Compa Family CPLDs Configurable Logic Module (CLM) User Guide

(UG030001, V1.3)

(16.08.2023)

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.3	16.08.2023	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
CLM	Configurable Logic Module
SP	Single Port
SDP	Simple Dual Port
LUT	Look Up Table

Related Documentation

The following documentation is related to this manual:

1. UG030008_Compa Family GTP User Guide

Table of Contents

Tables

Figures

Chapter 1 General Introduction

CLM (Configurable Logic Module) is the basic logic unit of Compa Family products, mainly composed of multi function LUT5, registers, expansion function selectors, fast carry logic, and cascade chains. CLM comes in two forms: CLMA and CLMS. Both CLMA and CLMS support logic, arithmetic, shift registers, and Distributed ROM functions, but only CLMS supports the Distributed RAM function. CLMs are interconnected with each other and with other on-die resources through the Signal Relay Block (SRB). The CLM Overall Block Diagram is presented in Figure 1-1.

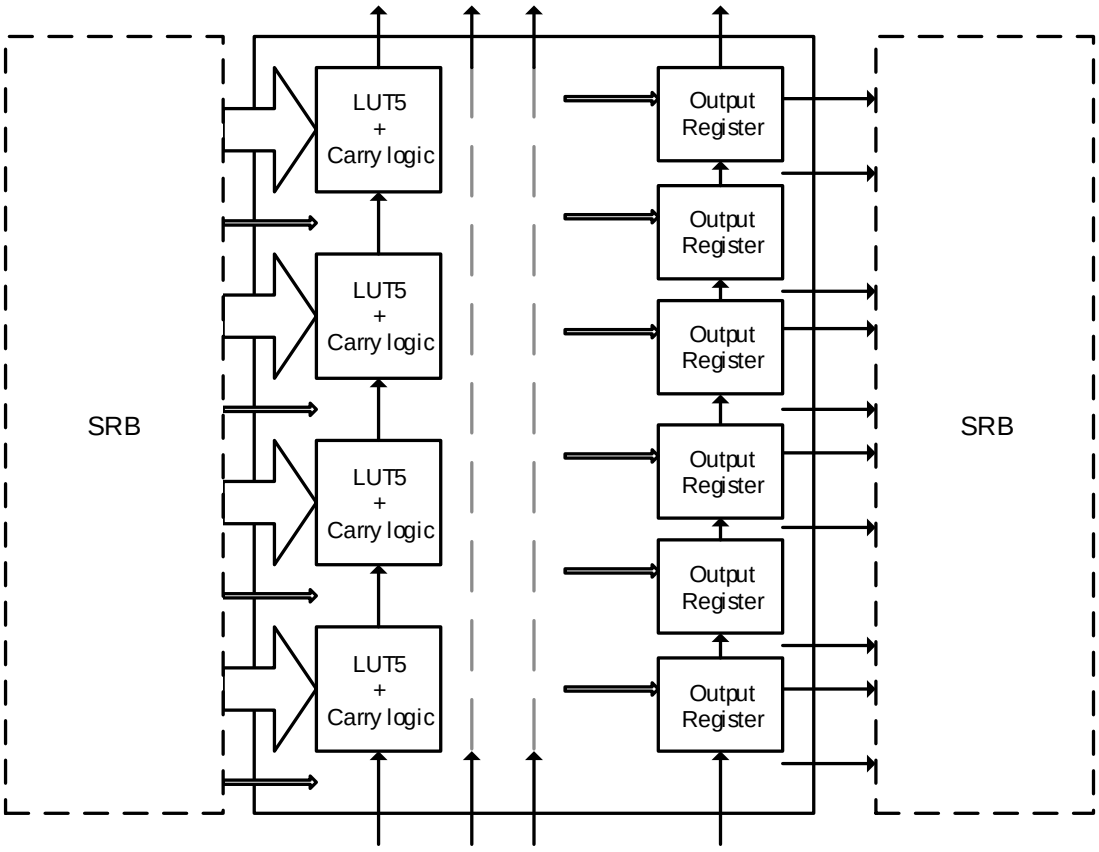

Figure 1-1 Overall Block Diagram of CLM

Key features of CLM include:

