BSTEPC16 User Manual

1 Ver.1.0

I. Product Description:

Version Number	date	Version Notes	Written by
V1.00	2013-4-12	Initial release	Que Xiaoqian

Table of contents

- Detailed Functional Description
- FPGAConfiguration
- Fast Parallel Slave Mode Configuration (FPP）
- Serial Slave Mode Configuration (PS）
- Simultaneous configuration
- Dynamic configuration (page mode)
- Real-time decompression
- Programmable configuration clock
- FLASHOnline Programming
- JTAGprogramming
- External FLASHinterface
- Package Specifications
- Ordering Information
- Product Overview

BSTEPC16 is a military-grade enhanced configuration device launched by Chengdu Sino-Microelectronics Technology Co., Ltd. This device provides a single-chip, high-speed, advanced configuration solution for high-density FPGAs. The device core is mainly divided into two parts: a configuration controller and a 16MBits FLASH memory. The FLASH memory is used to store the configuration information of a single or multiple FPGAs in the configuration system. After the FPGA configuration is completed, the FLASH can be accessed through the external interface, so that the unused part of the FLASH can store user data.

II. Features

The following Altera devices can be configured on a single chip: ACEX IK, APEX 20K (including APEX 20K, APEX 20KC, and APEX 20KE), APEX II, Arria GX, Cyclone, Cyclone II, FLEX 10K (including FLEX 10KE and FLEX 10KA), Mercury, Stratix II, and Stratix II GX devices.

The BSTEPC16 uses 16-Mbit FLASH to store configuration information. The chip's on-chip

decompression function doubles the chip's effective configuration information storage density.

ExternalThe FLASH interface supports parallel programming of FLASH and allows external processors to access unused storage. The external FLASH interface has block/sector protection capabilities.

Page mode supports up to 8 configuration information for remote and local reconfiguration of the entire system.

Support8-bit parallel configuration (FPP), output 8 bits of configuration data in each DCLK cycle.

Supports n-bit simultaneous configuration of FPGA (n = 1, 2, 4, 8).

A is selected by pin2ms or 100ms power-on reset time.

A configuration clock supports programmable selection of clock source and frequency synthesis.

- Supports multiple configurable clock sources (internal crystal oscillator and external clock input)
- The external clock frequency can reach up to 100M.

The internal configuration clock defaults to 10MHz, and higher frequency internal clocks 33, 50 and 66MHz can be selected.

The clock can be divided into integers of 1 to 16 times, and fractional divisions of 1.5 and 2.5 times through a programmable frequency division circuit.

- 88 PINCeramic FBGA package.

A Supply voltage3.3V (core and I/O).

- Compatible with IEEE Std.1532 In-System Programming (ISP) specification.

A through The JTAG interface supports ISP.

A is controlled by private instructions. The nINIT_CONF pin implements FPGA configuration, and the internal pull-up resistor on this pin is always active.

There are programmable internal weak pull-up resistors on the OE and nCS pins.

A External The address and control signals of the FLASH interface have weak pull-up resistors, and the data lines have holding circuits.

III. Application

BSTEPC16 is used to configure Altera's FPGA. The FPGAs that support configuration are shown in Table 3.1:

Table 3.1. FPGAs that support BSTEPC16 configuration

FPGA Series	FPGA Devices
Aria GX	EP1AGX20C
	EP1AGX35C EP1AGX25D
	EP1AGX50C EP1AGX50D
	EP1AGX60C EP1AGX60D EP1AGX60E
	EP1AGX90E
Stratix	EP1S10
	EP1S20
	EP1S25 EP1S30 EP1S40 EP1S60 EP1S80
Stratix GX	EP1SGX10 EP1SGX25 EP1SGX40
Stratix II	EP2S15 EP2S30 EP2S60 EP2S90 EP2S130

	EP2S180
Stratix II GX	EP2SGX30C EP2SGX30D EP2SGX60C EP2SGX60D EP2SGX60E EP2SGX90E EP2SGX90F EP2SGX130G
Cyclone	EP1C3 EP1C4 EP1C6 EP1C12

	EP1C20
Cyclone II	EP2C5 EP2C8 EP2C20 EP2C35 EP2C50 EP2C70
ACEX 1K	EP1K10 EP1K30 EP1K50 EP1K100
APEX 20K	EP20K100 EP20K200 EP20K400
APEX 20KE	EP20K30E EP20K60E EP20K100E EP20K160E EP20K200E EP20K300E EP20K400E EP20K600E EP20K1000E

	EP20K1500E
APEX II asd	EP2A15 EP2A25 EP2A40 EP2A70
FLEX 10K	EPF10K10 EPF10K20 EPF10K30 EPF10K40 EPF10K50 EPF10K70 EPF10K100
FLEX 10KA	EPF10K10A EPF10K30A EPF10K50V EPF10K100A EPF10K130V EPF10K250A
FLEX 10KE	EPF10K30E EPF10K50E EPF10K50S EPF10K100B EPF10K100E EPF10K130E EPF10K200E EPF10K200S
Mercury	EP1M120 EP1M350

