Logos Family FPGAs Configurable Logic Module (CLM) User Guide

(UG020001, V1.2.1)

(30.06.2021)

Shenzhen Pango Microsystems Co., Ltd.

Revisions History

Document Revision

Version	Date	Revisions
V1.2.1	30.06.2021	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	**Meaning**
GTP	Generic Technology Primitive
SRB	Signal Relay Block
CLM	Configurable Logic Module
LUT	Look-Up Table

Related Documentation

The following documentation is related to this manual:

1. ***Distributed RAM IP User Guide***

Table of Contents

Tables

Figures

Chapter 1 General Introduction

CLM (Configurable Logic Module) is the basic logic unit of Logos family products, mainly composed of multifunctional LUT5, registers, and expansion function selectors. CLMs are distributed in columns within the Logos family products, supporting two types of structures: CLMA and CLMS, with a distribution ratio of 3:1. Both CLMA and CLMS can implement logic, arithmetic, shift registers, and ROM functions, but only CLMS supports distributed RAM functionality. CLMs are interconnected with each other and with other on-die resources through the Signal Relay Block (SRB).

The use of CLMs can be facilitated through the Pango Design Suite software by Shenzhen Pango Microsystems Co., Ltd. CLMS can also generate distributed RAM IP using the IP Compiler tool embedded in the Pango Design Suite

CLM's key features include:

- Featuring an innovative LUT5 logic architecture
- Each CLM contains 4 multifunctional LUT5s
- Each CLM contains 6 registers
- Arithmetic functional mode supported
- Fast arithmetic carry logic supported
- Efficiently implementing multiplexer
- Capable of implementing ROM functions
- Distributed RAM mode supported
- Cascade chains supported

Device capacity is typically measured by logic units composed of an LUT4 and a flip-flop. As mentioned above, the CLMs in the Logos family FPGAs products use the LUT5 logic architecture, which includes a wealth of flip-flops, latches, carry logic, as well as the ability to create distributed RAM/ROM on-die, thereby increasing their effective capacity. The ratio of the number of logic units to LUT5s is 1.2:1.

Chapter 2 Function Description

2.1 Introduction to CLM Structure

The logic block diagram of CLMA is as follows: each CLMA includes 4 LUT5As, 6 registers, multiple expansion function selectors, and 4 independent cascade chains, etc. LUT5A features 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.

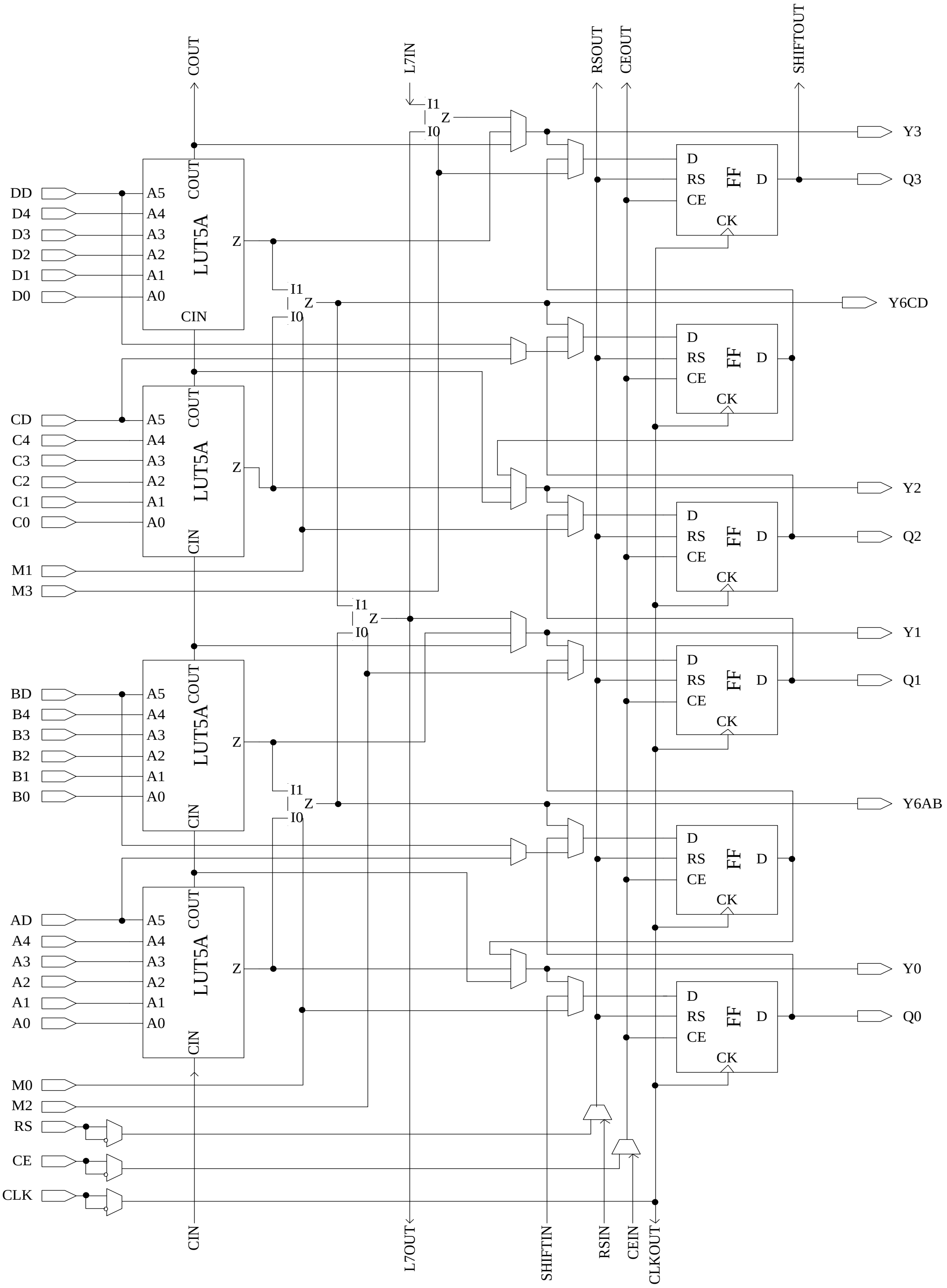

Figure 2-1 CLMA Logic Block Diagram

CLMS is an extension of CLMA, adding support for distributed RAM on the basis of all the functionalities of CLMA, with its multifunctional LUT5 referred to as LUT5S. 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 as follows, where AD–DD input ports are the write data inputs for the distributed RAM mode of 4 LUT6s.