- Each CLM contains four multi function LUT5s, six output registers, expasion function selectors, fast carry logic, and four independent cascade chains
- Supports Logic functions
- Supports arithmetic functions and fast arithmetic carry logic
- Shift register functions

- Efficiently implementing multiplexer functions
- Enables Distributed ROM
- CLMS enables Distributed RAM
- Supports cascading of multiple CLMs

The use of CLM can be completed with associated synthesis tools and the Pango Design Suite software by Shenzhen Pango Microsystems Co., Ltd. CLM can also generate Distributed RAM IP using the IP Compiler tool embedded in Pango Design Suite, see the "Distributed RAM IP User Guide" attached in the IP Compiler tool.

Chapter 2 Function Description

2.1 Introduction to CLM Structure

Functionally, the CLM is divided into CLMA and CLMS.

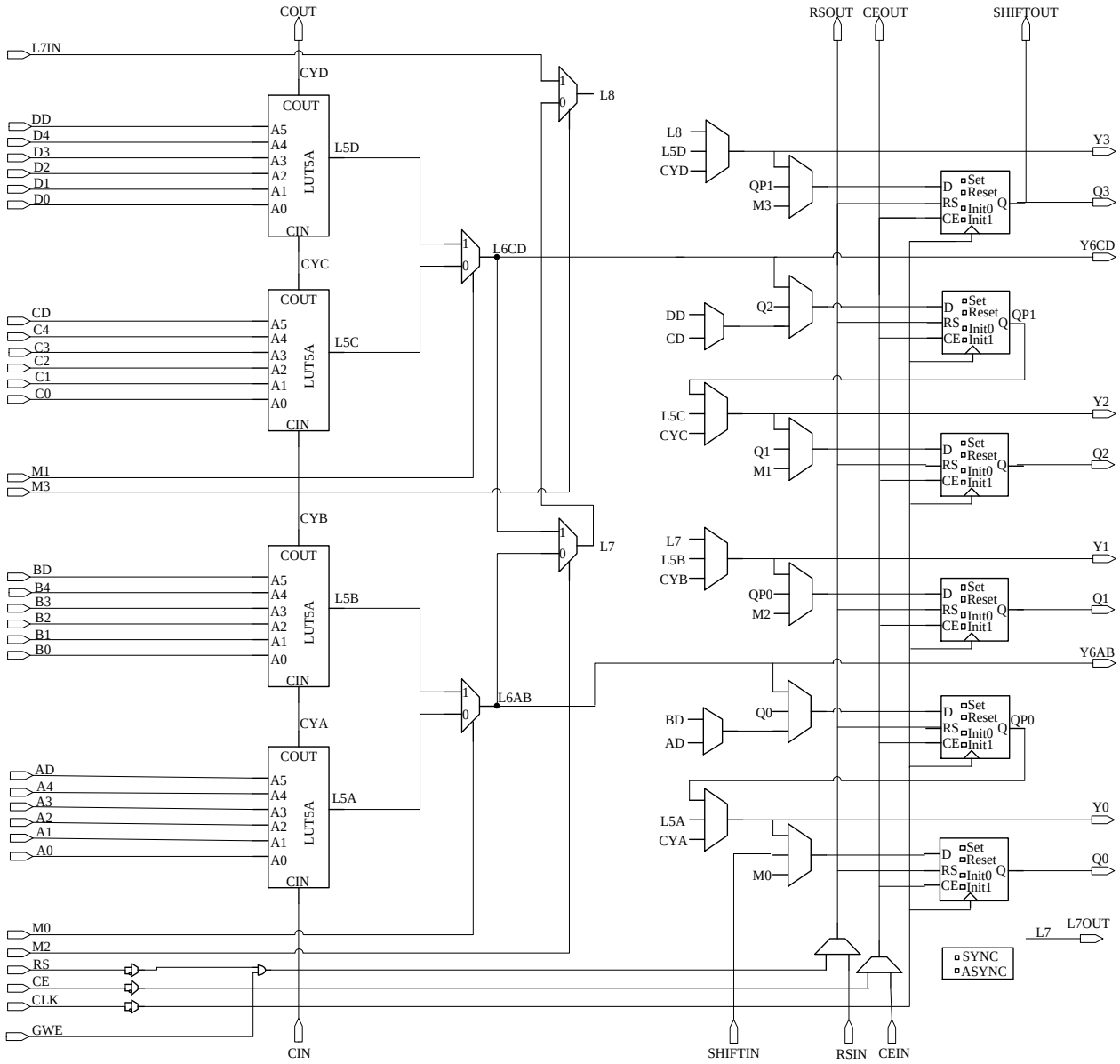

Figure 2-1 Logic Block Diagram of CLMA

The logic block diagram of CLMA is shown in Figure 2-1. Each CLMA contains four LUT5As, six registers, multiple extension function selectors, and four independent cascade chains. LUT5A functions an innovative architectural design that integrates dedicated circuits on top of a 5-input look up table to achieve a 4:1 multiplexer function and fast arithmetic carry logic; the expansion

function selector is primarily used to implement wide-bit look up tables and output selection functions; the cascade chains include arithmetic logic carry chains (from CIN to COUT), dedicated shift register chains (from SHIFTIN to SHIFTOUT), register reset/set cascade chains (from RSIN to RSOUT), and register CE cascade chains (from CEIN to CEOUT). Furthermore, there is a dedicated cascade chain based on LUT7 (from L7IN to L7OUT) between two adjacent CLMs, used to combine and generate LUT8.

CLMS is an extension of CLMA, adding support for Distributed RAM on top of all functions supported by CLMA. CLMS can be configured as either a 16*4 SP (Single Port) RAM or a 16*4 SDP (Simple Dual Port) RAM. The logic block diagram of CLMS is shown in Figure 2-2:

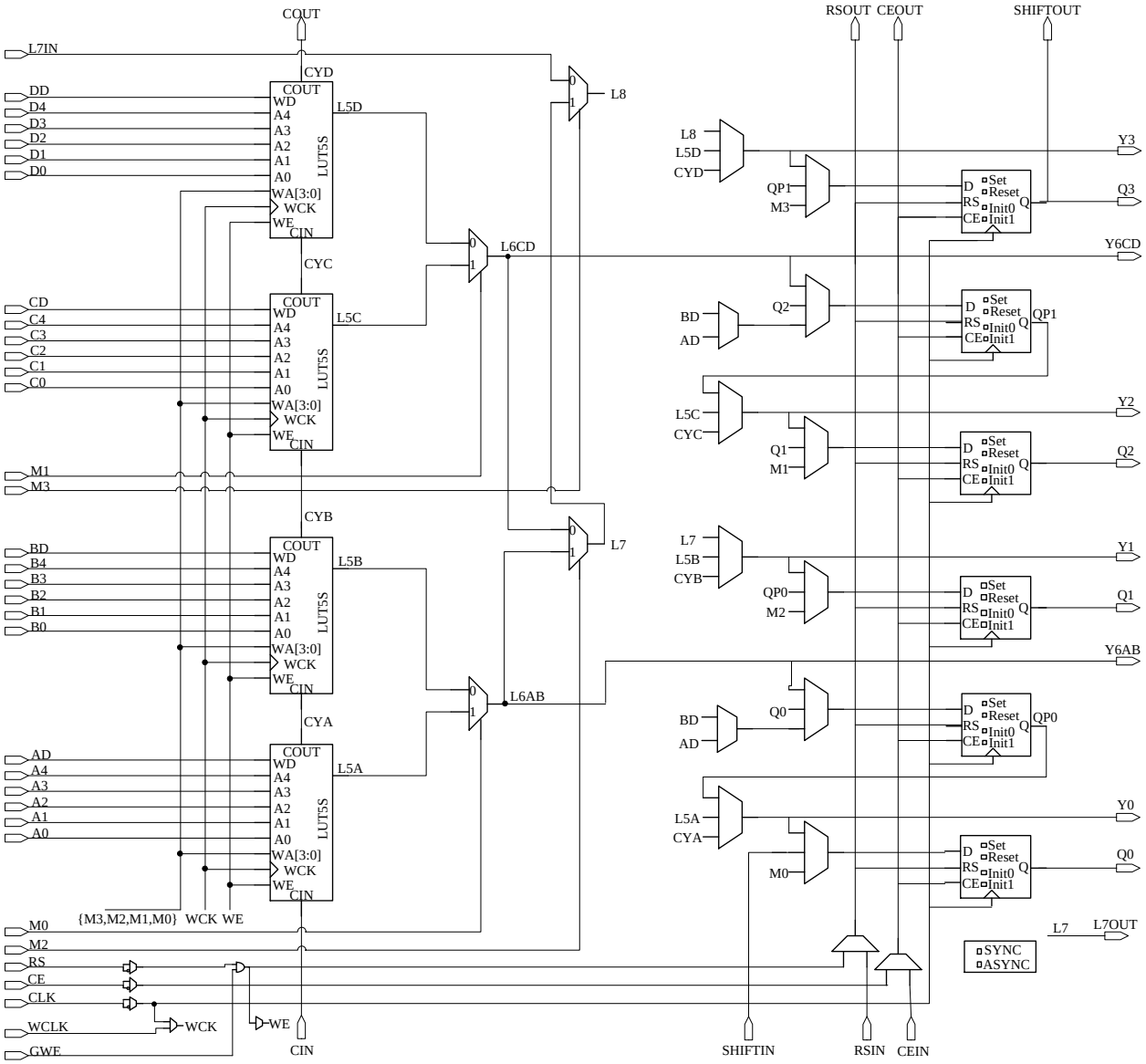

Figure 2-2 Logic Block Diagram of CLMS

Table 2-1 CLM Input Signal List

Input Signal	Signal Source	Description of Signal	Default Conditions
A4,A3,A2,A1,A0	SRB	LUT input	High
B4,B3,B2,B1,B0	SRB	LUT input	High
C4,C3,C2,C1,C0	SRB	LUT input	High
D4,D3,D2,D1,D0	SRB	LUT input	High
AD	SRB	LUT input; Distributed RAM data input (CLMS)	High
BD	SRB	LUT input; Distributed RAM data input (CLMS)	High
CD	SRB	LUT input; Distributed RAM data input (CLMS)	High
DD	SRB	LUT input; Distributed RAM data input (CLMS)	High
M3,M2,M1,M0	SRB	Wide-bit MUX select input; Distributed RAM address input (CLMS)	High
RS	SRB	Local reset/set input	High
CE	SRB	Local clock enable input	High
CLK	SRB	Clock Input	High
WCLK	CLM	Optional Distributed RAM write clock input (only on CLMS)	Low
L7IN	CLM	LUT8 extended MUX input	High
CIN	CLM	Carry chain input; from the CLM downstream.	High
RSCI	CLM	Set/reset control cascade input. from the CLM downstream.	Low
CECI	CLM	Clock enable control cascade input. from the CLM downstream.	High
SRCI	CLM	Shift register chain input. from the CLM downstream.	High
GWE	CCS	Global write enable signal, which, when low, prevents writing to Distributed RAM and registers.	Low

Table 2-2 CLM Output Signal List

Output signal	Destination	Description of Signal	Default Conditions
Y3,Y2,Y1,Y0	SRB	LUT or carry chain output	High
Y6AB	SRB	LUT6 output	High
Y6CD	SRB	LUT6 output	High
Q3,Q2,Q1,Q0	SRB	Register output	High
L7OUT	CLM	LUT7 extended MUX output	High
COUT	CLM	Carry chain output; to the CLM upstream.	High
RSCO	CLM	Set/reset control cascade output. to the CLM upstream.	Low
CECO	CLM	Clock enable control cascade output. to the CLM upstream.	High
SRCO	CLM	Shift register chain output. to the CLM upstream.	High

2.2 Working Modes of LUT5A and LUT5S

LUT5A and LUT5S are the LUT5s in CLMA and CLMS, respectively. LUT5A and LUT5S can be flexibly configured to support various functions such as basic logic, multiplexing, arithmetic logic, Distributed ROM, and Distributed RAM (only on LUT5S).