IV. Module block diagram and description

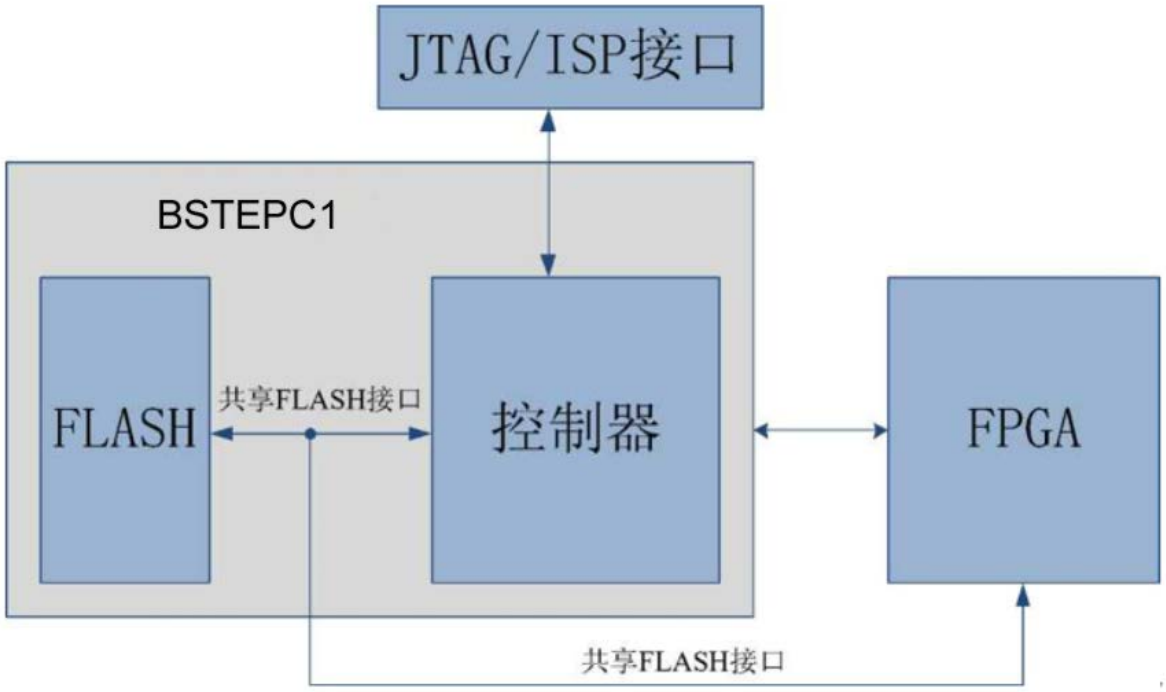


Figure 4.1. Functional block diagram

The circuit structure of the BSTEPC16 device consists of the following two parts: controller and FLASH.

4.1. Controller

The controller configures the FPGA based on the configuration points selected in the Quartus II software.

4.2. FLASH

FLASH is used as a storage for configuration data. After configuration is completed, FLASH can be accessed through an external FLASH interface.

4.2.1. FLASH Overview

The FLASH in BSTEPC16 is a memory with a storage capacity of 16-Mbit (16777216-bit) and a data bus read and write width of 16 bits.

The parameter module of FLASH can be locked through the pin WP#, so that the parameter module cannot be erased or programmed.

The internal operation of FLASH is performed by sending instructions to the command

user interface (CUI). The write state machine (WSM) inside FLASH will automatically execute the corresponding algorithm and timing to complete the erase and program operations. The status register inside FLASH will display the status of WSM, thereby determining the status of the erase and program operations.

Pin RP# provides additional protection to avoid some erroneous operations.

4.2.2. FLASH Features

A module division

- 8 8KB parameter modules
- 31 64KB main modules

Module lock

- Pin WP# can lock the parameter module

Automatic programming and module erasure determination

- Status register

4.2.3. FLASH bus operation

The online reading, programming and erasing operations of FLASH are performed by an external CPU or other processor. All bus cycles to or from FLASH conform to the standard processor cycle. Four control signals determine the data in and out of FLASH: CE#, OE#, WE# and RP#. The bus operation truth table is shown in Table 4.1 below:

Table 4.1. Bus operation truth table

Model	Note	RP#	CE#	OE#	WE#	DQ[7:0]	DQ[15:8]
Read(array, status, ID)	-	VIH	VIL	VIL	VIH	DOUT	DOUT
Output Shutdown	-	VIH	VIL	VIH	VIH	High Z	High Z
Standby	(1)	VIL	VIH	X	X	High Z	High Z
Reset	(1)	VIL	X	X	X	High Z	High Z
Write	-	VIH	VIL	VIH	VIL	DIN	DIN

Note:

(1) For control and address pins, X must be high or low.

