或 LUT6S)可实现加减法运算、计数器、比较器、快速异或逻辑运算以及宽位与逻辑运算等。

以加法器为例，实现一个全加器，其真值表如表 2-2 所示。

表 2-2 1 位全加器真值表

| A | B | CIN | COUT | SUM |
|---|---|-----|------|-----|
| 0 | 0 | 0   | 0    | 0   |
| 0 | 0 | 1   | 0    | 1   |
| 0 | 1 | 0   | 0    | 1   |
| 0 | 1 | 1   | 1    | 0   |

| A | B | CIN | COUT | SUM |
|---|---|-----|------|-----|
| 1 | 0 | 0   | 0    | 1   |
| 1 | 0 | 1   | 1    | 0   |
| 1 | 1 | 0   | 1    | 0   |
| 1 | 1 | 1   | 1    | 1   |

逻辑表达式为:

$$COUT = A \oplus B * CIN + !(A \oplus B) * B$$

$$SUM = A \oplus B \oplus CIN = A \oplus B * !CIN + !(A \oplus B) * CIN$$

CLM 的结构通过如下图 2-3 所示的硬件电路实现全加器。

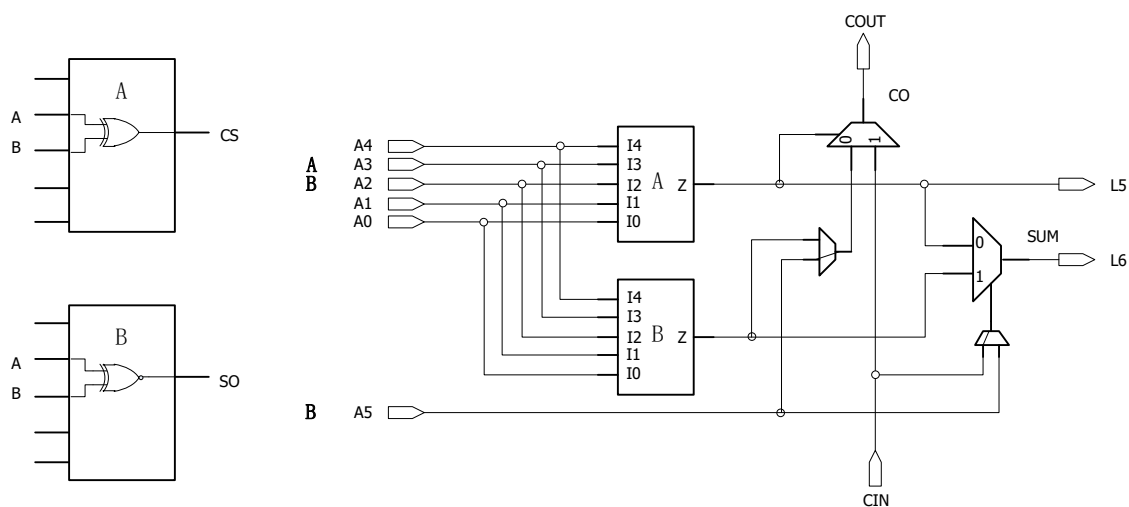


图 2-3 加法器实现示意图

LUT5\_A 配置成  $A \oplus B$  功能，作为进位链 mux 的选择端，当其为高电平时选择 CIN，为低电平时选择 B 输入，其输出作为进位输出。全加器的 SUM 输出，CIN 作为 LUT6 MUX 的选择端口，CIN 为高电平选择 LUT5\_B，CIN 为低电平选择 LUT5\_A，其中 LUT5\_B 配置成  $A \odot B$ 。

通过上述方式，实现加法器功能。

## 2.2.4 ROM 模式

在 ROM 模式下，LUT6A(或 LUT6S)可以用作  $64 \times 1$  ROM，还可以通过内置的扩展功能选择器进行深度级联。ROM 数据的初始化在编程配置过程中完成。

支持的全部 ROM 模式如表 2-3 所示。

表 2-3 CLM 支持的 ROM 模式

| ROM 模式        | GTP         | 需要的 LUT 数量 |
|---------------|-------------|------------|
| $32 \times 1$ | GTP_ROM32X1 | 1          |
| $32 \times 2$ | GTP_ROM32X2 | 1          |

| ROM 模式 | GTP          | 需要的 LUT 数量 |
|--------|--------------|------------|
| 64×1   | GTP_ROM64X1  | 1          |
| 128×1  | GTP_ROM128X1 | 2          |
| 256×1  | GTP_ROM256X1 | 4          |

### 2.2.5 分布式 RAM 模式

分布式 RAM 提供了在非常小的阵列使用储存元素和在较大的阵列使用 DRM 的折衷方案，用户可通过内存的大小灵活使用分布式 RAM 和 DRM。分布式 RAM 可以通过软件 Pango Design Suite 内嵌的 IP Compiler 工具或例化 GTP 原语生成。

一般来说，分布式 RAM 建议用于所有深度小于等于 64 的场景，除非目标器件的逻辑资源短缺。分布式 RAM 在资源、性能和功耗方面有更好的表现。对于深度大于 64 但小于等于 128 的场景，决定使用分布式 RAM 还是 DRM 取决于以下因素：

- 可用性，如果 DRM 资源不足，则可以使用分布式 RAM；
- 延迟方面存在需求，例如需要异步读能力，则必须使用分布式 RAM；
- 数据宽度，大于等于 16 位应该尽可能使用 DRM；
- 必要的性能要求，分布式 RAM 通常比 DRM 拥有更短的输出延迟和更少的位置限制。

此模式只涉及 CLMS。单个 CLMS 可以根据需要配置成各种大小的 RAM 块，RAM 的地址与位宽根据配置会做调整。表 2-4 所列是 Logos2 能够支持的 RAM 模式。

表 2-4 CLMS 支持的 RAM 模式

| RAM 模式     | GTP            | 描述   |
|------------|----------------|--|
| RAM64X1SP  | GTP_RAM64X1SP  | 一个 LUT6 配置成 64bits 的单端口 RAM  |
| RAM32X1SP  | GTP_RAM32X1SP  | 一个 LUT6 可以配置成两个 32bits 共享地址的单端口 RAM                                    |
| RAM64X1DP  | GTP_RAM64X1DP  | 一个 LUT6 可以配置成 64bits 的简单双端口 RAM  |
| RAM32X1DP  | GTP_RAM32X1DP  | 一个 LUT6 可以配置成两个 32bits 共享地址的简单双端口 RAM                                  |
| RAM128X1SP | GTP_RAM128X1SP | 两个 LUT6 配置成 128bits 的单端口 RAM，LUT6S_A 与 LUT6S_B 或者 LUT6S_C 与 LUT6S_D 组合 |
| RAM128X1DP | GTP_RAM128X1DP | 两个 LUT6 配置成 128bits 的简单双端口 RAM，LUT6S_A 与 LUT6S_B 组合                    |
| RAM256X1SP | GTP_RAM256X1SP | 四个 LUT6 配置成 256bits 的单端口 RAM，LUT6S_A、LUT6S_B、LUT6S_C、LUT6S_D 组合        |
| RAM32X2SP  | GTP_RAM32X2SP  | 一个 LUT6 可以配置成一个 2x32bits 的单端口 RAM                                      |
| RAM32X2DP  | GTP_RAM32X2DP  | 一个 LUT6 可以配置成一个 2x32bits 的简单双端口 RAM                                    |
| RAM32X2X4  | GTP_RAM32X2X4  | 一个 CLMS 可以配置成 4 个 2x32bits 共享写地址的可配置 RAM                               |
| RAM64X1X4  | GTP_RAM64X1X4  | 一个 CLMS 可以配置成 4 个 1x64bits 共享写地址的可配置 RAM                               |

这里单端口 SP 指的是写地址与读地址共用，简单双端口(S)DP 是一端端口作为写地址进行同步写，另一端端口作为读地址进行异步读。分布式 RAM 模式下，复用 LUT6S\_D 的输入

作为写地址，读地址是复用 LUT6S 本身的输入。用户使用简单双端口时需要防止读写冲突，即同一时刻不能读写相同地址。

### 2.2.5.1 RAM32X1SP

以 LUT6S\_A 为例实现一个 RAM32X1SP，写地址复用 LUT6S\_D 的输入 D[4:0]，读地址是 LUT6S\_A 本身的输入 A[4:0]，因为是单端口类型的 SRAM，需要把 A[4:0]与 D[4:0]连接在一起；写数据端口是 AD 或 M0 输入，读数据输出端口是 Y0，写使能则是 WE 或者 CE。1 个 LUT6S 可以例化 2 个共享地址的 RAM32X1SP。

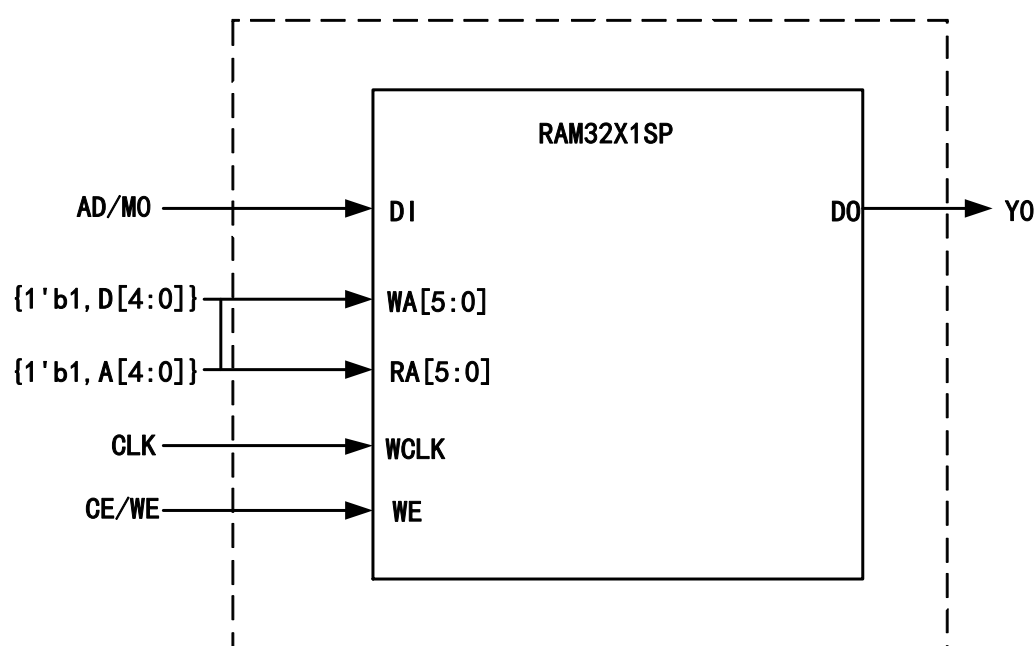


图 2-4 RAM32X1SP 实现方式逻辑示意图

### 2.2.5.2 RAM32X1DP

以 LUT6S\_A 为例实现一个 RAM32X1DP，读写地址独立，复用 LUT6S\_D 的输入端 D[4:0] 作为 RAM 的写地址，读地址为 LUT6S\_A 本身的输入 A[4:0]，写数据端口为 AD 或 M0 输入，读数据输出端口为 Y0，写使能为 WE 或者 CE。一个 LUT6S 可例化 2 个共享地址的 RAM32X1DP。