2.2.1 Logic Functional Mode

In the logic function mode, each LUT5A (or LUT5S) enables one LUT5, and more LUT inputs when combined with expansion function selectors: two LUT6s, one LUT7, or one LUT8 when combined with an adjacent CLM.

One CLM enables:

- Two LUT6s (with M0 and M1 as select inputs); Y6AB/Y6CD work as a combined output.
- One LUT7 (with M2 as the select input); L7 works as the combined output and as the local CLM's L7 hardware cascade chain output.

Two adjacent CLMs enable:

- One LUT8 (with M3 as the select input, connected through the internal L7 cascade chain hardware); L8 works as a combined output.

2.2.2 Multiplexer Mode

In multiplexer mode, each LUT5A (or LUT5S) enables one 4:1 MUX selector, and each CLM supports four 4:1 MUX selectors. Combined with the expansion function selector, every two LUT5As (or LUT5Ss) enable one 8:1 MUX selector, and each CLM supports two 8:1 MUX selectors; each CLM supports one 16:1 MUX selector. Using the dedicated cascade chain of LUT7, every two CLMs can support one 32:1 multiplexer. Wider multiplexed data selection can be generated through the combination of CLMs.

Although the MUX selector features a combinational logic output, its output results can also be output through the registers in the CLM.

2.2.3 Arithmetic Functional Mode

LUT5A (or LUT5S) supports the implementation of efficient arithmetic functions; LUTs, carry logic, and registers (optional) enable the following arithmetic functions:

- Addition
- Subtraction

- Addition/subtraction with dynamic control
- Increment counter
- Decrement counter
- Increment/decrement counters with dynamic control
- Unidirectional counters with preload
- Comparison operation: A= =B, A! =B, A>=B, A<=B
- Fast logic cascade arithmetic
- Wide AND operation

2.2.4 Distributed ROM Mode

In ROM mode, LUT5A (or LUT5S) works a 32*1 ROM, and can also be depth-cascaded through the built-in expansion function selector. The initialization data for the ROM is put into SRAM through the bitstream during the programming configuration process, with the five inputs of LUT5 serving as the read addresses to access the SRAM.

All supported ROM modes are shown in Table 2-3.

Table 2-3 ROM Modes Supported by CLM

ROM Mode	**GTP**	**Number of LUT5s Required**
32×1	GTP_ROM32X1	1
64×1	GTP_ROM64X1	2
128×1	GTP_ROM128X1	4
256×1	GTP_ROM256X1	8 (Two CLMs)

For an introduction to GTP, please refer to "UG030008_Compa Family GTP User Guide."

2.2.5 Distributed RAM Mode

In CLMS, one LUT5S enables one 16x1 simple dual-port RAM, and one CLMS enables one 16x4 single-port RAM. The logic function block diagram is presented in Figure 2-3. AD/BD/CD/DD are used as the data input DI[3:0], M[3:0] as the write address input, A[3:0]/B[3:0]/C[3:0]/D[3:0] as the read address input, sharing ADDR[3:0]; RS port is used as write enable WE; CLK as the clock input WCLK; Y[3:0] or Q[3:0] (register output) as the read data port DO[3:0].

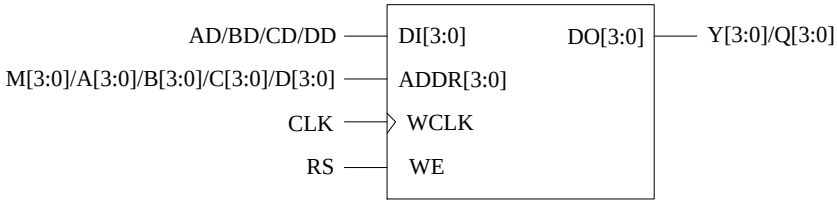

Figure 2-3 Logical Diagram of Four LUT5Ss Configured as a 16x4 Single-Port RAM in CLMS

In CLMS, one LUT5S enables one 16x1 single-port RAM, and one CLMS enables one 16x4 simple dual-port RAM. The logic function block diagram is presented in Figure 2-4. AD/BD/CD/DD are used as the write data input DI[3:0]; M[3:0] as the write address input WADDR[3:0]; A[3:0]/B[3:0]/C[3:0]/D[3:0] as the read address input RADDR[3:0]; RS as write enable WE; WCLK/CLK as the write clock input WCLK; CLK as the read clock input RCLK; DO[3:0] choose Y[3:0] for asynchronous read data output and Q[3:0] for synchronous read data output.