4.2.3.1. Read Operation

There are three read modes for FLASH: read array, read ID, and read status register. These modes are independent of VCCW voltage. FLASH can only enter the corresponding mode when the corresponding read mode command enters CUI. When it is just powered on or after exiting the reset state, FLASH will automatically enter the read array mode by default.

CE# and OE# must be driven to be valid to enable FLASH to output data. CE# is the FLASH selection control pin. When CE# is valid, FLASH will be enabled. OE# is the data output control port.

Drives the selected memory data onto the I/O bus. For all read modes, WE# and RP# must be at VIH.

4.2.3.2. Output Shutdown

When OE# is at high level VIH, the output of FLASH will be invalid and the output will be in high impedance state.

4.2.3.3. Standby

When the CE# pin is high VIH and RP# is low VIL, the FLASH will enter standby mode. The standby mode will reduce the power consumption of the FLASH. In standby mode, the output of the FLASH will be in a high impedance state independent of OE#.

4.2.3.4. Reset

In read mode, when the low pulse time of RP# meets tPLPH, FLASH will be reset, the output will be in high impedance state and all internal circuits will be turned off. If you want to exit the reset state and return to the read mode, a recovery time of tPHQV is required. If you want to start a write operation in the reset state, a time delay of tPHWL or tPHEL is required. When CUI is reset to read array mode, the status register will become 80H.

When the low pulse time of RP# is greater than tPLPH (during programming or erasing), the programming or erasing operation will be terminated, and the content of the FLASH at the aborted location (for programming) or module (for erasing) will no longer be valid, and

the data will be partially erased or written. The timing of the abort process is: when the low pulse time of RP# meets tPLRH, the ongoing operation will be shut down. After this tPLRH time, FLASH will be reset to read mode (if RP# has become high during tPLRH) or enter the reset state (RP# is still low after tPLRH). For these two cases, if you want to restart the read or write operation, you must have a delay time of tPHQV or tPHWL/tPHEL. The relevant timing is shown in Figures 4.7, 4.8, 4.9 and Table 4.9.

For many devices (such as CPUs), it is very important to de-assert RP# during the system reset process. When the system comes out of reset, the processor will want to get the reset state of the FLASH. When reading during the erase or programming process, the FLASH provides information about the status register. If the CPU is reset without a FLASH reset, CPU initialization may not occur (because the FLASH provides register information, not array data). The FLASH allows the CPU to be initialized after a system reset through the PR# input. In this application, RP# will be controlled by the system RESET# signal to reset the system CPU. It is recommended to connect RP# to the system reset signal.

4.2.3.5. Write

The write operation is valid when CE#, WE# are low and OE# is high. The FLASH operation is controlled by writing commands to CUI through standard processor timing. CUI does not occupy the storage space of FLASH. The address and data bus are latched at the falling edge of WE#. Figure 4.6 shows the programming and erasing operations. The corresponding instructions are shown in Table 4.3.

There are two commands that change the data in the array: Program (40H) and Erase (20H). These two commands written to CUI will enable the corresponding functions (unless the operation is aborted when RP# is driven low to meet tPLRH or the related suspend instruction is written).

4.2.4. FLASH operation mode

FLASH has three read modes and two write modes. The read modes are array read, ID read and status register read. The write modes are programming and block erase. The three additional modes (erase suspend after programming, erase suspend after reading and program suspend after reading) are only valid during the suspend operation. The commands used in these modes are summarized in Table 4.2.

Table 4.2. Command code and description

Coding	Flash Mode	Describe
00,01 60,2F C0,98	No Function/Reserved Commands	These commands have no functionality. They are reserved for future functionality expansion.
FF	Read Array	Put FLASH in read array mode, in which array data will be output through the DQ pin.
40 asd	Programming establishment	This is a two-cycle command. The first cycle is to let CUI
		Prepare for programming operation. The second cycle latches the address and data information and initializes WSM to execute the programming algorithm. When CE#, OE# are triggered, FLASH will output the status register data. After programming, a read command is required to read the array data.
10	Reserved programming setup	Same as programming command
20	Erase Establishment	CUI prepares for the erase confirmation command. If the next command is not an erase confirmation command, CUI will set the status registers SR4 and SR5 to high level "1", putting the FLASH in the read status register mode and waiting for the next instruction.
D0	Erase Confirmation Program/erase recovery	If the previous command is an erase setup instruction, CUI will close the address and data latches and start erasing the specified module. During the erase process, FLASH will only respond to the read status register and erase suspend commands. When CE#, OE# is triggered, FLASH will output the data of the status register. If a program or erase operation has been suspended, this command can resume the program or erase operation.
B0	Program/Erase Suspend	Executing this instruction will suspend the program/erase operation. When the suspend is successful, the status register (Program Suspend (SR2), Erase Suspend (SR6)) will show the corresponding status, and the WSM status bit (SR7) will become high level "1" (ready). Except for turning off the WSM to reset the FLASH when RP# is low, the WSM will remain in the SUSPEND state and will not be affected by all input control pins (except RP#).
70	Read Status Register	This command puts the FLASH into the mode of reading the status register. When reading the device, the address will be ignored and the contents of the status register will be output. The FLASH will automatically enter this mode after a program or erase operation is initiated.
50	Clear Status Register	WSM can compare SR1, SR3, SR4 in the status register with
		SR5 is set to high level "1" and cannot be cleared. Executing this