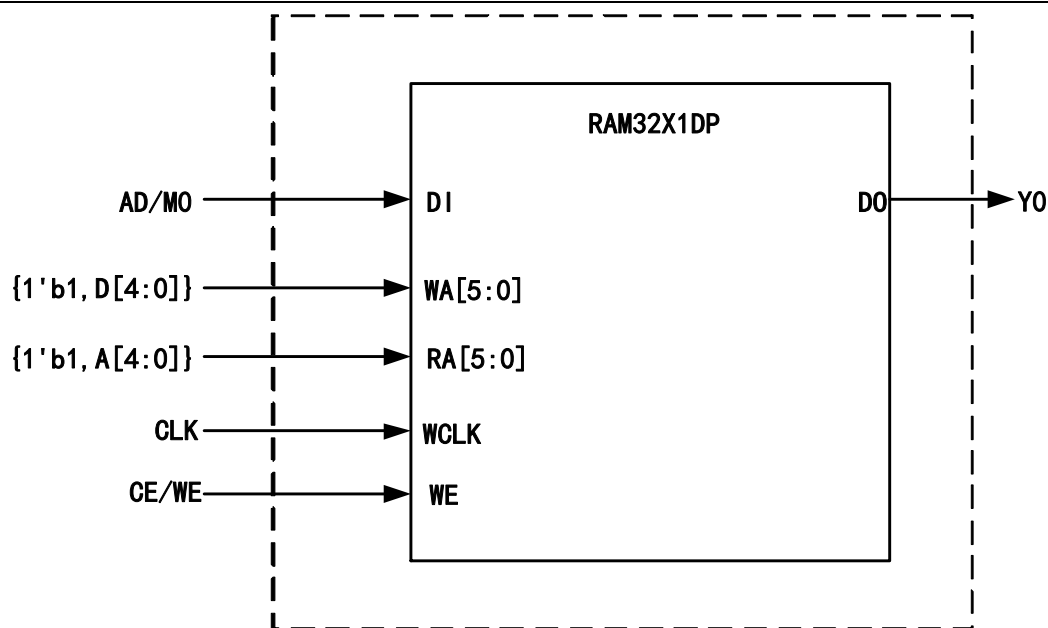


图 2-5 RAM32X1DP 实现方式逻辑示意图

### 2.2.5.3 RAM64X1SP

以 LUT6S\_A 为例实现一个 RAM64X1SP，写地址复用 LUT6S\_D 的输入 D[5:0]，读地址是 LUT6S\_A 本身的输入 A[5:0]，因为是单端口类型的 SRAM，需要把 A[5:0]与 D[5:0]连接在一起；写数据端口是 AD 输入，读数据输出端口是 Y0，写使能则是 WE 或者 CE。1 个 LUT6S 可以例化 1 个 RAM64X1SP，一个 CLMS 可例化 4 个 RAM64X1SP。

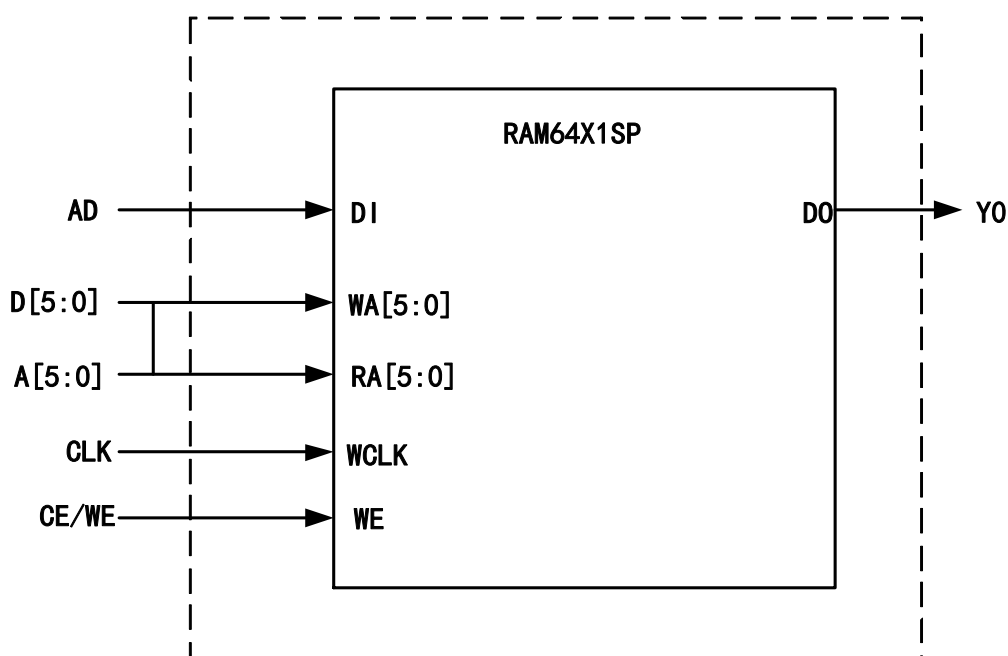


图 2-6 RAM64X1SP 实现方式逻辑示意图

### 2.2.5.4 RAM64X1DP

以 LUT6S\_A 为例实现一个 RAM64X1DP，读写地址独立，复用 LUTS\_6D 的 6 输入端 D[5:0]

作为 RAM 的写地址，读地址为 LUTS\_6A 本身的输入 A[5:0]，写数据端口为 AD 输入，读数数据输出端口为 Y0，写使能为 WE 或者 CE。一个 LUT6S 可以例化 1 个 RAM64X1DP，一个 CLMS 可例化 3 个 RAM64X1DP（LUT6S\_D 的读写地址连接在了一起，无法作为简单双口分布式 RAM 使用）。

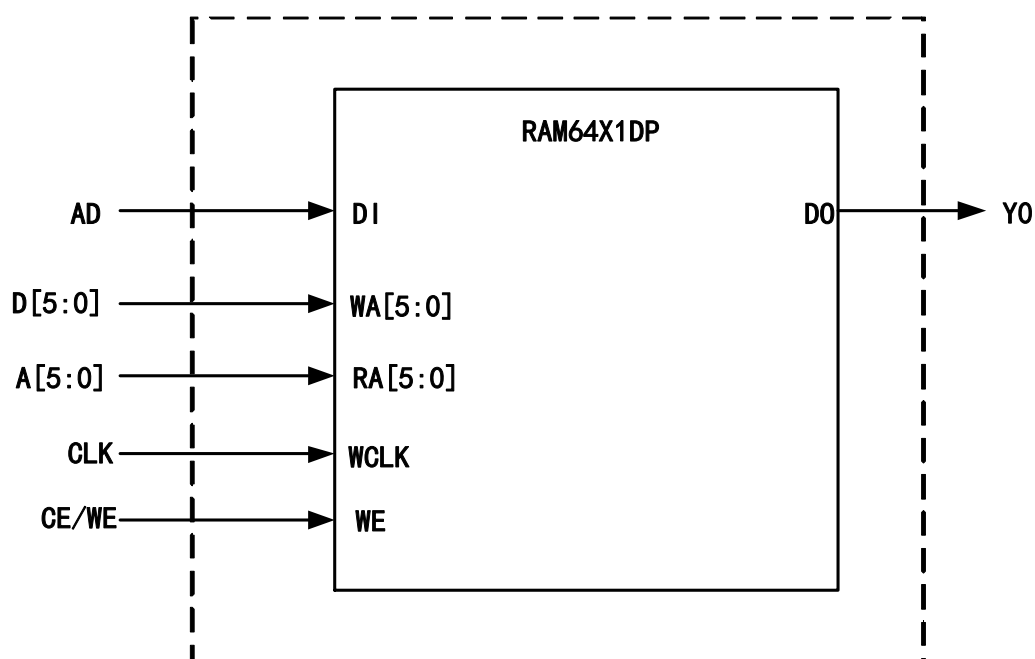


图 2-7 RAM64X1DP 实现方式逻辑示意图

#### 2.2.5.5 RAM128X1SP

图 2-8 是一个 RAM128X1SP 的示意图，需要两个 LUT6S 来实现。DI 分别是 RAM 的写数据，读地址和写地址需要连接在一起，这里读地址用到了 7 bits，所以最高位地址为 M0。一个 CLMS 可以例化两个 RAM128X1SP。

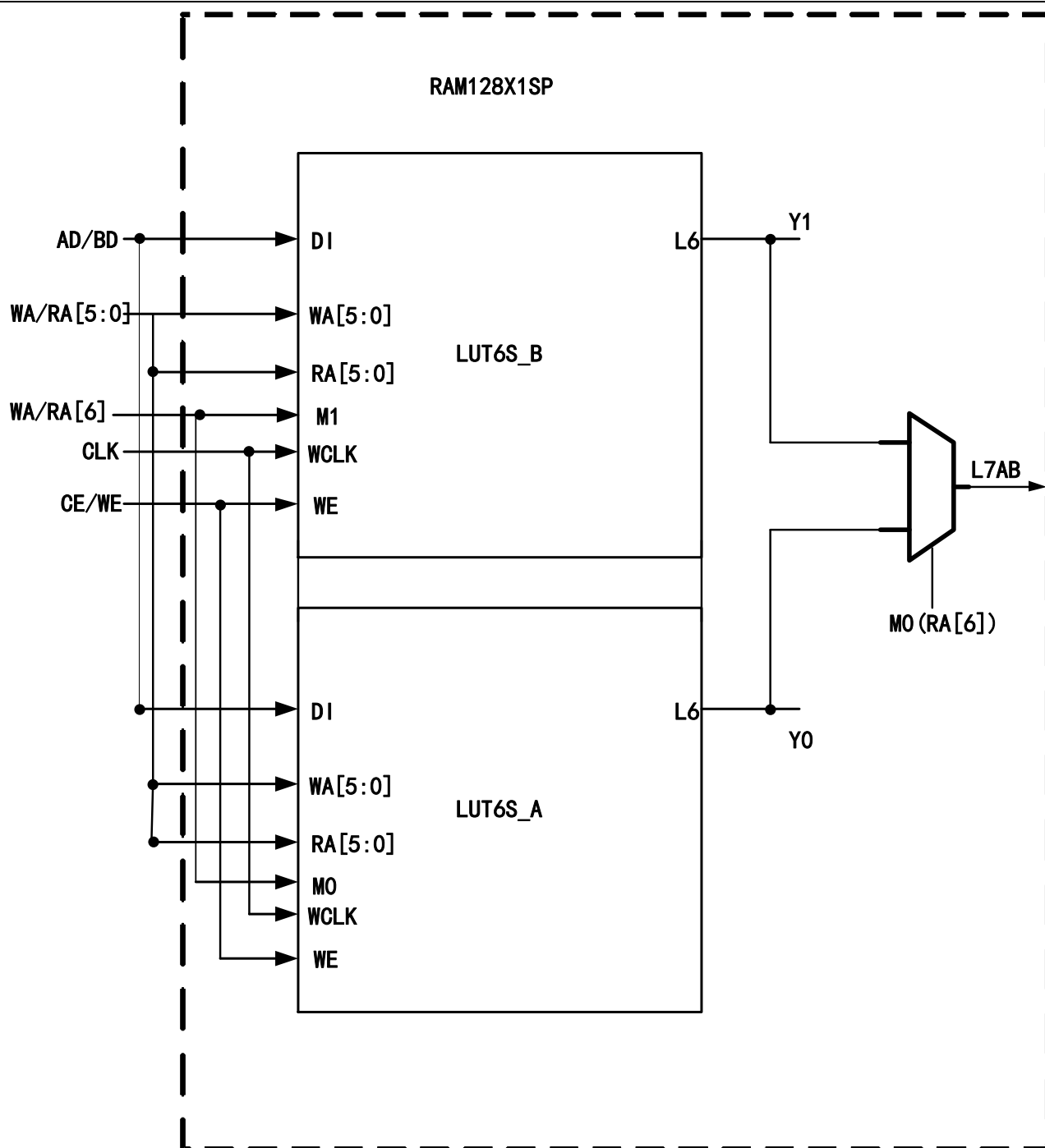


图 2-8 RAM128X1SP 实现方式逻辑示意图

### 2.2.5.6 RAM128X1DP

图 2-9 是一个 RAM128X1DP 的示意图，需要两个 LUT6S 来实现。DI 分别是 RAM 的写数据，读地址复用 LUT6S\_D 的输入 D[5:0]，并以 M1 作为读地址的最高位 WA[6]；写地址以 LUT6S\_A 和 LUT6S\_B 本身的 6 位输入作为低六位，以 M0 作为读地址的最高位。一个 CLMS 可以例化 1 个 RAM128X1DP（LUT6S\_D 的读写地址连接在了一起，无法作为简单双口分布式 RAM 使用）。