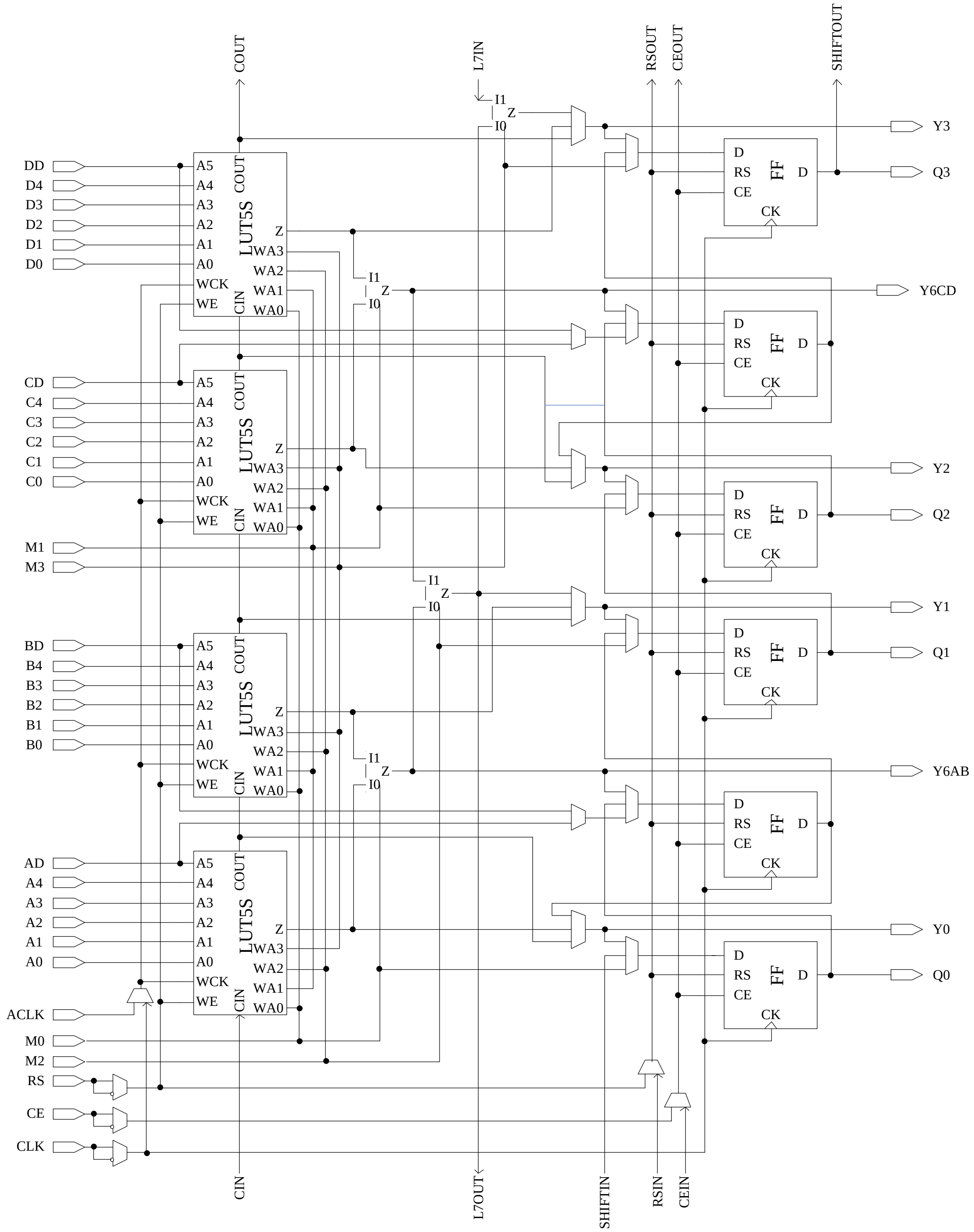

Figure 2-2 CLMS Logic Block Diagram

In the two diagrams above, A0–A5/B0–B5/C0–C5/D0–D5 are the five input ports for the four LUT5s, M0–M2 are the control ports for the multiplexer of extended LUT7/LUT8, RS is the register reset/set signal, CE is the clock enable signal, CLK is the clock input signal, Y0–Y3 are the data outputs ports of the four LUT5s, Q0–Q3 are the output ports of the registers, and Y6AB/Y6CD are the data output ports of the LUT6, which is composed of two LUT5s.

Some internal resources can be shared by CLMA and CLMS. While using shared resources, if some shared resources such as RS, CE, and CLK are occupied, then the users can not use the remaining resources independently. For example, if a register and a CE clock enable signal in the CLMA are used, the remaining registers cannot be controlled with an independent CE clock enablement.

2.2 Operating Modes of LUT5A and LUT5S

LUT5A and LUT5S can be flexibly configured to support various functions such as basic logic, multiplexing, arithmetic logic, ROM functions, and distributed RAM functions (limited to LUT5S).

In ROM mode, LUT5A (or LUT5S) can be used as a 32*1 ROM and can be depth-cascaded using the built-in expansion function selector. The initialization of ROM data is completed during the programming configuration process.

When the four LUT5Ss in CLMS are configured as distributed RAM, their logic function block diagram is as follows: AD/BD/CD/DD are used as data inputs WDI[3:0], ports M0/M1/M2/M3 are used as write address inputs; the RS port is used as write enable; ACLK/CLK are used as clock inputs; A[4:0]/B[4:0]/C[4:0]/D[4:0] are used as read address inputs.

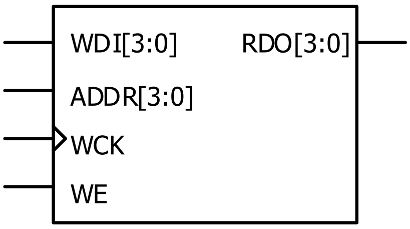

Figure 2-3 Diagram of the Four LUT5Ss in CLMS Configured as 16x4 SP RAM

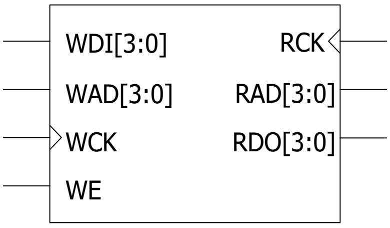

Figure 2-4 Diagram of the Four LUT5Ss in CLMS Configured as 16x4 SDP RAM

2.2.1 Logic Functional Mode

In logic function mode, each LUT5A (or LUT5S) can implement one LUT5, and combined with the expansion function selector, each CLM can support two LUT6s, one LUT7, or combine with an adjacent CLM to implement one LUT8.

2.2.2 Multiplexer Mode

In multiplexer mode, each LUT5A (or LUT5S) can implement one 4:1 multiplexer, and each CLM can support four 4:1 multiplexers. Combined with the expansion function selector, every two LUT5A (or LUT5S) can support one 8:1 multiplexer, and each CLM can support two 8:1 multiplexers; each CLM can support one 16:1 multiplexer. 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.

2.2.3 Arithmetic Functional Mode

In arithmetic function mode, LUT5A (or LUT5S) can implement addition and subtraction operations; counters; comparators; fast XOR logic operations, AND logical operation with large bit width, etc.