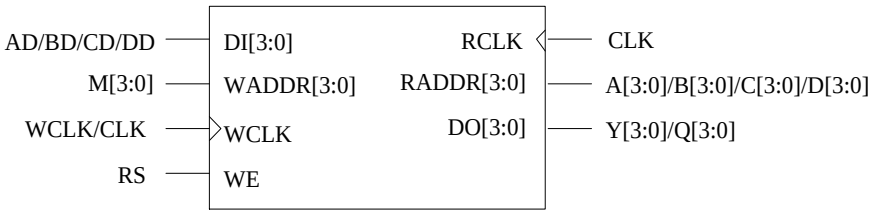

Figure 2-4 Logical Diagram of Four LUT5Ss Configured as a 16x4 Simple Dual-Port RAM in CLMS

The port timing diagram for the distributed single-port RAM supported by the Compa Family CPLDs is shown in Figure 2-5, where MEM is the data stored at the corresponding address. The distributed single-port RAM operates with synchronous write and asynchronous read. Before data is written on the rising edge of the third clock pulse, the address has already changed to ADDR0, and DO outputs the old data MEM(ADDR0) due to the address change; data D0 is written to the RAM's ADDR0 address and, after a certain delay, is output to the DO terminal; when the address changes to ADDR1 and WE is 0, DO outputs MEM(ADDR1).

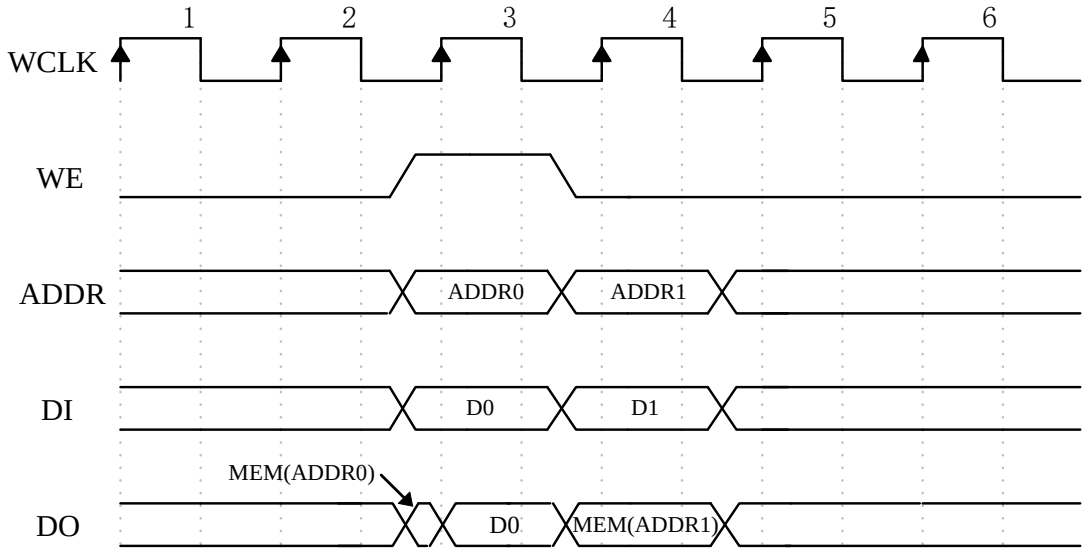

Figure 2-5 Distributed Single-Port RAM Timing Diagram

The port timing diagram for the distributed simple dual-port RAM supported by the Compa Family CPLDs is shown in Figure 2-6, where MEM is the data stored at the corresponding address. The RAM operates with synchronous write and asynchronous read. During the third and fourth clock cycles, with WE high, data D0 and D1 are written to addresses ADDR0 and ADDR1 on the clock's rising edge; when the read address RADDR changes to ADDR0 and ADDR1, data D0 and D1 are respectively read out at DO.