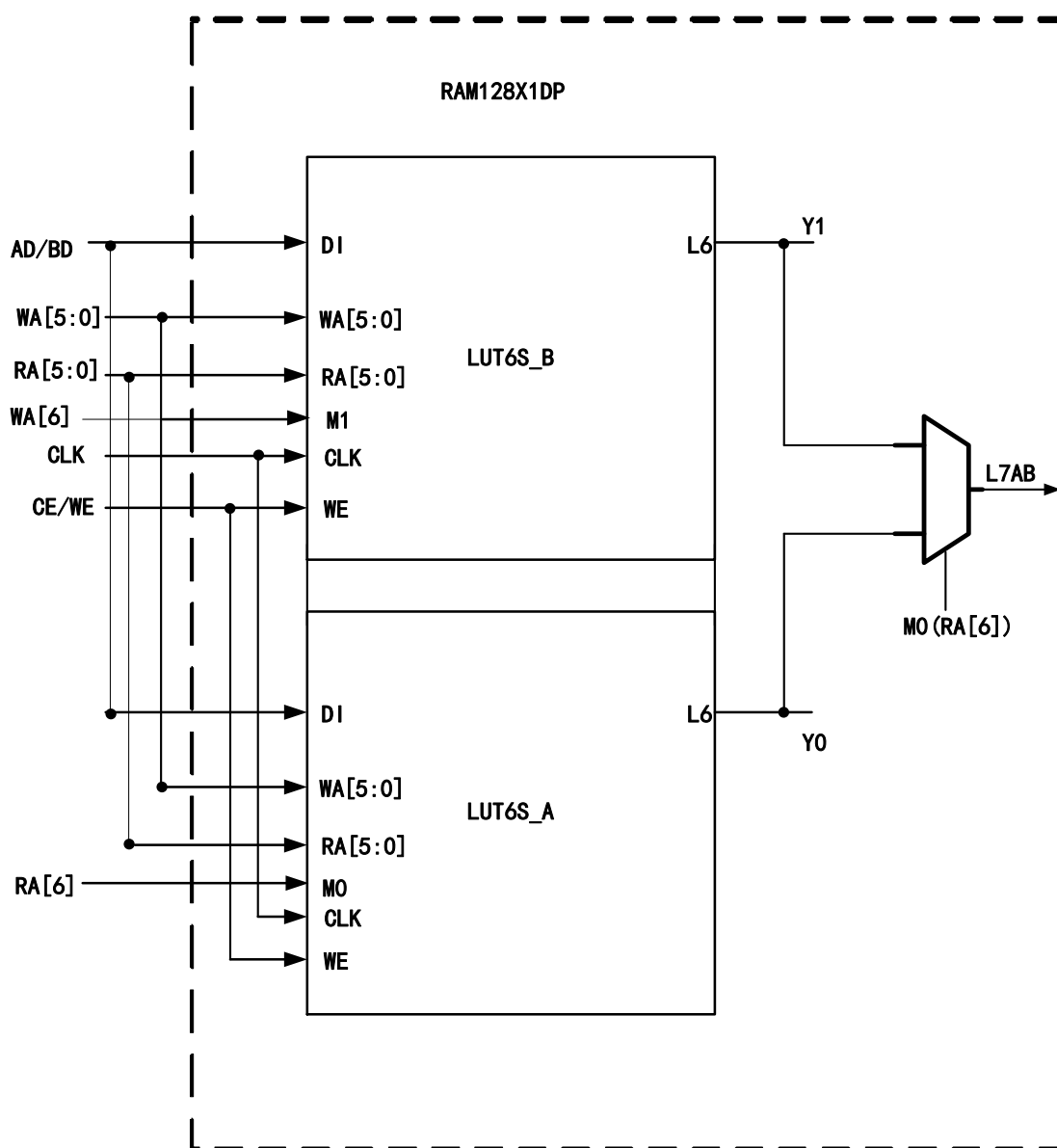


图 2-9 RAM128X1DP 实现方式逻辑示意图

### 2.2.5.7 RAM256X1SP

图 2-10 是一个 RAM256X1SP 的示意图，需要四个 LUT6S 实现。D 是 RAM 的写数据，输入 4 个 LUT6S 的输入端 AD/BD/CD/DD，读地址和写地址需要连接在一块，这里读地址用到了 8 bits，最高位地址是 M2，一个 CLMS 可以例化 1 个 RAM256X1SP。

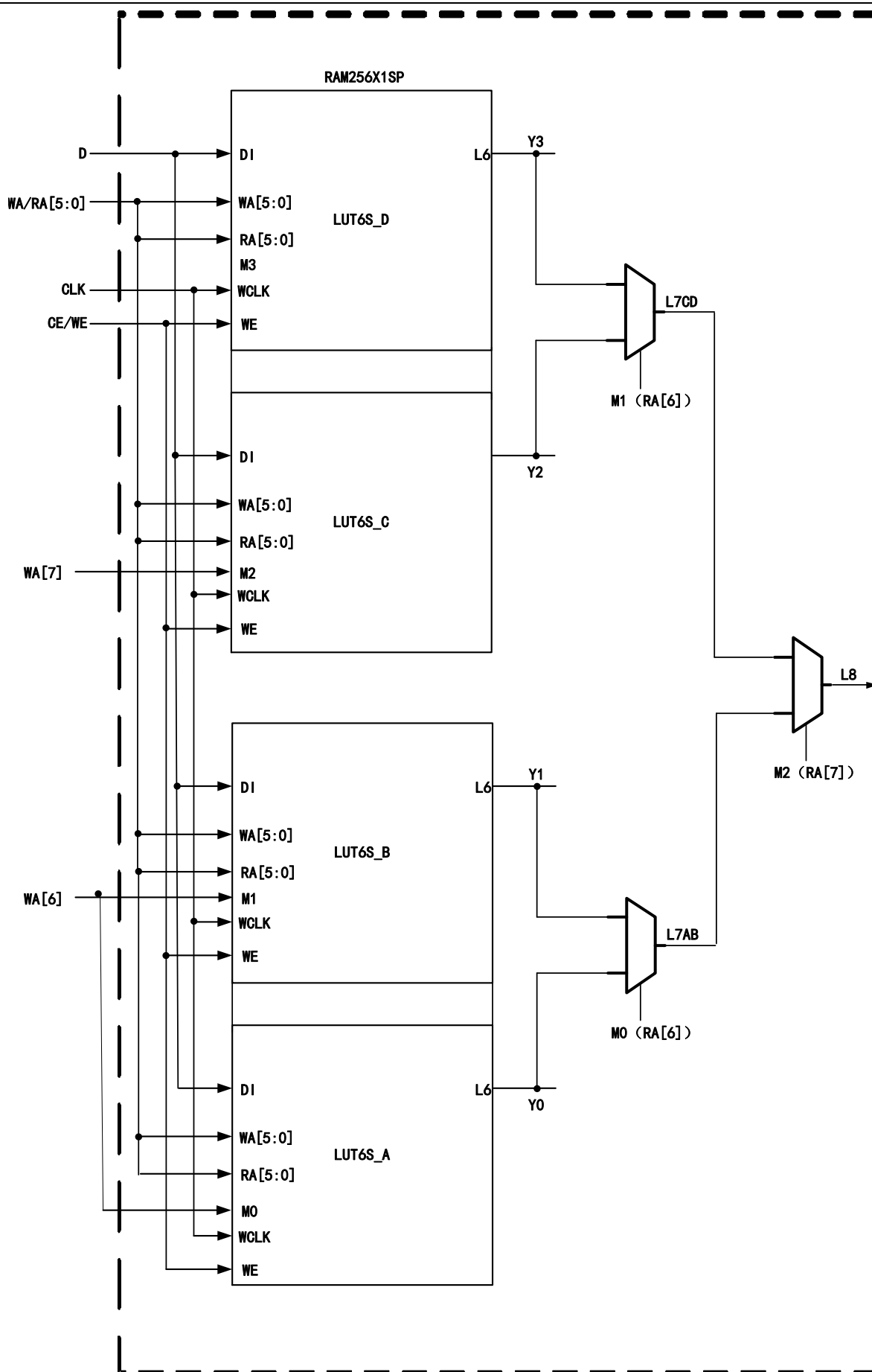


图 2-10 RAM256X1SP 实现方式逻辑示意图

### 2.2.5.8 RAM32X2SP

以 LUT6S\_A 为例实现一个 RAM32X2SP，写地址复用 LUT6S\_D 的输入 D[4:0]，读地址是 LUT6S\_A 本身的输入 A[4:0]，因为是单端口类型的 SRAM，需要把 A[4:0]与 D[4:0]连接在一起；写数据端口是{AD,M0}输入，读数据输出端口是{CR0,Y0}，写使能则是 WE 或者 CE。1 个 LUT6S 可以例化 1 个 RAM32X2SP。