Taking addition operation as an example, the implementation method for the basic logic arithmetic unit is:

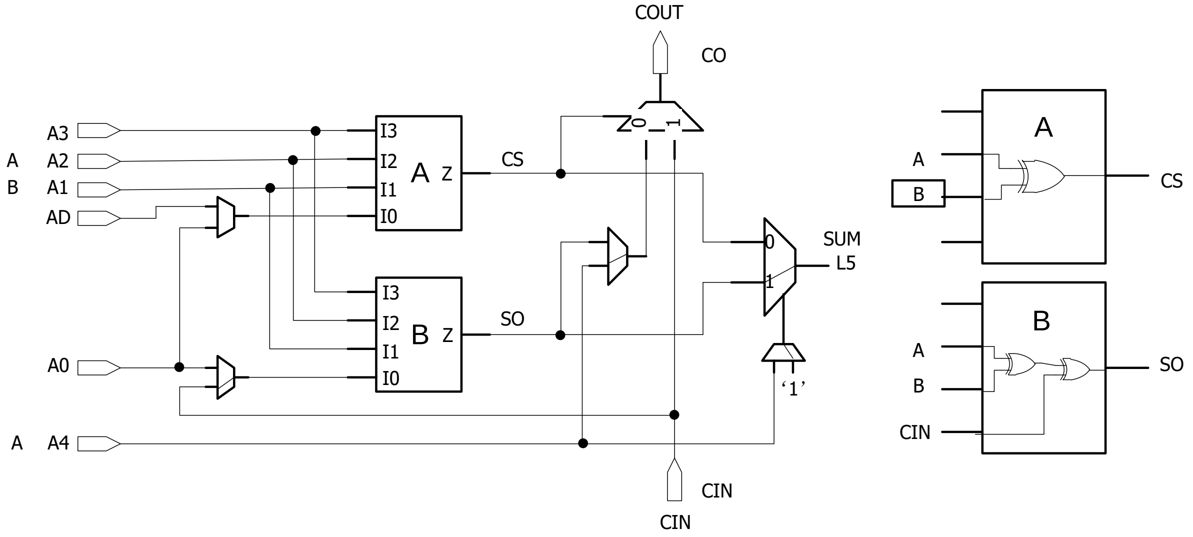

Figure 2-5 Example of Arithmetic Function (Adder)

Addition operation expression: (SUM,CO) = A+B+CIN; the Logic Truth Table (including CS and CO which are the output of the intermediate logic operation) is:

Table 2-1 Logic Truth Table for 1-bit Full Adder

A	B	CIN	SUM(SO)	CO	CS	CO
0	0	0	0	0	0	A
0	0	1	1	0	0	A
0	1	0	1	0	1	CIN

A	B	CIN	SUM(SO)	CO	CS	CO
0	1	1	0	1	1	CIN
1	0	0	1	0	1	CIN
1	0	1	0	1	1	CIN
1	1	0	0	1	0	A
1	1	1	1	1	0	A

- CS = A XOR B; SO = A XOR B XOR CIN
- CO = /CS * A + CS * CIN; SUM = SO

2.2.4 ROM Mode

In ROM mode, LUT5A (or LUT5S) can be used as a 32*1 ROM and can be depth-cascaded using the built-in expansion function selector. The initialization of ROM data is completed during the configuration process.

All supported ROM modes are shown in the table below.

Table 2-2 ROM Modes Supported by CLM

ROM Mode	GTP	Required Number of LUTs
32×1	GTP_ROM32X1	1
64×1	GTP_ROM64X1	2
128×1	GTP_ROM128X1	4
256×1	GTP_ROM256X1	8

2.2.5 Distributed RAM Mode

In general, distributed RAM should be used for all memories that consist of 64 bits or less, unless there is a shortage of CLMS or logic resources for the target device. Distributed RAM is more efficient in terms of resources, performance, and power. For depths greater than 64 bits but less than or equal to 128 bits, the decision on the best resource to use depends on these factors:

Availability, if DRM resources are insufficient, distributed RAM can be used;

The latency requirements. If asynchronous read capability is needed, distributed RAMs must be used.

The data width. Widths greater than 16 bits should use block RAM, if available.

The necessary performance requirements. Registered distributed RAMs generally have shorter clock-to-out timing and fewer placement restrictions than DRMs.

This mode only involves CLMS. A single CLMS can be configured into RAM blocks of various sizes as needed, with the RAM's address and bit width adjusted according to the configuration. The table below lists all the RAM modes supported by the Logos family, among which GTP_RAM16X4SP and GTP_RAM16X4DP are only supported by PGL22G.

Table 2-3 RAM Modes Supported by CLMS

RAM Mode	GTP	Description
RAM16X1SP	GTP_RAM16X1SP	An LUT5S can be configured as a single-port RAM with two 16-bit shared addresses
RAM16X1DP	GTP_RAM16X1DP	An LUT5S can be configured as a simple dual-port RAM with two 16-bit shared addresses
RAM16X4SP	GTP_RAM16X4SP	Four LUT5Ss can be configured as a single-port RAM of 4x16bits
RAM16X4DP	GTP_RAM16X4DP	Four LUT5Ss can be configured as a simple dual-port RAM of 4x16bits
RAM32X1SP	GTP_RAM32X1SP	An LUT5S can be configured as a single-port RAM of 32bits
RAM32X1DP	GTP_RAM32X1DP	An LUT5S can be configured as a simple dual-port RAM of 32bits

Single port: The write and read addresses being shared by one port. Both read and write access RAM through this port, and only one can be accessed at the same time, either read or write. Simple dual-port: It means there are two ports, you can use one for synchronous write as write address, and another one for asynchronous read as read address.

2.2.5.1 RAM16X1SP

Taking LUT5S_A as an example to implement a RAM16X1SP, ports M0/M1/M2/M3 are used for write address input, and the read address is the input A[3:0] of LUT5S_A itself. Since it is a single-port type SRAM, A[3:0] must be connected with M[3:0]; the write data port is the AD input, the read data output port is Y0, and the write enable is RS. One LUT5S can instantiate two RAM16X1SP with shared addresses.

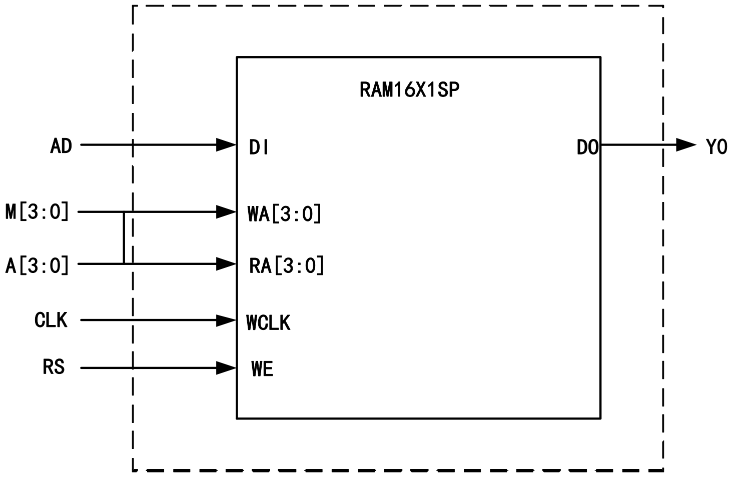

Figure 2-6 Logical Diagram for RAM16X1SP Implementation

2.2.5.2 RAM16X1DP

Taking LUT5S_A as an example to implement a RAM16X1DP, with independent read and write addresses, ports M0/M1/M2/M3 are used for write address input, the read address is the input A[3:0] of LUT5S_A, the write data port is AD input, the read data output port is Y0, and the write enable is RS. One LUT5S can instantiate two RAM16X1DP with shared addresses.