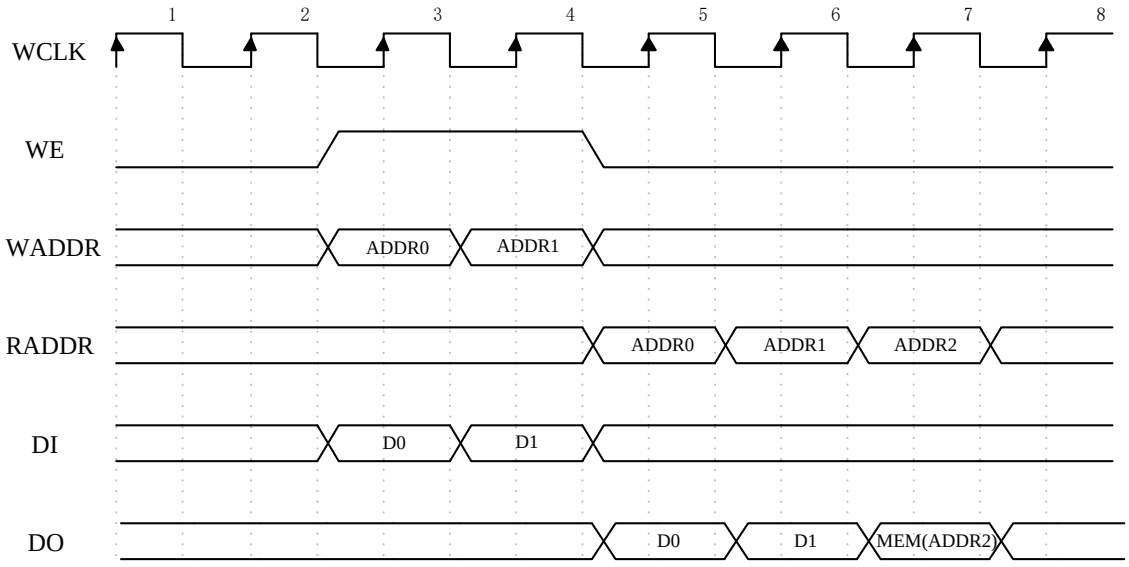

Figure 2-6 Distributed Simple Dual-Port RAM Timing Diagram

All supported RAM modes are shown in Table 2-4.

Table 2-4 RAM Modes Supported by CLMS

RAM Mode	GTP	Number of LUT5s Required
RAM16X1SP	GTP_RAM16X1SP	1
RAM16X4SP	GTP_RAM16X4SP	4
RAM16X1DP	GTP_RAM16X1DP	1
RAM16X4DP	GTP_RAM16X4DP	4

For an introduction to GTP, please refer to "UG030008_Compa Family GTP User Guide."

2.3 Working Modes of the Output Register

There are six output registers in the CLM, including four basic output registers and two additional output registers.

2.3.1 Basic Output Register

The logical diagram of the basic output register is shown in Figure 2-7.

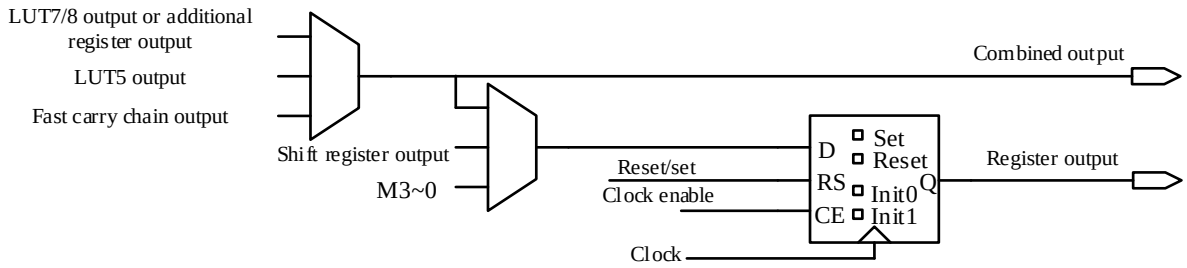

Figure 2-7 Logical Diagram of the Basic Output Register

The input sources for the basic output register include:

- Outputs from other LUT5
- Outputs from LUT7/8 or additional output registers
- Fast carry chain output
- Shift register chain output
- SRB inputs (M3~0)

The basic output registers output to SRB.