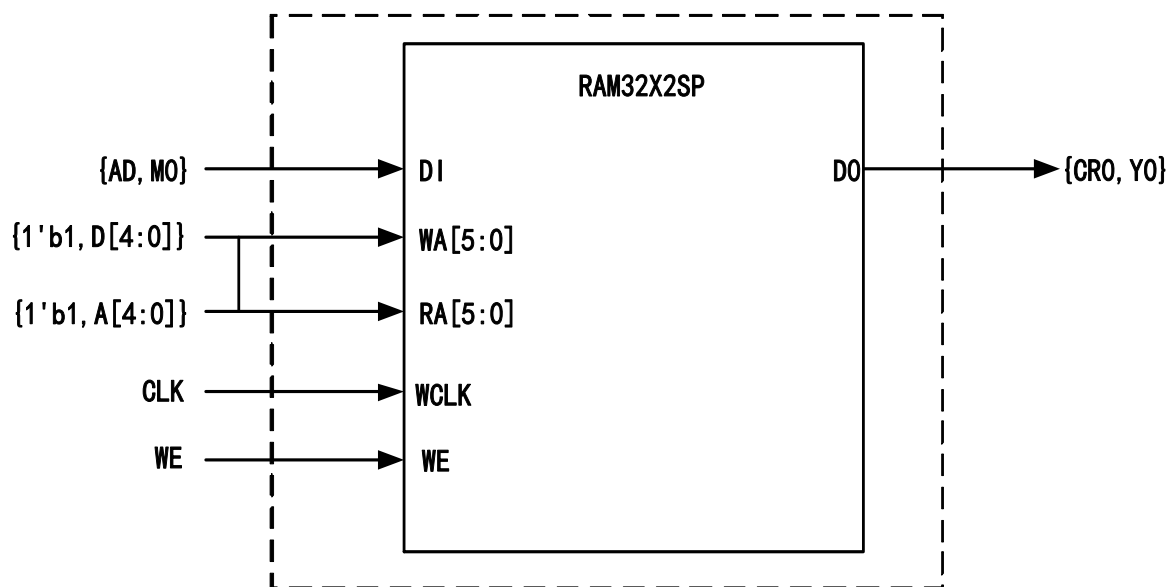


图 2-11 RAM32X2SP 实现方式逻辑示意图

### 2.2.5.9 RAM32X2DP

以 LUT6S\_A 为例实现一个 RAM32X2DP，读写地址独立，复用 LUT6S\_D 的输入端 D[4:0]作为 RAM 的写地址，读地址为 LUT6S\_A 本身的输入 A[4:0]，写数据端口为{AD,M0}输入，读数据输出端口为{CR0,Y0}，写使能为 WE 或者 CE。一个 LUT6S 可例化 1 个 RAM32X2DP。

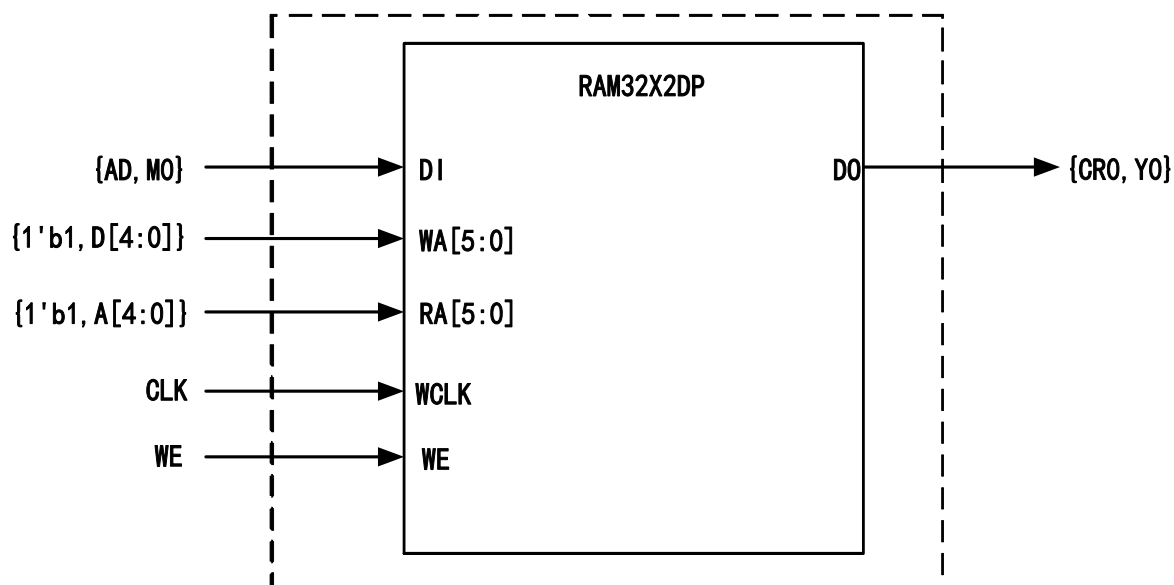


图 2-12 RAM32X2DP 实现方式逻辑示意图

## 2.2.5.10 RAM32X2X4

图 2-13 是一个 RAM32X2X4 的示意图，需要四个 LUT6S 实现。共 4 个读写通道，分别写入 4 个 LUT6S 的存储空间，LUT6S\_D 的读地址和写地址需要连接在一块，LUT6S\_A/B/C 的写地址复用 LUT6S\_D 的读写地址，一个 CLMS 可以例化 1 个 RAM32X2X4。

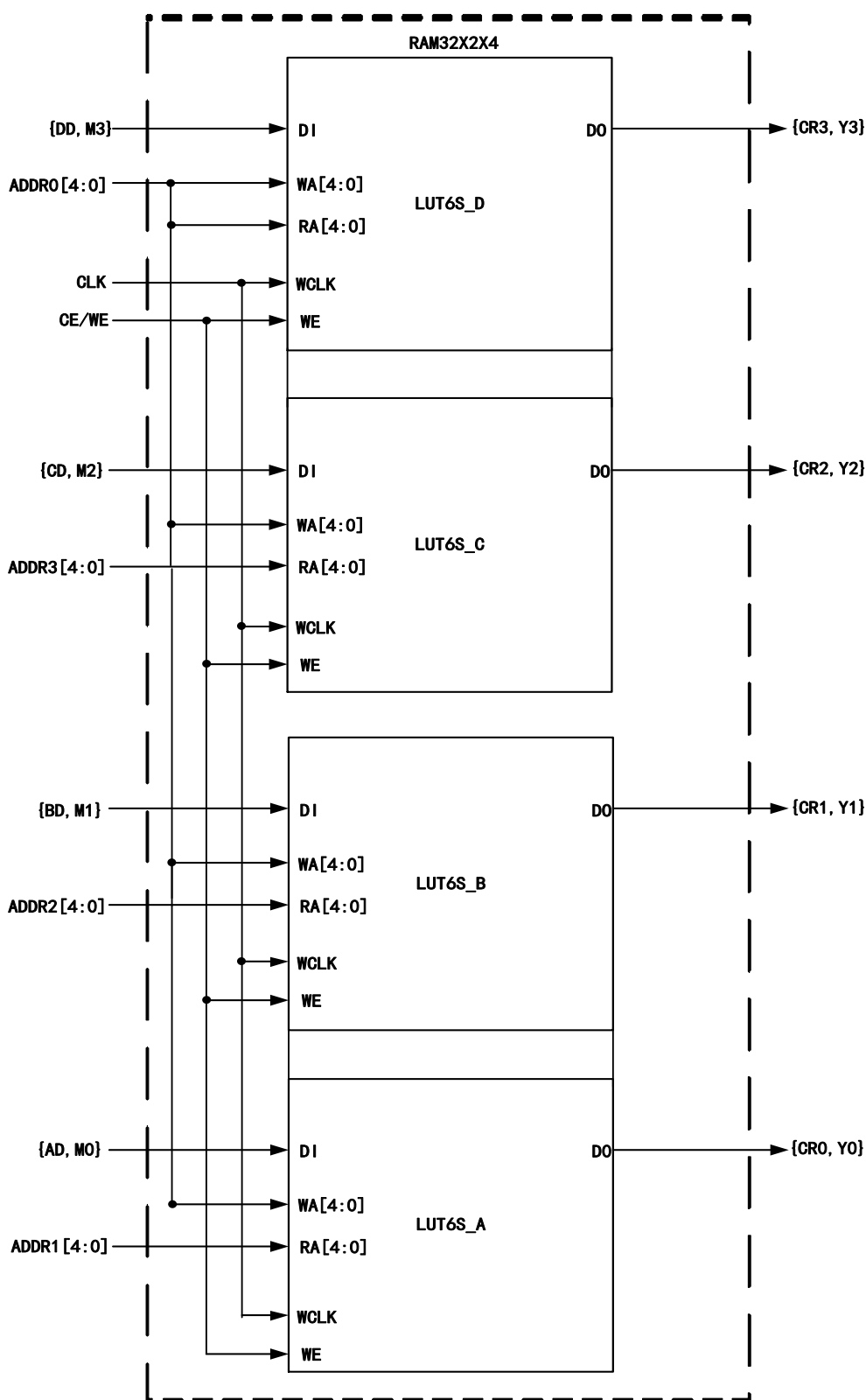


图 2-13 RAM32X2X4 实现方式逻辑示意图

### 2.2.5.11 RAM64X1X4

图 2-14 是一个 RAM64X1X4 的示意图，需要四个 LUT6S 实现。共 4 个读写通道，分别写入 4 个 LUT6S 的存储空间，LUT6S\_D 的读地址和写地址需要连接在一块，LUT6S\_A/B/C 的写地址复用 LUT6S\_D 的读写地址，一个 CLMS 可以例化 1 个 RAM64X1X4。

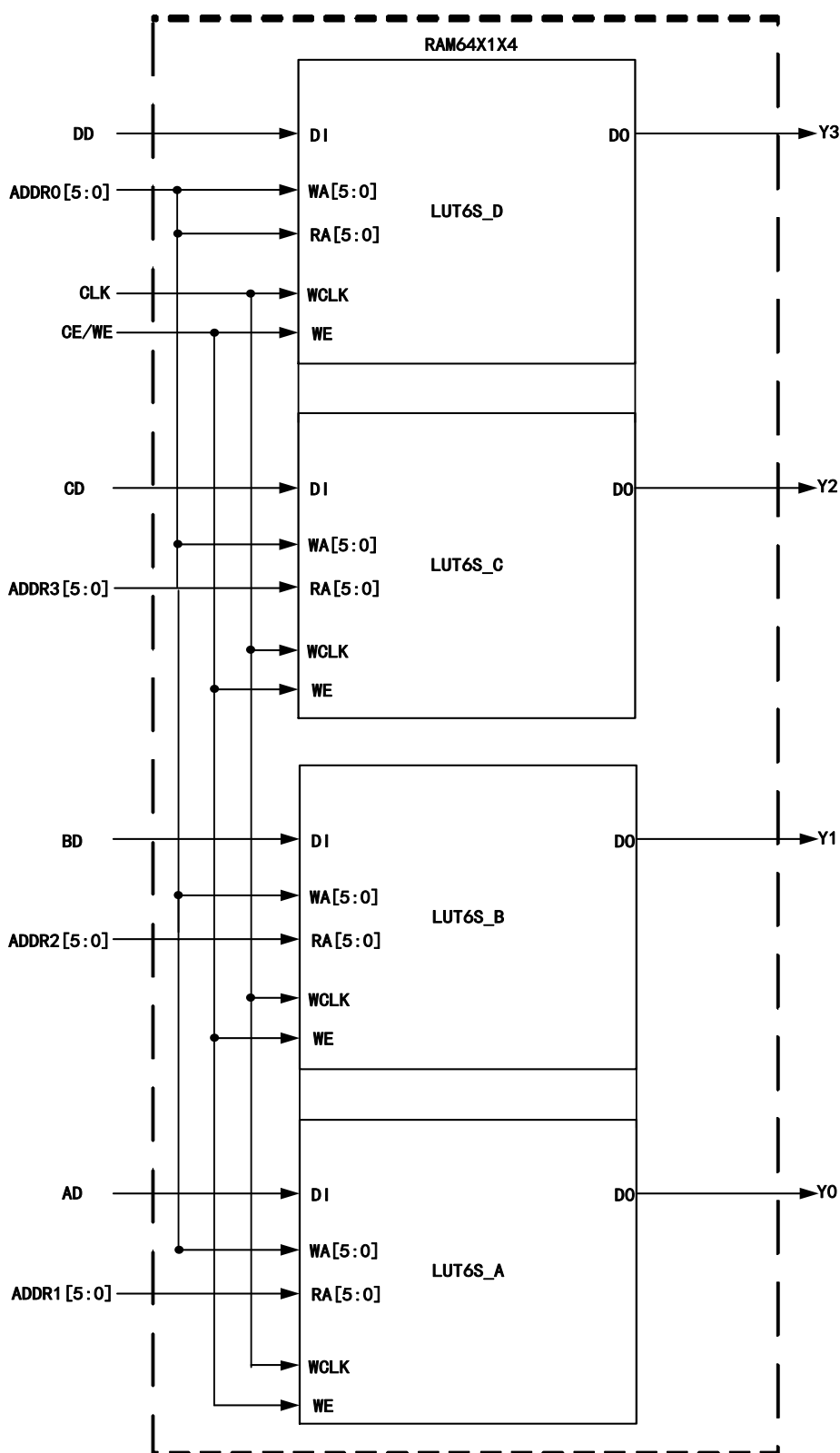


图 2-14 RAM64X1X4 实现方式逻辑示意图



## 2.3 CLM 寄存器的工作模式

Logos2 系列 FPGA 的 CLM 共有 8 个寄存器，其可配置的属性主要包括：

- 灵活的数据输入选择；
- 支持同步复位，同步置位，异步复位或异步置位模式；
- 寄存器的时钟（CLK）、时钟使能（CE）、本地复位/置位（RS）信号均支持极性选择；
- 时钟使能（CE）、本地复位/置位（RS）信号均支持快速级联链；
- 支持移位寄存器的快速级联链。