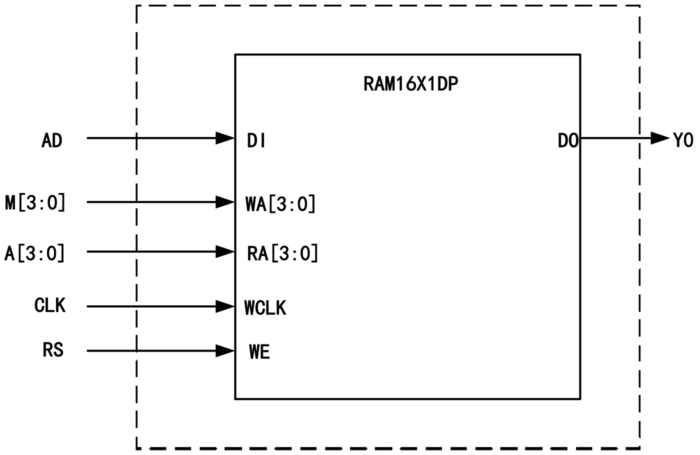

Figure 2-7 Logical Diagram for RAM16X1DP Implementation

2.2.5.3 RAM16X4SP

When the four LUT5S in CLMS are configured as distributed RAM, their logic function block diagram is as follows: AD/BD/CD/DD are used as data inputs AD[3:0]; ports M0/M1/M2/M3 are used for write address input, and A[3:0]/B[3:0]/C[3:0]/D[3:0] are used for read address input. Since it is a single-port type SRAM, the read and write addresses must be connected together; RS port is used for write enable; WCLK is used for clock input.

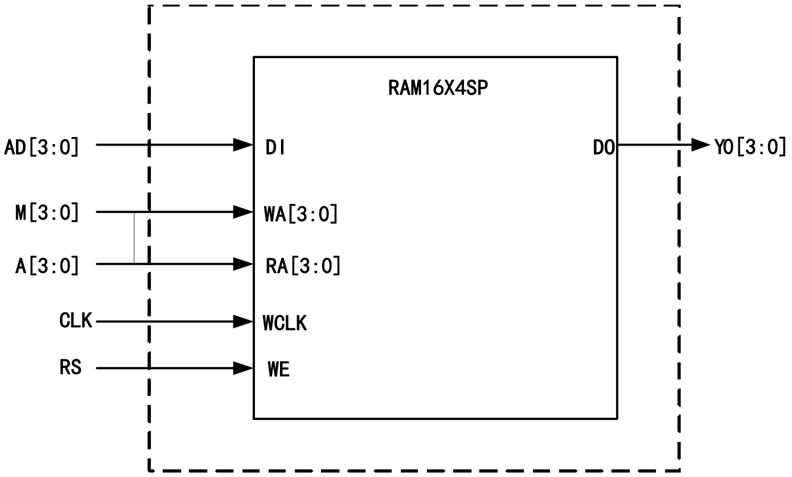

Figure 2-8 Logical Diagram of the RAM16X4DP Implementation

2.2.5.4 RAM16X4DP

When the four LUT5Ss in CLMS are configured as distributed RAM, their logic function block diagram is as follows: AD/BD/CD/DD are used as data inputs AD[3:0], ports M0/M1/M2/M3 are used as write address inputs; the RS port is used as write enable; WCLK are used as clock inputs; A[4:0]/B[4:0]/C[4:0]/D[4:0] are used as read address inputs.

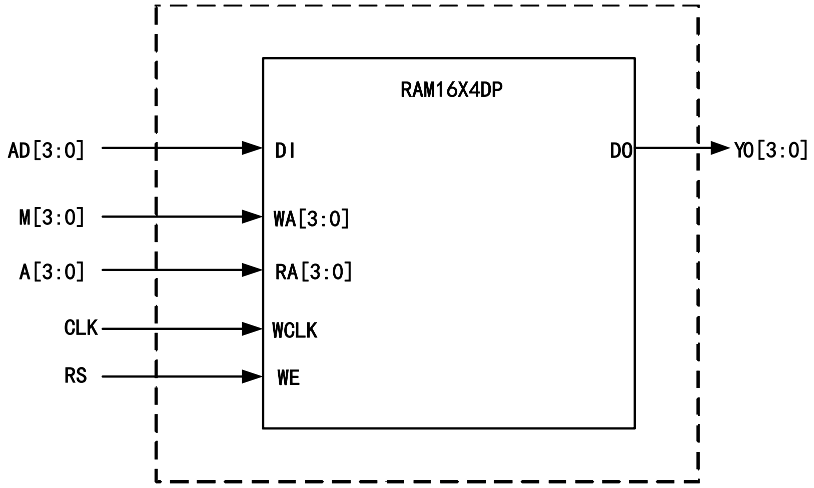

Figure 2-9 Logical Diagram of the RAM16X4DP Implementation

2.2.5.5 RAM32X1SP

Taking LUT5S_A as an example to implement a RAM32X1SP, ports M0/M1/M2/M3 are used for write address input, and the read address is the input A[4:0] of LUT5S_A itself. Since it is a single-port type SRAM, A[4:0] must be connected with{CE, M[3:0]}; the write data port is the AD input, the read data output port is Y0, and the write enable is RS. One LUT5S can instantiate one RAM32X1SP.

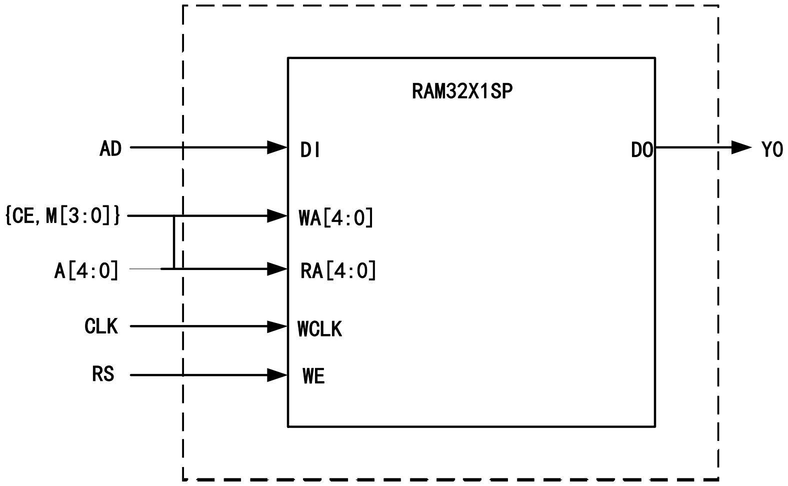

Figure 2-10 Logical Diagram of the RAM32X1SP Implementation

2.2.5.6 RAM32X1DP

Taking LUT5S_A as an example to implement a RAM32X1DP, with independent read address and write address, ports M0/M1/M2/M3/CE are used as write address inputs, the read address is the input A[4:0] of LUT5S_A itself, the write data port is the AD input, the read data output port is Y0, and the write enable is RS. One LUT5S can instantiate one RAM32X1DP.