Configuration features of the basic output registers:

- Data input comes from LUT, fast carry chain, or SRB
- Programmable reset/set (synchronous/asynchronous, CLM global configuration mode)

- Programmable clock rising/falling edge trigger (CLM global configuration mode)
- Can be used for logic function mode, arithmetic function mode, ROM mode, distributed RAM mode
- Programmable clock/clock enable/local reset and set control signal polarity (CLM global configuration mode)
- Global reset/set (GRS) asynchronously initializes to 0/1
- Configurable as a shift register (comprised of four basic registers and two additional registers)
- Global write enable register (GWE), which allows the register to write when it is high

2.3.2 Additional Output Register

The logical diagram of the additional output register is shown in Figure 2-8.

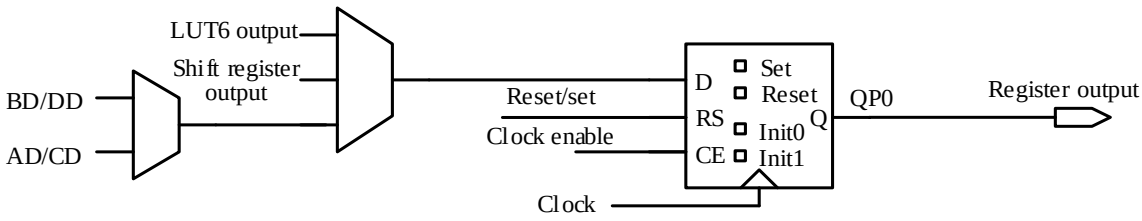

Figure 2-8 Logical Diagram of the Additional Output Register

Through the additional input datapath selector, the input sources for the additional output register include:

- LUT6 output
- SRB inputs
- Shift register chain output

The additional output registers do not output directly to, but rather through the combinational output MUX selector to SRB.

Control configuration features of the additional output register:

- Data input comes from LUT or SRB (AD/BD/CD/DD)
- Programmable reset/set (synchronous/asynchronous, for CLM global configuration mode)
- Programmable clock rising/falling edge trigger (CLM global configuration mode)
- Can be used for logic function mode, arithmetic function mode, ROM mode
- Programmable clock/clock enable/local reset and set control signal polarity (CLM global configuration mode)
- Global reset/set (GRS) asynchronously initializes to 0/1
- Global write enable register (GWE), which allows the register to write when it is high

Chapter 3 Using CLM

The use of CLMs can be facilitated through the Pango Design Suite software by Shenzhen Pango Microsystems Co., Ltd. CLM can be used in the following three ways.

Generate Distributed RAM IP using the IP Compiler tool embedded in Pango Design Suite, see the "Distributed RAM IP User Guide" document attached in the IP Compiler tool;

Use CLM by invoking GTP primitives in the design; an introduction to commonly used CLM primitives can be found in the previous section, and for detailed information, consult the GTP manual. In the "arch\vendor\pango\verilog\simulation" subdirectory of the software installation directory, there are simulation models for various GTPs for reference during design;

Users can add attributes in the design code to constrain the tool to map the corresponding instance to the CLM.

For the design purpose of data storage, distributed memory, block memory, registers, and other resources can be used to achieve user design. Parameters can be added to the code to inform the tool designer which resources are expected to be used to implement the design. For detailed descriptions, see the documentation "ADS_Synthesis_User_Guide" in the software installation directory.

Taking a 16×1 RAM as an example, the scenario of the constraint object as a module can be seen in the following code.

```
module ipm_distributed_spram_v1_2_ram16x1
#(
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
```

Taking a 16×1 RAM as an example, the application scenario of the constraint object as a RAM signal can be seen in the following code.

```
module ipm_distributed_spram_v1_2_ram16x1
#(
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
```

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its series products at any time without prior notice.