按照寄存器的数据输入来源把寄存器分成两类，4 个主寄存器以及 4 个附加寄存器。现在分别介绍这两种寄存器。

### 2.3.1 主寄存器

主寄存器主要是存储 LUT6 的输出数据。其结构如图 2-15 所示。

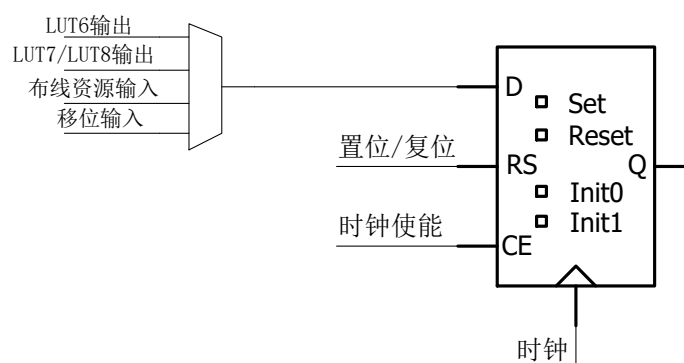


图 2-15 主寄存器示意图

主寄存器的输入源有：LUT6 输出、LUT7/LUT8 输出、布线资源输出（M0/M1/M2/M3）、移位寄存器链输出。主寄存器输出到布线资源。

主寄存器功能特性如下：

- 数据输入来自 LUT、布线资源输出、移位寄存器输出
- 可编程复位/置位（同步/异步，CLM 全局配置模式）
- 可编程时钟上升沿/下降沿触发器（CLM 全局配置模式）
- 可用于逻辑功能模式、算术功能模式、ROM 模式、分布式 RAM 模式
- 可编程时钟/时钟使能/本地复位置位控制信号极性（CLM 全局配置模式）
- 全局复位/置位(GRS)时异步初始化 0/1
- 可配置为移位寄存器（由 4 个主寄存器和 4 个附加寄存器共同组成）

### 2.3.2 附加寄存器

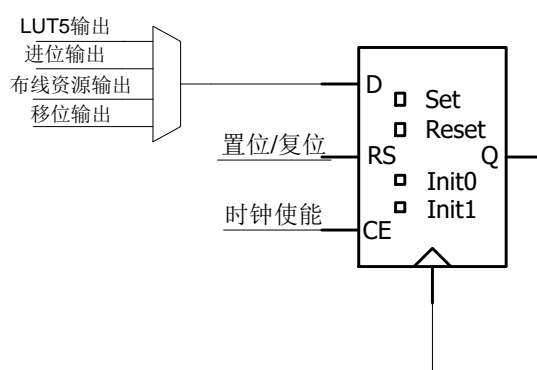


图 2-16 附加寄存器示意图

附加寄存器的输入源有：LUT5 输出、进位输出、布线资源输出（PG2L100H/PG2L200H 为 M0/M1/M2/M3，PG2L25H/PG2L50H 为 AD/BD/CD/DD）、移位寄存器链输出。附加寄存器输出到组合输出 CR，并通过它进入绕线资源。

附加寄存器功能特性如下：

- 数据输入来自 LUT、布线资源输出、快速进位链、移位寄存器输出
- 可编程复位/置位（同步/异步，CLM 全局配置模式）
- 可编程时钟上升沿/下降沿触发器（CLM 全局配置模式）
- 可用于逻辑功能模式、算术功能模式、ROM 模式、分布式 RAM 模式
- 可编程时钟/时钟使能/本地复位置位控制信号极性（CLM 全局配置模式）
- 全局复位/置位(GRS)时异步初始化 0/1
- 可配置为移位寄存器（由 4 个主寄存器和 4 个附加寄存器共同组成）

### 2.3.3 寄存器控制信号

在每个 CLM 中，所有的寄存器都共用时钟（CLK）、时钟使能（CE）、本地复位/置位（RS）信号，CLK、CE、RS 都由 CLM 全局控制极性。寄存器控制信号还具有如下特性：

- CE 端口输入可禁止，即内部时钟一直处于有效状态
- RS 端口输入可禁止，即内部寄存器本地复位/置位一直处于无效状态
- 8 个寄存器组成 8 位移位寄存器链，时钟来自本地 CLK
- CE、RS 有内建控制级联链（基于 CLM 列，自下而上）

特别的是，PG2L200H 的 CLM 结构和 Logos2 系列其他器件存在差异，PG2L200H 的 CLM 有两个时钟使能（CE）端口 CE0、CE1，分别控制 4 个寄存器，同时还新增了一个 2 选 1 的 MUX，当 8 个寄存器的 CE 信号相同时，可以从 CE0 输入信号，通过新增 MUX 把信号送到 8 个寄存器。PG2L200H 和 Logos2 系列其他器件的 CE 信号差异示意图如图 2-17 所示。

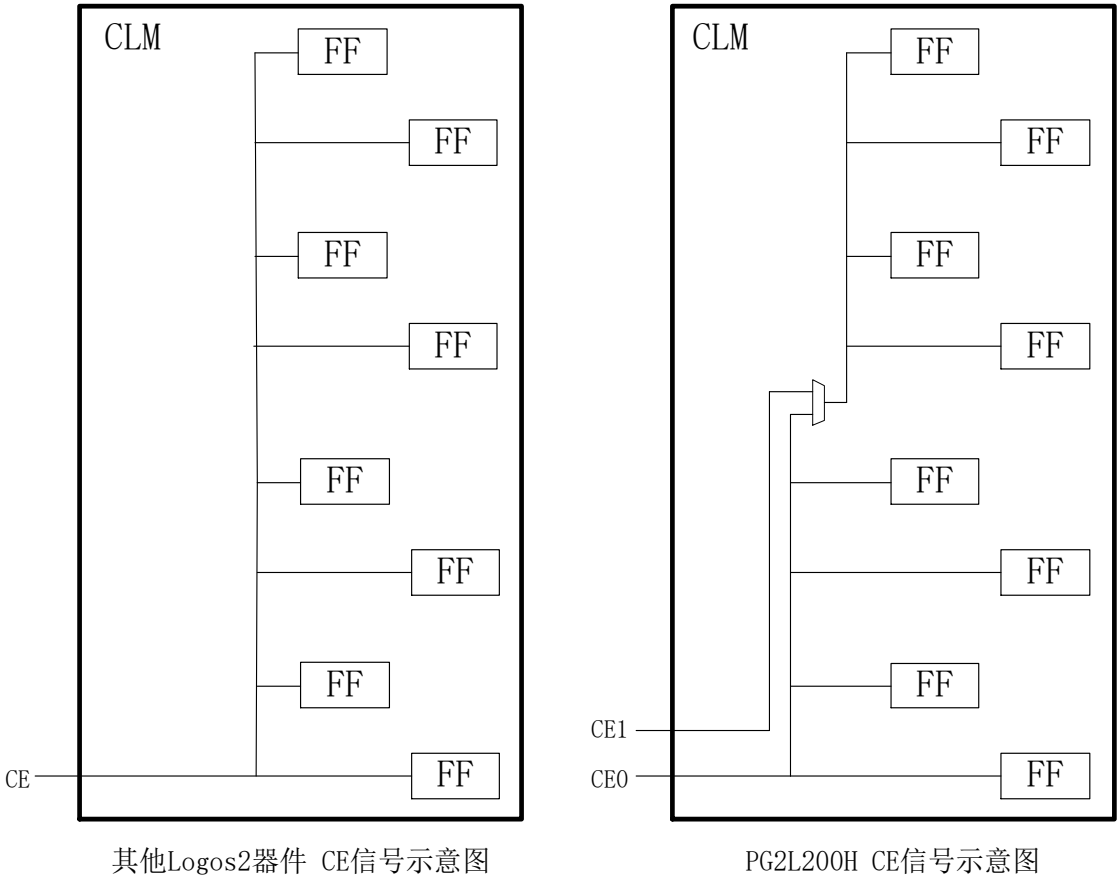


图 2-17 PG2L200H 和 Logos2 系列其他器件的 CE 信号差异示意图

2.4 CLM 时序参数

表 2-5 为 CLM 典型时序参数说明：

表 2-5 CLM 典型时序参数

| 参数      | 控制信号            | 说明                                      |
|---------|-----------------|---|
| Tceck   | CE              | CE 相对于 DFF 的 setup                      |
| Tsu/Thd | Ax/Bx/Cx/Dx     | Ax/Bx/Cx/Dx 相对于 DFF 的 setup/hold        |
| Titoy6  | Y0/Y1/Y2/Y3     | LUT6 输入 Ax/Bx/Cx/Dx 到 Y0/Y1/Y2/Y3 delay |
| Trsck   | RS              | RS 相对于 DFF 的 setup                      |
| Tco     | Q0/Q1/Q2/Q3     | CLK 输入相对于 Q0/Q1/Q2/Q3 的 TCO             |
| Tctocr  | CR0/CR1/CR2/CR3 | CLK 输入相对于 CR0/CR1/CR2/CR3 的 TCO         |