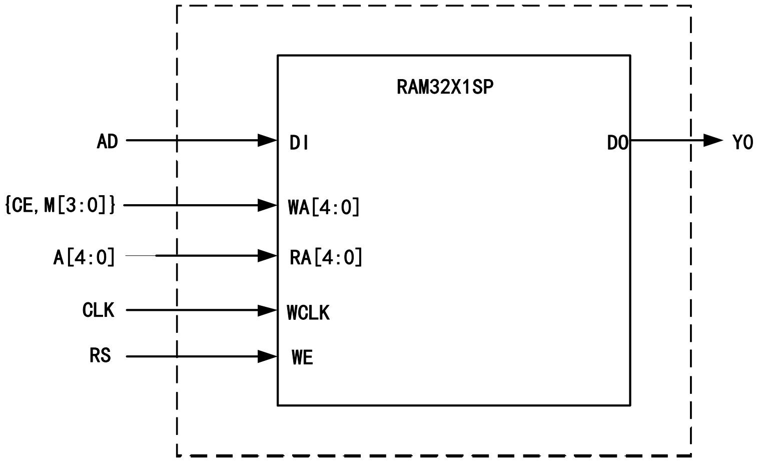

Figure 2-11 Logical Diagram of the RAM32X1DP Implementation

2.3 Operating Modes of the CLM Register

The CLM of Logos family FPGAs has six output registers, whose configurable attributes mainly include:

- Flexible data input selection;
- Support for synchronous reset, synchronous set, asynchronous reset, or asynchronous set modes;
- The register's clock (CLK), clock enable (CE), and local reset/set (RS) signals all support polarity selection;
- Clock enable (CE) and local reset/set (RS) signals both support the fast cascade chain;
- Fast cascade chain for the shift register supported.

Output registers are divided into two categories according to the data input source of the output register: basic output register and additional output register. There are 6 output registers in the CLM of the Logos family FPGAs, including 4 basic output registers and 2 additional output registers, which are now introduced separately.

2.3.1 Basic Output Register

The main register primarily stores the output data of LUT5. The structure is shown in the following figure.

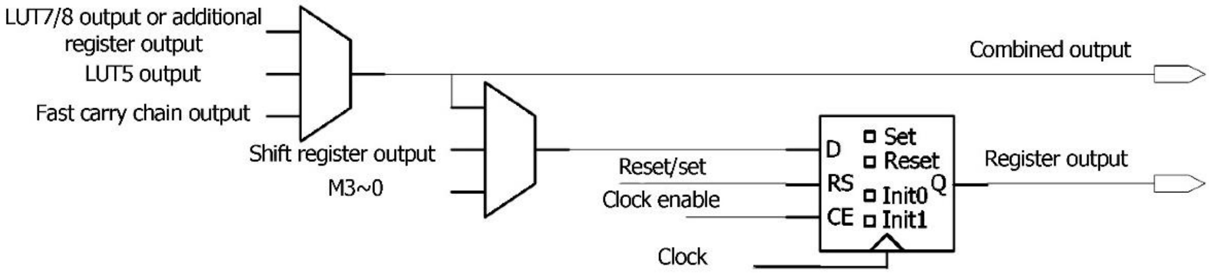

Figure 2-12 Logical Diagram of the Basic Output Register

Through the basic input data path selector, the input sources of the basic output register include LUT5 output, LUT7/8 output or additional output register output, fast carry chain output, shift register chain output, and routing resource input (M3–0). The basic output register outputs to the routing resources.

Control configuration features of the basic output register:

- Data input comes from LUT, fast carry chain, or routing resources
- 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 set control signal polarity (CLM global configuration mode)
- Global reset/set (GRS) asynchronously initializes to 0/1
- Dynamic data path selection of combined output data/routing resource data (reuse local RS signal, for CLM global configuration)
- Configurable as a shift register (comprised of 4 basic registers and 2 additional registers)

2.3.2 Additional Output Register

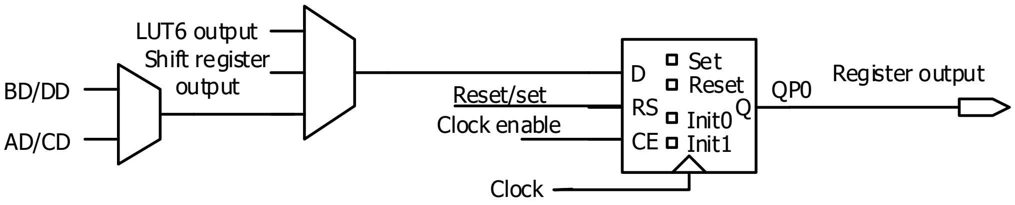

Figure 2-13 Diagram of the Additional Output Register

Through the additional input data path selector, the input sources for the additional output register are: LUT6 output, routing resource input, and shift register chain output. The output of the additional output register does not go directly to the wiring resources but can be directed to the routing resources through an output multiplexer.

Control and configuration features of the additional output register:

- Data input comes from LUT or routing resources (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 set control signal polarity (CLM global configuration mode)
- Global reset/set (GRS) asynchronously initializes to 0/1
- Dynamic data path selection of LUT6 output data/routing resource input data (reuse local RS signal, for CLM global configuration)

2.3.3 Register Control Signal

In each CLM, all registers share the clock (CLK), clock enable (CE), and local reset set (RS) signals, with the register control signals having the following features:

- CLK, CE, and RS all have CLM global polarity control
- CE port input can be disabled, meaning the internal clock is always valid after CE port input is disabled
- RS port input can be disabled, meaning the internal register local reset/set is always invalid
- 6 registers form a 6-bit shift register chain, with the clock coming from the local CLK
- CE and RS have built-in control cascade chains (the CLM in the same column are cascaded in an order from bottom to top)

Chapter 3 Application Examples

3.1 Commonly used GTP

This section mainly introduces several commonly used GTP primitives of CLM, which users can instantiate directly in their designs.