注：详细的时序参数说明见《DS04001\_Logos2 系列 FPGA 器件数据手册》。

图 2-18 为 CLM 的时序图。

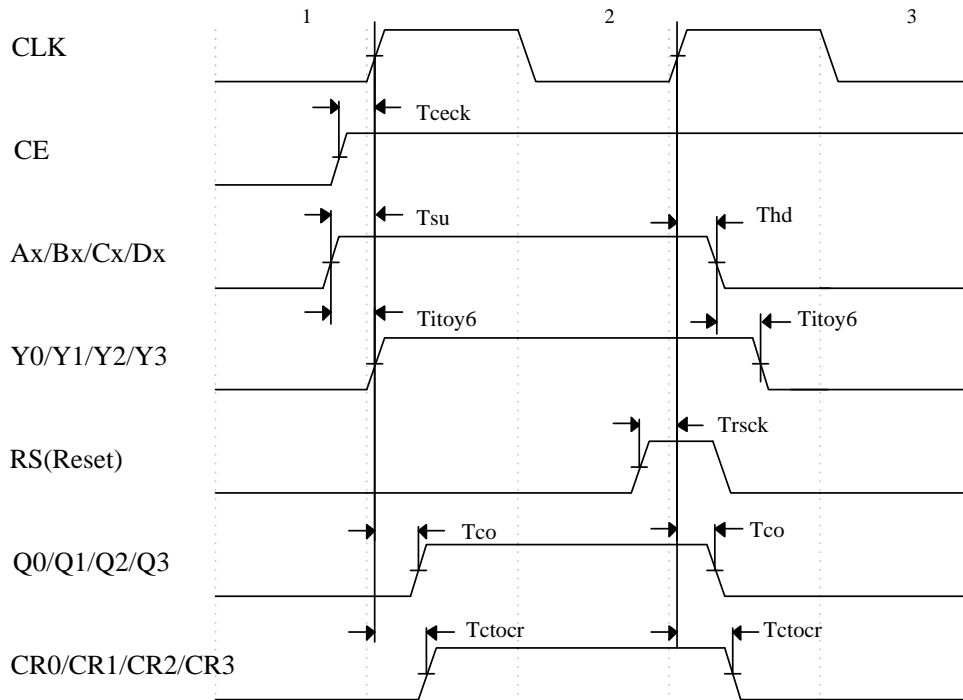


图 2-18 CLM 时序图

注：x = 0,1,2,3,4,5。

## 3 应用示例

### 3.1 常用 GTP

本节主要介绍几种常用的 CLM 的 GTP 原语，用户可以在设计中直接例化调用，各 GTP 的支持情况、端口、参数等以《UG040007\_Logos2 系列产品 GTP 用户指南》为准。

#### 3.1.1 多路选择器 GTP

表 3-1 多路选择器 GTP

| GTP          | 输入来源                  | 硬件资源              | 描述                      |
|--------------|-----------------------|-------------------|-------------------------|
| GTP_MUX2LUT7 | LUT6 的输出              | L7ABMUX 或 L7CDMUX | 用于构建 LUT7、8:1 多路选择器等逻辑  |
| GTP_MUX2LUT8 | L7ABMUX 和 L7CDMUX 的输出 | L8MUX             | 用于构建 LUT8、16:1 多路选择器等逻辑 |

GTP\_MUX2LUT7 和 GTP\_MUX2LUT8 的端口相同，示意图如图 3-1 所示。

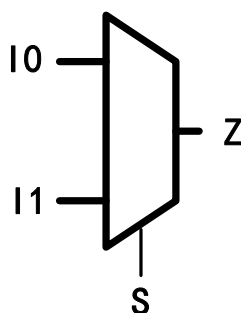


图 3-1 多路选择器 GTP 端口示意图

多路选择器的端口描述如表 3-2 所示。

表 3-2 多路选择器的端口描述

| Port | 输入/输出 | 功能描述 |
|------|-------|------|
| I0   | 输入    | 输入信号 |
| I1   | 输入    | 输入信号 |
| S    | 输入    | 选择信号 |
| Z    | 输出    | 输出信号 |

多路选择器的真值表如表 3-3 所示（X 表示任意值）。当选择信号 S 为 1'b0 时输出为 I0，S 为 1'b1 时输出为 I1。

表 3-3 多路选择器的真值表

| 输入 |    |   | 输出 |
|----|----|---|----|
| I1 | I0 | S | Z  |
| X  | 0  | 0 | 0  |
| X  | 1  | 0 | 1  |
| 0  | X  | 1 | 0  |
| 1  | X  | 1 | 1  |

### 3.1.2 进位链 GTP

进位链原语 GTP\_LUT6CARRY 表征的是 CLM 中的快速进位逻辑，构建算术逻辑如加法器、乘法器等常用到此 GTP（详见 2.2 LUT6A 和 LUT6S 的工作模式）。

GTP\_LUT6CARRY 是一个快速进位逻辑，不同的 INIT 参数值对应不同的功能，使用时必须指定对应的 INIT 参数。

表 3-4 进位链 GTP 参数说明

| 参数名称        | 参数类型     | 有效值               | 功能描述   |
|-------------|----------|-------------------|--|
| INIT        | <binary> | 0~2^64-1          | 输出配置参数   |
| I5_TO_CARRY | <string> | “TRUE” or “FALSE” | “TRUE”：选择 I5 输入到进位逻辑；<br>“FALSE”：选择 LUT5B 输入到进位逻辑。     |
| I5_TO_LUT   | <string> | “TRUE” or “FALSE” | “TRUE”：选择 I5 输入到 LUT 逻辑；<br>“FALSE”：选择 CIN 输入到 LUT 逻辑。 |

GTP\_LUT6CARRY 原语的端口示意图如图 3-2 所示。

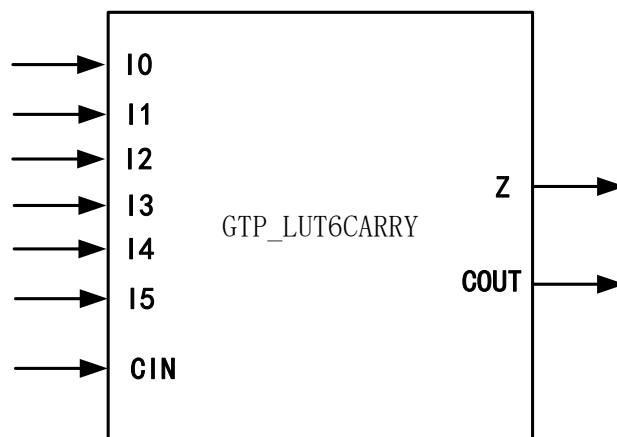


图 3-2 进位链 GTP 端口示意图

GTP\_LUT6CARRY 原语的端口描述见表 3-5 所示。

表 3-5 进位链 GTP 端口描述

| Port | 输入/输出 | 功能描述 |
|------|-------|------|
| I0   | 输入    | 输入信号 |
| I1   | 输入    | 输入信号 |
| I2   | 输入    | 输入信号 |

| Port | 输入/输出 | 功能描述       |
|------|-------|------------|
| I3   | 输入    | 输入信号       |
| I4   | 输入    | 输入信号       |
| I5   | 输入    | 输入信号       |
| CIN  | 输入    | 输入信号，级联输入。 |
| COUT | 输出    | 级联输出       |
| Z    | 输出    | 输出信号       |

GTP\_LUT6CARRY 原语的内部结构示意图如图 3-3 所示，GTP\_LUT6CARRY 由两个 lut5 和 4 个二输入数据选择器 mux 组成，其中，mux0 和 mux2 由内部信号动态选择输出到 COUT 和 Z，mux1 和 mux3 则由参数 I5\_TO\_CARRY、I5\_TO\_LUT 进行配置静态选择固定的内部信号。

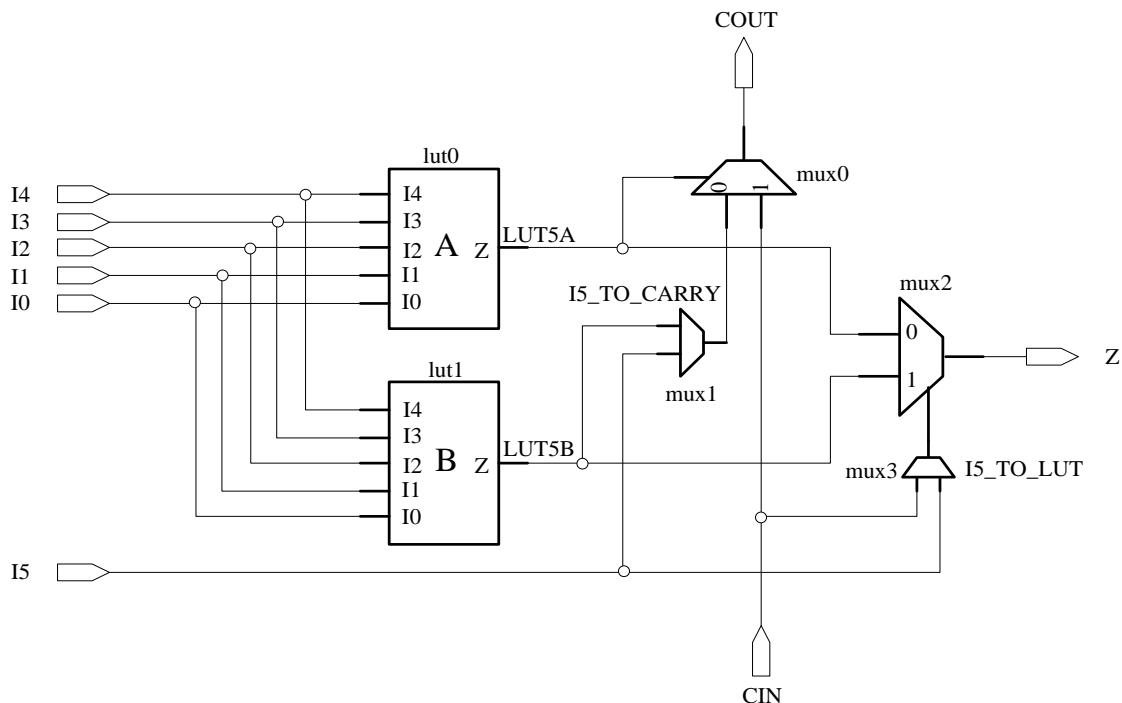


图 3-3 GTP\_LUT6CARRY 内部结构示意图

GTP\_LUT6CARRY 的真值表见表 3-6、表 3-7 和表 3-8。

表 3-6 内部信号 LUT5A、LUT5B 真值表

| 输入信号 |    |    |    |    | 输出信号     |         |
|------|----|----|----|----|----------|---------|
| I4   | I3 | I2 | I1 | I0 | LUT5B    | LUT5A   |
| 0    | 0  | 0  | 0  | 0  | INIT[32] | INIT[0] |
| 0    | 0  | 0  | 0  | 1  | INIT[33] | INIT[1] |
| 0    | 0  | 0  | 1  | 0  | INIT[34] | INIT[2] |
| 0    | 0  | 0  | 1  | 1  | INIT[35] | INIT[3] |
| 0    | 0  | 1  | 0  | 0  | INIT[36] | INIT[4] |
| 0    | 0  | 1  | 0  | 1  | INIT[37] | INIT[5] |

| 输入信号 |    |    |    |    | 输出信号     |          |
|------|----|----|----|----|----------|----------|
| I4   | I3 | I2 | I1 | I0 | LUT5B    | LUT5A    |
| 0    | 0  | 1  | 1  | 0  | INIT[38] | INIT[6]  |
| 0    | 0  | 1  | 1  | 1  | INIT[39] | INIT[7]  |
| 0    | 1  | 0  | 0  | 0  | INIT[40] | INIT[8]  |
| 0    | 1  | 0  | 0  | 1  | INIT[41] | INIT[9]  |
| 0    | 1  | 0  | 1  | 0  | INIT[42] | INIT[10] |
| 0    | 1  | 0  | 1  | 1  | INIT[43] | INIT[11] |
| 0    | 1  | 1  | 0  | 0  | INIT[44] | INIT[12] |
| 0    | 1  | 1  | 0  | 1  | INIT[45] | INIT[13] |
| 0    | 1  | 1  | 1  | 0  | INIT[46] | INIT[14] |
| 0    | 1  | 1  | 1  | 1  | INIT[47] | INIT[15] |
| 1    | 0  | 0  | 0  | 0  | INIT[48] | INIT[16] |
| 1    | 0  | 0  | 0  | 1  | INIT[49] | INIT[17] |
| 1    | 0  | 0  | 1  | 0  | INIT[50] | INIT[18] |
| 1    | 0  | 0  | 1  | 1  | INIT[51] | INIT[19] |
| 1    | 0  | 1  | 0  | 0  | INIT[52] | INIT[20] |
| 1    | 0  | 1  | 0  | 1  | INIT[53] | INIT[21] |
| 1    | 0  | 1  | 1  | 0  | INIT[54] | INIT[22] |
| 1    | 0  | 1  | 1  | 1  | INIT[55] | INIT[23] |
| 1    | 1  | 0  | 0  | 0  | INIT[56] | INIT[24] |
| 1    | 1  | 0  | 0  | 1  | INIT[57] | INIT[25] |
| 1    | 1  | 0  | 1  | 0  | INIT[58] | INIT[26] |
| 1    | 1  | 0  | 1  | 1  | INIT[59] | INIT[27] |
| 1    | 1  | 1  | 0  | 0  | INIT[60] | INIT[28] |
| 1    | 1  | 1  | 0  | 1  | INIT[61] | INIT[29] |
| 1    | 1  | 1  | 1  | 0  | INIT[62] | INIT[30] |
| 1    | 1  | 1  | 1  | 1  | INIT[63] | INIT[31] |