3.1.1 Multiplexer GTP

Table 3-1 Multiplexer GTP

GTP	Input Source	Hardware Resource	Description
GTP_MUX2LUT6	Output of LUT5	L7ABMU or L7CDMUX	Used for constructing LUT6, 4:1 multiplexer, and other logic
GTP_MUX2LUT7	Output of LUT6	L7ABMU or L7CDMUX	Used for constructing LUT7, 8:1 multiplexer, and other logic
GTP_MUX2LUT8	Output of L7ABMUX and L7CDMUX	L8MUX	Used for constructing LUT8, 16:1 multiplexer, and other logic

GTP_MUX2LUT6, GTP_MUX2LUT7, and GTP_MUX2LUT8 have the same ports, as shown in the figure below.

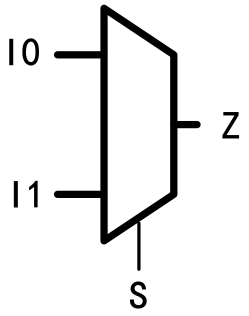

Figure 3-1 Diagram of Multiplexer GTP Port

The port description of the multiplexer is shown in the table below.

Table 3-2 Port Description of Multiplexer

Port	Direction	Function Description
I0	Input	Input Signal
I1	Input	Input Signal
S	Input	Selection Signal
Z	Output	Output Signal

The truth value table for the multiplexer is shown in the table below-. When the select signal S is 1'b0, the output is I0; when S is 1'b1, the output is I1.

Table 3-3 Truth Value Table of the Multiplexer

Input			Output
I1	I0	S	Z
X	0	0	0
X	1	0	1
0	X	1	0
1	X	1	1

Note: X represents any value.

3.1.2 Carry chain GTP

The carry chain primitive GTP_LUT5CARRY represents the fast carry logic in the CLM, often used in constructing arithmetic logic such as adders and multipliers.

GTP_LUT5CARRY is a fast carry logic, where different INIT parameter values correspond to different functions, and the corresponding INIT parameter must be specified when used.

Table 3-4 Parameter Description of Carry Chain GTP

Parameter Name	Parameter Type	Valid Values	Function Description
INIT	<binary>	0 – 2^32-1	Output Configuration Parameter
ID_TO_LUT	<string>	"TRUE","FALSE"	Selection parameters of I0 and ID
CIN_TO_LUT	<string>	"TRUE","FALSE"	Selection parameters of CIN and I0
I4_TO_CARRY	<string>	"TRUE","FALSE"	Selection parameters of I4 and LUT4 output
I4_TO_LUT	<string>	"TRUE","FALSE"	Selection parameters of I4 and 1’b1

The port diagram of the GTP_LUT5CARRY primitive is shown in the figure below.

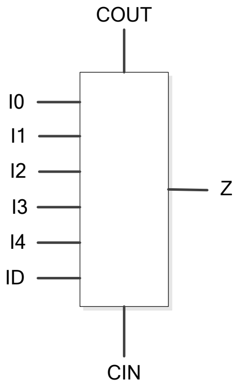

Figure 3-2 Port Diagram of the Carry Chain GTP

For the port description of the GTP_LUT5CARRY primitive, see the table below.

Table 3-5 Port Diagram of the Carry Chain GTP

Port	Direction	Function Description
I0	Input	Input Signal
ID	Input	Input signal, valid when ID_TO_LUT = "TRUE"
I1	Input	Input Signal
I2	Input	Input Signal
I3	Input	Input Signal
I4	Input	Input signal, valid when I4_TO_CARRY = "TRUE"
CIN	Input	Cascade input, valid when CIN_TO_LUT = "TRUE", must be connected to COUT
COUT	Output	Cascade output
Z	Output	Output Signal

3.1.3 Distributed RAM GTP

To use the distributed RAM resources in CLMS, one can instantiate distributed RAM primitives, the Logos family FPGAs supports distributed RAM GTP as shown in the table below, where GTP_RAM16X4SP and GTP_RAM16X4DP only support PGL22G.

Table 3-6 RAM GTP

GTP	Density	Type	Address Inputs
GTP_RAM16X1SP	16-bit	Single port	A0–3/B0–3/C0–3/D0–3 (write/read)
GTP_RAM16X1DP	16-bit	Simple dual port	D0–3 (write) A0–3/B0–3/C0–3 (read)
GTP_RAM16X4SP	4x16-bit	Single port	A0–3/B0–3/C0–3/D0–3 (write/read)
GTP_RAM16X4DP	4x16-bit	Simple dual port	M0–3 (write) A0–3/B0–3/C0–3/D0–3 (read)
GTP_RAM32X1SP	32-bit	Single port	A0–4/B0–4/C0–4/D0–4 (write/read)
GTP_RAM32X1DP	32-bit	Simple dual port	CE, M0–3 (write) A0–4/B0–4/C0–4/D0–4 (read)

Distributed RAM can have initial value parameters set, taking 32-bit RAM as an example, see the table below.

Table 3-7 Parameter Description of Distributed RAM GTP

Parameter Name	Parameter Type	Valid Values	Function Description
INIT	<binary>	32’h0–32’hffff_ffff	Memory initialization configuration parameters

The port diagram of the Logos family FPGAs-supported distributed single-port and simple dual-port RAM is shown in the figure below.

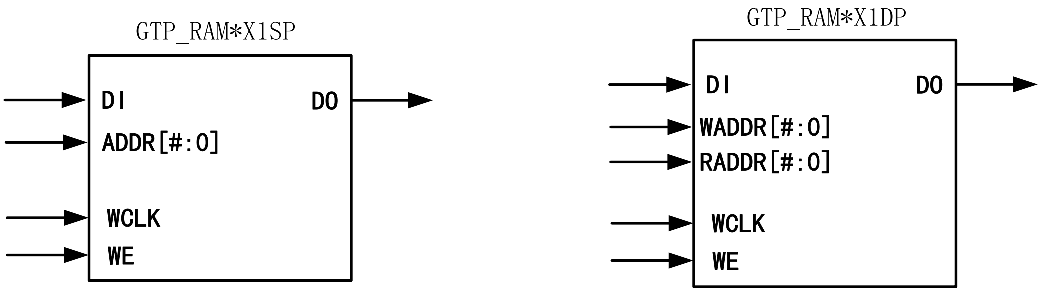

Figure 3-3 Port Diagram of RAM GTP

The port description of the distributed single-port and simple dual-port RAM supported by the Logos family FPGAs is shown in the table below.

Table 3-8 Port Description of Distributed RAM GTP

Port Name	Direction	Description
DI	Input	Write data signal
DO	Output	Read data signal
[R/W]ADDR	Input	Single port: ADDR: Address signal, shared for read and write; Simple dual port: WADDR: write address RADDR: read address
WCLK	Input	Clock signal
WE	Input	Write enable signal, active high

The port timing diagram of the distributed single-port RAM supported by the Logos family FPGAs is shown in the figure below.

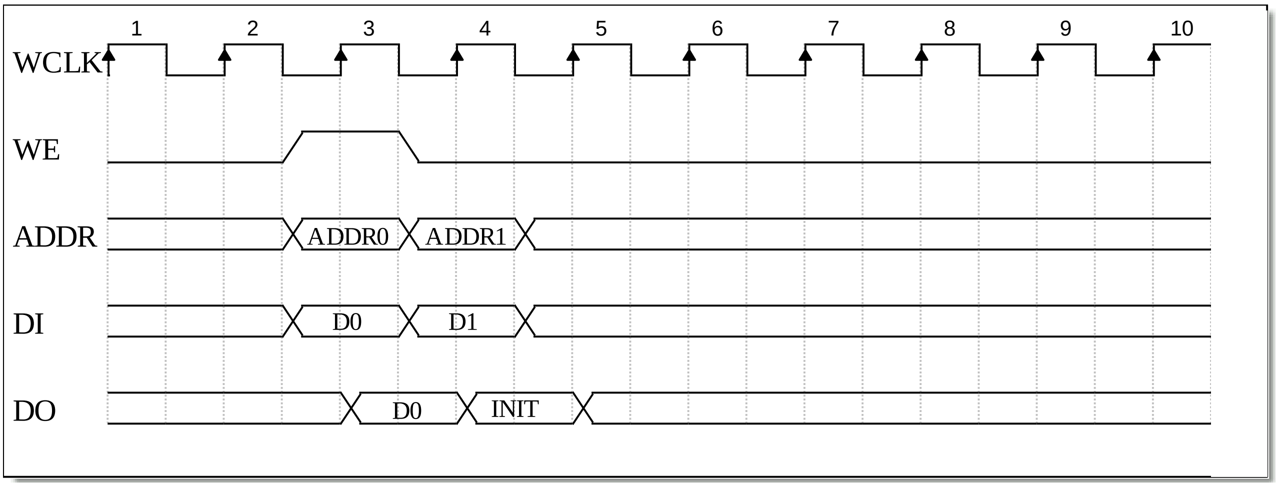

Figure 3-4 Timing Diagram of Distributed Single-Port RAM

The port timing diagram of the distributed simple dual-port RAM supported by the Logos family FPGAs is shown in the figure below.

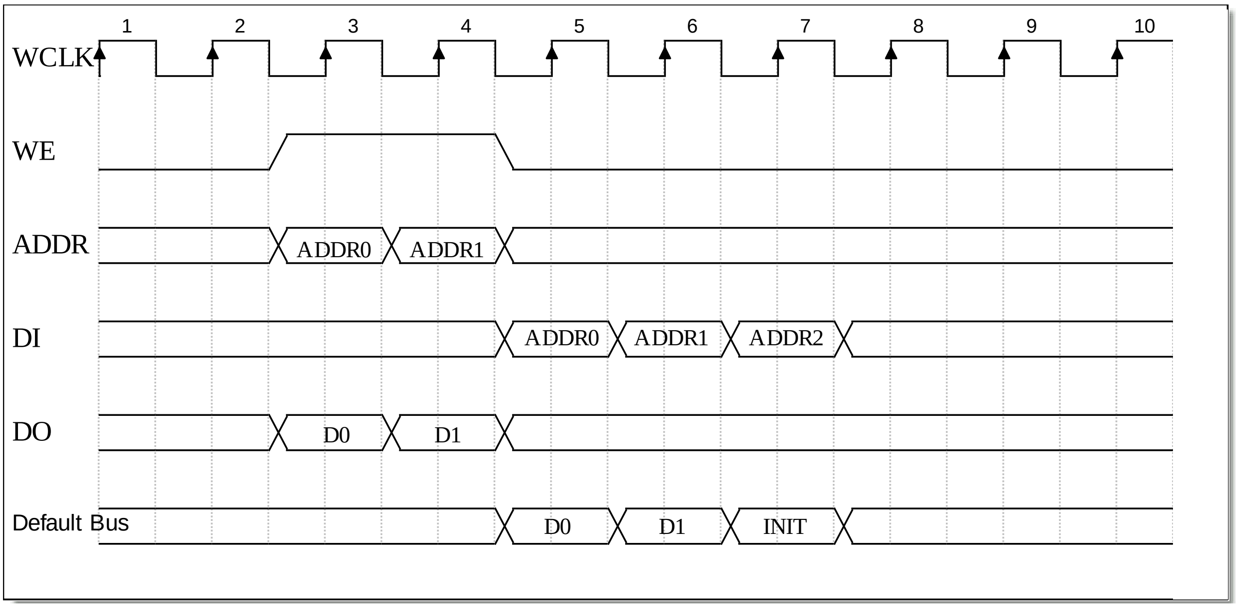

Figure 3-5 Timing Diagram of Distributed Simple Dual-Port RAM

3.1.4 ROM GTP

To use CLM as a read-only memory, one can instantiate ROM primitives, as the Logos family FPGAs supports the ROM GTP shown in the table below.

Table 3-9 ROM GTP

GTP	**Density**	**Address Inputs**
GTP_ROM32X1	32-bit	A0–4/B0–4/C0–4/D0–4
GTP_ROM64X1	64-bit	A0–4/B0–4/C0–4/D0–4 as the lower 5-bit address, M0/M1 as the highest bit address
GTP_ROM128X1	128-bit	A0–4/B0–4/C0–4/D0–4 as the lower 5-bit address, M0/M1 as the 7th bit address, M2 as the highest bit address
GTP_ROM256X1	256-bit	Composed of two cascaded CLMs

Distributed ROM can have initial value parameters set, taking 256bit ROM as an example, see the table below.

Table 3-10 Parameter Description of Distributed ROM GTP

Parameter Name	**Parameter Type**	**Valid Values**	**Function Description**
INIT	<binary>	256’h0–256’hffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff_ffff	ROM Initialization Configuration Parameters

Taking GTP_ROM256X1 as an example, the port diagram ROM GTP is shown in the figure below.

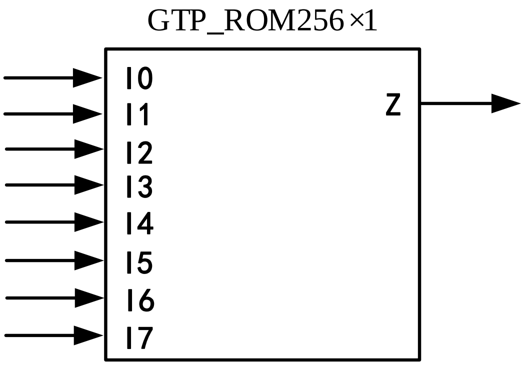

Figure 3-6 Port Diagram of ROM GTP

For the port description of GTP_ROM256X1, see the table below.

Table 3-11 Port Description of Distributed ROM GTP

Port	**Direction**	**Function Description**
I0	Input	ROM read address addr[0]
I1	Input	ROM read address addr[1]
I2	Input	ROM read address addr[2]
I3	Input	ROM read address addr[3]
I4	Input	ROM read address addr[4]
I5	Input	ROM read address addr[5]
I6	Input	ROM read address addr[6]
I7	Input	ROM read address addr[7]
Z	Output	Read data

This GTP implements the ROM storage function. Inputs I7–I0 constitute the read data address, reading the value of the specified bit of the ROM's initial configuration parameters.