表 3-7 输出信号 COUT 真值表

| 参数          | 输入信号  |       |    |     | 输出信号 |
|-------------|-------|-------|----|-----|------|
| I5_TO_CARRY | LUT5A | LUT5B | I5 | CIN | COUT |
| “FALSE”     | 0     | 0     | X  | 0   | 0    |
| “FALSE”     | 0     | 0     | X  | 1   | 0    |
| “FALSE”     | 0     | 1     | X  | 0   | 1    |
| “FALSE”     | 0     | 1     | X  | 1   | 1    |
| “FALSE”     | 1     | 0     | X  | 0   | 0    |
| “FALSE”     | 1     | 0     | X  | 1   | 1    |
| “FALSE”     | 1     | 1     | X  | 0   | 0    |
| “FALSE”     | 1     | 1     | X  | 1   | 1    |



| 参数          | 输入信号  |       |    |     | 输出信号 |
|-------------|-------|-------|----|-----|------|
| I5_TO_CARRY | LUT5A | LUT5B | I5 | CIN | COUT |
| “TRUE”      | 0     | X     | 0  | 0   | 0    |
| “TRUE”      | 0     | X     | 0  | 1   | 0    |
| “TRUE”      | 0     | X     | 1  | 0   | 1    |
| “TRUE”      | 0     | X     | 1  | 1   | 1    |
| “TRUE”      | 1     | X     | 0  | 0   | 0    |
| “TRUE”      | 1     | X     | 0  | 1   | 1    |
| “TRUE”      | 1     | X     | 1  | 0   | 0    |
| “TRUE”      | 1     | X     | 1  | 1   | 1    |

表 3-8 输出信号 Z 真值表

| 参数        | 输入信号 |     |       |       | 输出信号 |
|-----------|------|-----|-------|-------|------|
| I5_TO_LUT | I5   | CIN | LUT5A | LUT5B | Z    |
| “FALSE”   | X    | 0   | 0     | 0     | 0    |
| “FALSE”   | X    | 0   | 0     | 1     | 0    |
| “FALSE”   | X    | 0   | 1     | 0     | 1    |
| “FALSE”   | X    | 0   | 1     | 1     | 1    |
| “FALSE”   | X    | 1   | 0     | 0     | 0    |
| “FALSE”   | X    | 1   | 0     | 1     | 1    |
| “FALSE”   | X    | 1   | 1     | 0     | 0    |
| “FALSE”   | X    | 1   | 1     | 1     | 1    |
| “TRUE”    | 0    | X   | 0     | 0     | 0    |
| “TRUE”    | 0    | X   | 0     | 1     | 0    |
| “TRUE”    | 0    | X   | 1     | 0     | 1    |
| “TRUE”    | 0    | X   | 1     | 1     | 1    |
| “TRUE”    | 1    | X   | 0     | 0     | 0    |
| “TRUE”    | 1    | X   | 0     | 1     | 1    |
| “TRUE”    | 1    | X   | 1     | 0     | 0    |
| “TRUE”    | 1    | X   | 1     | 1     | 1    |

### 3.1.3 分布式 RAM GTP

要使用 CLMS 中的分布式 RAM 资源可以采用例化分布式 RAM 原语的方式，Logos2 系列 FPGA 支持如表 3-9 所示的分布式 RAM GTP。

表 3-9 RAM GTP

| GTP           | 容量     | 类型    | 地址输入                                |
|---------------|--------|-------|-------------------------------------|
| GTP_RAM32X1SP | 32-bit | 单端口   | A0~4/B0~4/C0~4/D0~4（write/read）     |
| GTP_RAM32X1DP | 32-bit | 简单双端口 | D0~4（write）<br>A0~4/B0~4/C0~4(read） |



| 端口名  | 输入/输出 | 说明  |
|------|-------|---|
|      |       | ADDR: 地址信号, 读写共用;<br>简单双端口:<br>WADDR: 写地址<br>RADDR: 读地址 |
| WCLK | 输入    | 时钟信号  |
| WE   | 输入    | 写使能信号, 高有效  |

Logos2 系列 FPGA 支持的分布式单端口 RAM 的端口时序图如图 3-5 所示, 图中 MEM 指对应地址存储的数据。分布式单端口 RAM 是同步写异读的读写方式, 在第 3 个时钟脉冲的上升沿写入数据之前, 地址已经跳变为 A0, DO 由于地址跳变而输出旧数据 MEM(A0); 数据 D0 写入 RAM 的 A0 地址, 经过一定延时后输出到 DO 端; 地址跳变为 A1 时, WE 为 0, 所以 DO 输出 MEM(A1)。

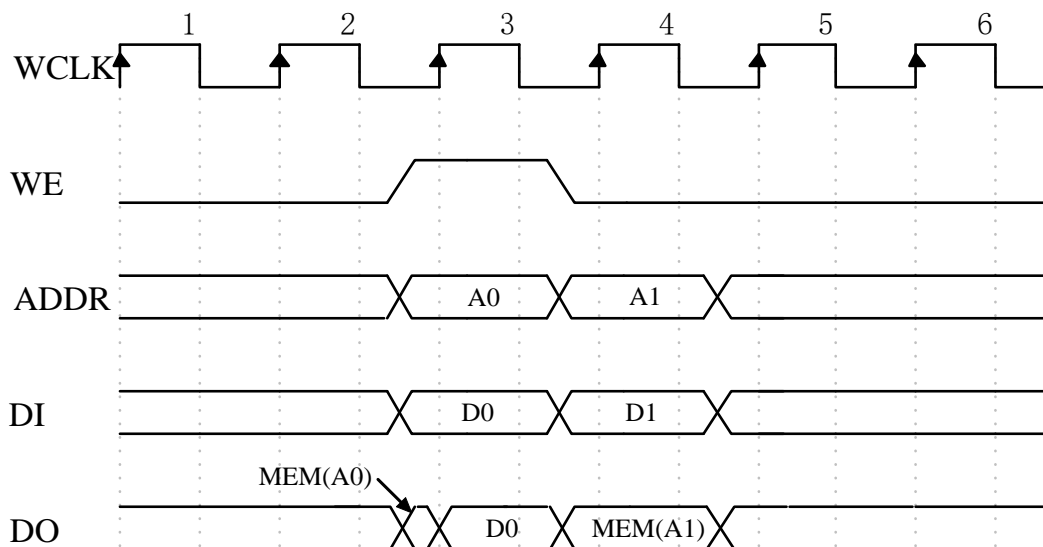


图 3-5 分布式单端口 RAM 时序图

Logos2 系列 FPGA 支持的分布式简单双端口 RAM 的端口时序图如图 3-6 所示, 图中 MEM 指对应地址存储的数据。RAM 是同步写异读的写方式, 在第 3 和第 4 个时钟周期中, WE 为高, 在时钟上升沿分别把数据 D0 和 D1 写入地址 A0 和 A1; 在读地址 RADDR 跳变为 A0 和 A1 时, 在 DO 分别读出数据 D0 和 D1。

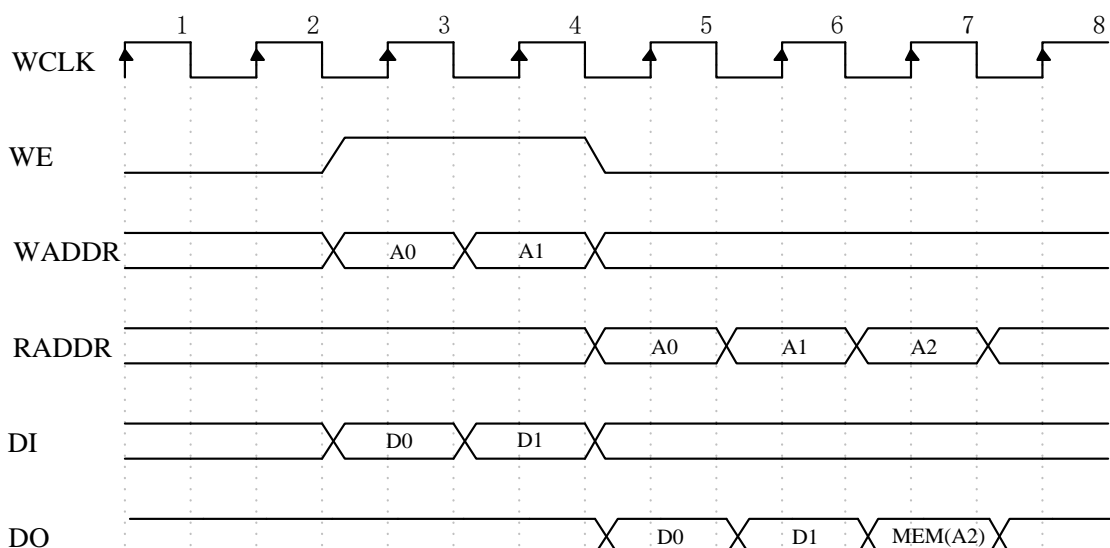


图 3-6 分布式简单双口 RAM 时序图

### 3.1.4 ROM GTP

要使用 CLM 作为只读存储器可以采用例化 ROM 原语的方式，Logos2 系列 FPGA 支持如表 3-12 所示的 ROM GTP。

表 3-12 ROM GTP

| GTP          | 容量      | 地址输入   |
|--------------|---------|--|
| GTP_ROM32X1  | 32-bit  | A0~4/B0~4/C0~4/D0~4  |
| GTP_ROM32X2  | 64-bit  | A0~4/B0~4/C0~4/D0~4  |
| GTP_ROM64X1  | 64-bit  | A0~5/B0~5/C0~5/D0~5  |
| GTP_ROM128X1 | 128-bit | A0~5/B0~5/C0~5/D0~5 作为低 6 位地址, M0/M1 作为最高位地址               |
| GTP_ROM256X1 | 256-bit | A0~5/B0~5/C0~5/D0~5 作为低 6 位地址, M0/M1 作为第 7 位地址, M2 作为最高位地址 |