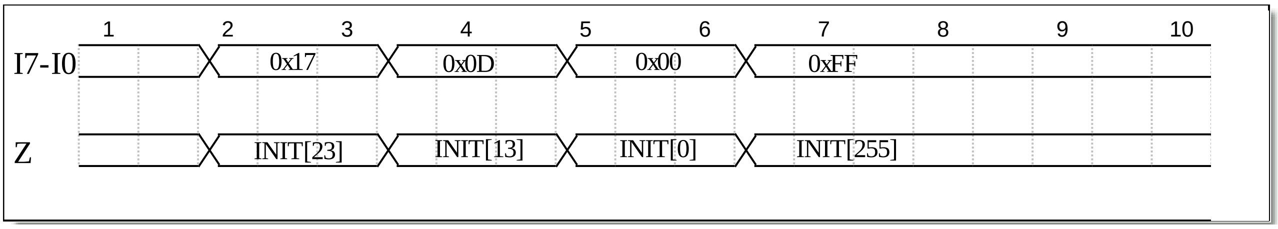

Figure 3-7 GTP_ROM256X1 Waveform Diagram

3.1.5 Flip-flop GTP

The Logos family FPGAs supports flip-flop GTP as shown in the table below.

Table 3-12 Latch GTP

GTP	Description
GTP_DFF	Flip-flop
GTP_DFF_C	Asynchronous clear flip-flop
GTP_DFF_E	Flip-flop with enable
GTP_DFF_CE	Asynchronous clear flip-flop with enable
GTP_DFF_P	Asynchronous set flip-flop
GTP_DFF_PE	Asynchronous set flip-flop with enable
GTP_DFF_R	Synchronous clear flip-flop
GTP_DFF_RE	Synchronous clear flip-flop with enable
GTP_DFF_S	Synchronous set flip-flop
GTP_DFF_SE	Synchronous set flip-flop with enable

The port diagram of the flip-flop GTP supported by the Logos family FPGAs is shown in the figure below.

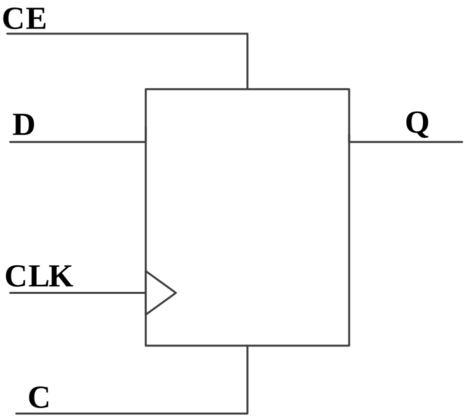

Figure 3-8 Port Diagram of GTP_DFF_CE

The port description of the flip-flop GTP supported by the Logos family FPGA is shown in the table below.

Table 3-13 Port Description of Flip-flop GTP

Port Name	Direction	Description
CLK	Input	Input Clock
D	Input	Input signal of trigger
Q	Output	Output signal of trigger
C	Input	Asynchronous clear signal
CE	Input	Enable signal, active high

Port Name	Direction	Description
P	Input	Asynchronous set signal
R	Input	Synchronization clear signal
S	Input	Synchronous set signal

The parameter description for the flip-flop GTP supported by the Logos family FPGAs is shown in the table below.

Table 3-14 Parameter Description of Flip-flop GTP

Parameter Name	Parameter Type	Setting Value	Function Description
GRS_EN	<string>	"TRUE","FALSE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	Initial value, asynchronously triggered by the global reset signal (this parameter is not supported in simulation)

3.1.6 Latch GTP

The Logos family FPGAs supports latch GTP as shown in the table below.

Table 3-15 Latch GTP

GTP	Description
GTP_DLATCH	Latch
GTP_DLATCH_C	Asynchronous clear latch
GTP_DLATCH_E	Latch with enable
GTP_DLATCH_CE	Asynchronous clear latch with enable
GTP_DLATCH_P	Asynchronous set latch
GTP_DLATCH_PE	Asynchronous set latch with enable

The port diagram of the latch GTP supported by the Logos family FPGAs is shown in the figure below.

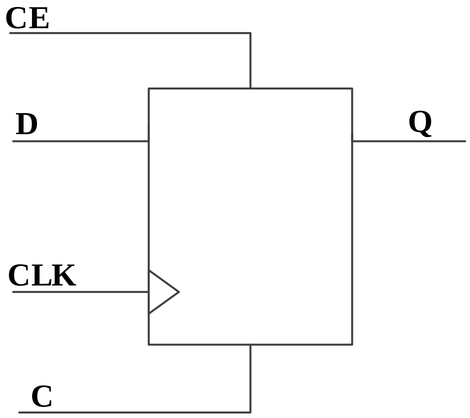

Figure 3-9 Port Diagram of GTP_DFF_CE

The port description of the latch GTP supported by the Logos family FPGAs is shown in the table below.

Table 3-16 Port Description of Latch GTP

Port Name	Direction	Description
G	Input	Input level
D	Input	Latch input signal
Q	Output	Latch output signal
P	Input	Asynchronous set signal
GE	Input	Enable signal, active high
C	Input	Asynchronous clear signal

The parameter description of the latch GTP supported by the Logos family FPGAs is shown in the table below.

Table 3-17 Parameter Description of Latch GTP

Parameter Name	Parameter Type	Setting Value	Function Description
GRS_EN	<string>	"TRUE","FALSE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	Initial value, asynchronously triggered by the global reset signal (this parameter is not supported in simulation)

3.2 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.

1. To generate distributed RAM IP using the IP Compiler tool embedded in Pango Design Suite, see the IP Compiler's documentation 'Distributed RAM IP User Guide' for details;
2. Use CLM by invoking GTP primitives in the design, see the previous section for an introduction to commonly used CLM primitives, or consult the GTP manual for detailed information. Verilog source code for each GTP is available in the "arch\vendor\pango\verilog\simulation" subdirectory of the software installation directory for design reference;
3. Users can add attributes to the design code to constrain the tool to map the corresponding instance to 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. See the table below for detailed descriptions.

Table 3-18 Storage Resource Constraint Methods

Resource Type	Constraint Command	Constraint Object	Property	Resource Type	Usage
RAM	syn_ramstyle	Module or RAM signal	block_ram	DRM	object /* synthesis syn_ramstyle="block_ram" */
			select_ram	CLM	
			no_rw_check	Register	
ROM	syn_romstyle	Module or ROM signal	block_rom	DRM	object /* synthesis syn_romstyle="block_rom" */
			select_rom	CLM	

Taking a 128×1 RAM as an example, see the following code for a scenario where the constraint object is a module.

```
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
```

Taking a 128×1 RAM as an example, see the following code for an application scenario where the constraint object is a RAM signal.

```
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" */;
……
Endmod
```

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 family products at any time without prior notice.