分布式 ROM 可以设置初始值参数，以 256bit ROM 为例，见表 3-13。

表 3-13 分布式 ROM GTP 参数说明

[illegible]

以 GTP\_ROM256X1 为例，ROM GTP 的端口示意图如图 3-7 所示。

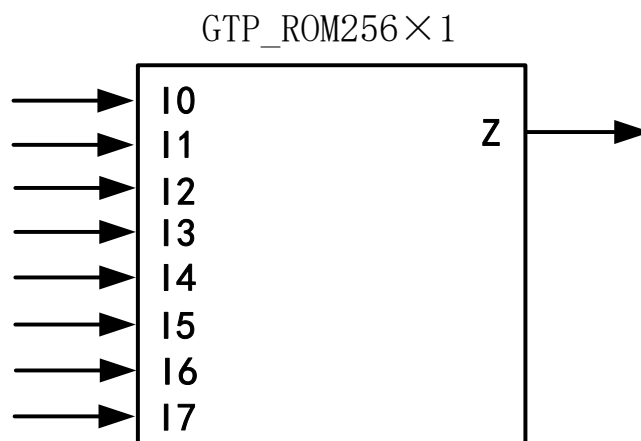


图 3-7 ROM GTP 端口示意图

GTP\_ROM256X1 的端口描述见表 3-14。

表 3-14 分布式 ROM GTP 端口说明

| Port | 输入/输出 | 功能描述            |
|------|-------|-----------------|
| I0   | 输入    | ROM 读地址 addr[0] |
| I1   | 输入    | ROM 读地址 addr[1] |
| I2   | 输入    | ROM 读地址 addr[2] |
| I3   | 输入    | ROM 读地址 addr[3] |
| I4   | 输入    | ROM 读地址 addr[4] |
| I5   | 输入    | ROM 读地址 addr[5] |
| I6   | 输入    | ROM 读地址 addr[6] |
| I7   | 输入    | ROM 读地址 addr[7] |
| Z    | 输出    | 读出数据            |

该 GTP 实现存储 ROM 功能。输入 I7~I0 构成读取数据地址，读取 ROM 初始配置参数的指定位的数值。

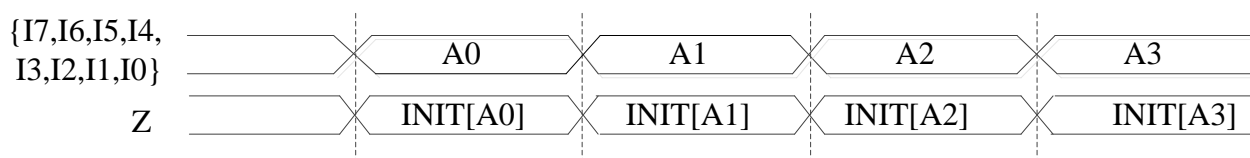


图 3-8 GTP\_ROM256X1 波形示意图

其他分布式 ROM GTP 说明详见《UG040007\_Logos2 系列产品 GTP 用户指南》。

### 3.1.5 触发器 GTP

Logos2 系列 FPGA 支持如表 3-15 所示的触发器 GTP。

表 3-15 触发器 GTP

| GTP        | 描述           |
|------------|--------------|
| GTP_DFF    | 触发器          |
| GTP_DFF_C  | 异步清零触发器      |
| GTP_DFF_E  | 带使能端的触发器     |
| GTP_DFF_CE | 带使能端的异步清零触发器 |
| GTP_DFF_P  | 异步置位触发器      |
| GTP_DFF_PE | 带使能端的异步置位触发器 |
| GTP_DFF_R  | 同步清零触发器      |
| GTP_DFF_RE | 带使能端的同步清零触发器 |
| GTP_DFF_S  | 同步置位触发器      |
| GTP_DFF_SE | 带使能端的同步置位触发器 |

以 GTP\_DFF\_CE 为例，触发器 GTP 的端口示意图如图 3-9 所示。

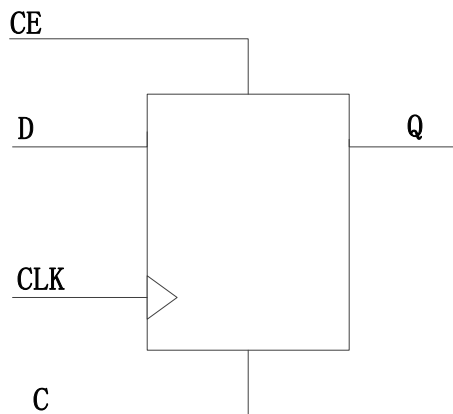


图 3-9 GTP\_DFF\_CE 端口示意图

Logos2 系列 FPGA 支持的触发器 GTP 的端口说明如表 3-16 所示。

表 3-16 触发器 GTP 端口描述

| 端口名 | 输入/输出 | 说明                   |
|-----|-------|----------------------|
| CLK | 输入    | 输入时钟                 |
| D   | 输入    | 触发器输入信号              |
| Q   | 输出    | 触发器输出信号              |
| C   | 输入    | 异步清零信号，高电平时使输出 Q 置 0 |
| CE  | 输入    | 使能信号，高有效             |
| P   | 输入    | 异步置位信号，高电平时使输出 Q 置 1 |
| R   | 输入    | 同步清零信号，高电平时使输出 Q 置 0 |

| 端口名 | 输入/输出 | 说明                   |
|-----|-------|----------------------|
| S   | 输入    | 同步置位信号，高电平时使输出 Q 置 1 |

以 GTP\_DFF\_CE 为例，触发器 GTP 的参数说明如表 3-17 所示。

表 3-17 GTP\_DFF\_CE 参数描述

| 参数名称   | 参数类型     | 设置值             | 功能描述                              |
|--------|----------|-----------------|-----------------------------------|
| GRS_EN | <string> | "TRUE", "FALSE" | 全局复位信号使能                          |
| INIT   | <binary> | 1'b0, 1'b1      | 初值，当 GRS_EN = “TRUE” 时由全局复位信号异步触发 |

其他触发器 GTP 说明详见《UG040007\_Logos2 系列产品 GTP 用户指南》。

### 3.1.6 锁存器 GTP

Logos2 系列 FPGA 支持如表 3-18 所示的锁存器 GTP。

表 3-18 锁存器 GTP

| GTP           | 描述           |
|---------------|--------------|
| GTP_DLATCH    | 锁存器          |
| GTP_DLATCH_C  | 异步清零锁存器      |
| GTP_DLATCH_E  | 带使能端的锁存器     |
| GTP_DLATCH_CE | 带使能端的异步清零锁存器 |
| GTP_DLATCH_P  | 异步置位锁存器      |
| GTP_DLATCH_PE | 带使能端的异步置位锁存器 |

以 GTP\_DLATCH\_CE 为例，锁存器 GTP 的端口示意图如图 3-10 所示。

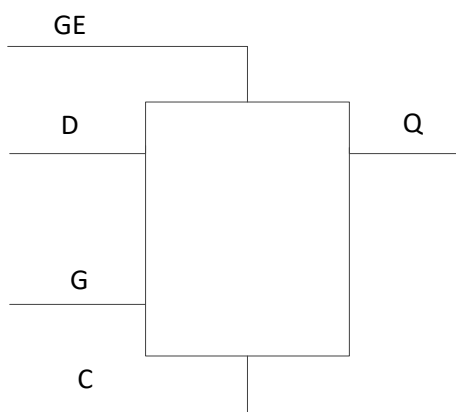


图 3-10 GTP\_DLATCH\_CE 端口示意图

Logos2 系列 FPGA 支持的锁存器 GTP 的端口说明如表 3-19 所示。

表 3-19 锁存器 GTP 端口描述

| 端口名 | 输入/输出 | 说明                          |
|-----|-------|-----------------------------|
| G   | 输入    | 输入电平，在 G 为高电平时，把输入信号传递给输出信号 |
| D   | 输入    | 锁存器输入信号                     |
| Q   | 输出    | 锁存器输出信号                     |
| P   | 输入    | 异步置位信号，高电平时使输出 Q 置 1        |
| GE  | 输入    | 使能信号，高有效                    |
| C   | 输入    | 异步清零信号，高电平时使输出 Q 置 0        |

以 GTP\_DLATCH\_CE 为例，锁存器 GTP 的参数说明如表 3-20 所示。

表 3-20 GTP\_DLATCH\_CE 参数描述

| 参数名称   | 参数类型     | 设置值             | 功能描述                              |
|--------|----------|-----------------|-----------------------------------|
| GRS_EN | <string> | "TRUE", "FALSE" | 全局复位信号使能                          |
| INIT   | <binary> | 1'b0, 1'b1      | 初值，当 GRS_EN = “TRUE” 时由全局复位信号异步触发 |

其他锁存器 GTP 说明详见《UG040007\_Logos2 系列产品 GTP 用户指南》。

## 3.2 CLM 的使用

CLM 的使用可通过深圳市紫光同创电子有限公司的软件 Pango Design Suite 来完成。可以以如下三种途径来使用 CLM。

- 通过 Pango Design Suite 内嵌的 IP Compiler 工具生成分布式 RAM IP，详见 IP Compiler 自带的文档《UG 061001 Distributed RAM IP 用户指南》；
- 通过在设计中调用 GTP 原语来使用 CLM，常用的 CLM 原语的介绍见常用 GTP 小节，也可以通过查阅 GTP 手册做详细了解，在软件安装目录的 arch\vendor\pango\verilog\simulation 子目录下，有各 GTP 的仿真模型，供用户做设计时参考；
- 用户可以在设计代码中添加属性，用来约束工具把相应的 instance 映射到 CLM。

对于实现数据存储的设计目的，可以分别用分布式存储器、块存储器、寄存器等资源来实现用户的设计，可以通过在代码中加参数来告诉工具设计者期望以哪种资源来实现设计。详细描述见软件安装目录下说明文档《ADS\_Synthesis\_User\_Guide》。

以一个 128×1 的 RAM 为例，约束对象为 module 的场景见如下代码。

```
module ipm_distributed_spram_v1_2_ram128x1
```



```
#(
parameter ADDR_WIDTH  = 4 ,
parameter DATA_WIDTH  = 4 ,
.....
)
(
    input    wire  [DATA_WIDTH-1:0] wr_data ,
    input    wire  [ADDR_WIDTH-1:0] addr ,
    .....
)/* synthesis syn_ramstyle = "select_ram" */;
reg  [DATA_WIDTH-1:0]    mem [2**ADDR_WIDTH-1:0];
.....
endmodule
```

以一个  $128 \times 1$  的 RAM 为例，约束对象为 ram signal 的应用场景见如下代码。

```
module ipm_distributed_spram_v1_2_ram128x1
#(
parameter ADDR_WIDTH  = 4 ,
parameter DATA_WIDTH  = 4 ,
.....
)
(
    input    wire  [DATA_WIDTH-1:0]          wr_data ,
    input    wire  [ADDR_WIDTH-1:0]          addr ,
    .....
);
reg  [DATA_WIDTH-1:0] mem [2**ADDR_WIDTH-1:0]/* synthesis syn_ramstyle = "select_ram" */;
.....
endmodule
```

## 免责声明

### 版权声明

本文档版权归深圳市紫光同创电子有限公司所有，并保留一切权利。未经书面许可，任何公司和个人不得将此文档中的任何部分公开、转载或以其他方式披露、散发给第三方。否则，公司必将追究其法律责任。

### 免责声明

- 1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。
- 2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。
- 3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。