		command can clear these status register bits.
90	Read ID	Put the device in read ID mode, so when reading the device, the manufacturer and device identification codes will be output (when A0=0, it is the manufacturer, when A0=1, it is the device, and all other address inputs must be 0).

4.2.4.1. Read Array

When RP# changes from VIL (reset) to VIH, FLASH will enter the read array mode by default without any CUI command and respond to the read control signals (CE#, address input and OE#).

When the FLASH is in array read mode, four control pins control the output of data:

- WE# must be high level (Vih)
- CE# must be low (Vil)
- OE# must be low (Vil)
- RP# must be high level (Vih)

Furthermore, the corresponding address must be input from the address pins. If the FLASH is not in read mode (such as after an erase and program operation), the read array command FFH must be sent through the CUI to enter the read array mode.

4.2.4.2. Read ID

If you want to read the manufacturer identification code (0089H) and device identification code (8891H) of FLASH, you must write the read ID instruction (90H) to put the device in the read ID mode. In the read ID mode, when A0=0, the output is the manufacturer identification code, and when A0=1, the output is the device identification code. When reading ID, A1-A19=0. If you want to return to the read array mode, you need to write the read array instruction (FFH).

4.2.4.3. Read Status Register

The status register of FLASH indicates whether programming or erasing is completed, success or failure, etc. Entering the read status register instruction (70H) to CUI can read the value of the status register. Unless a new instruction is entered, the read operation after entering the read status register mode reads the value of the status register. If you want to enter the read array mode, you need to enter the read array instruction (FFH).

The status register bits are output on the data port DQ[7:0], and the other bits DQ[15:8] will output 00H in the read status register mode.

The contents of the status register will be latched at the falling edge of OE# or CE#. CE# and OE# must be valid to correctly read the value of the status register, thereby displaying the working status of the FLASH.

When WSM is valid, SR7 will reflect the status of WSM, and other status register bits will reflect whether WSM successfully completed the desired operation (as shown in Table 4.4).

4.2.4.4. Clear Status Register

The FLASH WSM will set the status register SR[7:0] to "1" and clear bits SR2, SR6, and SR7 to "0". Because SR1, SR3, SR4, and SR5 represent error conditions, these bits can only be cleared by the clear status register instruction (50H). After an error occurs, these bits should be cleared before starting other operations.

4.2.4.5. Programming Mode

The programming operation requires two write cycles. When the programming setup instruction (40H) is written into CUI, the second write needs to specify the address and data you want to write. WSM will automatically execute the corresponding internal timing to write the specified data to the specified address. Programming FLASH can only change the high level to the low level, and cannot change the low level to the high level. If you try to program "0" to "1", the content of the storage unit will not change and no error will occur.

The status register will indicate the programming status: when programming is in progress, SR7 is always "0". The value of the status register can be read by triggering CE# and OE#. When programming is in progress, the only valid instructions are the read status register instruction, the programming suspend instruction and the programming resume instruction. The flowcharts for programming and reading the status register are shown in Figures 4.2 and 4.3 respectively.

When programming is complete, the programming status bits should be checked. If the programming operation fails, SR4 will go high to indicate programming failure. If SR3 is set high (VCCW is not in the normal operating range), the WSM will not execute the corresponding programming instructions. If SR1 goes high, programming of the locked

module will be aborted and fail.

Before starting other operations, the status register should be cleared. After programming is completed, other CUI instructions can be executed. However, if you want to exit the read status register mode, you can reset the CUI to read array mode.

Bus Operation	Order	Notes
Write	Programming	DQ=40H
Write	programming	DQ = Programming Data ADDR = Programming address
read		Enable CE# and OE# to read status Status register value
wait		View SR7 1=WSM completed 0=WSM busy
The above programming operation can be repeated. The status register check can be performed after programming is complete or after a lot of programming is complete. After the last programming is completed, write the command FFH to return the FLASH to the array read mode.		

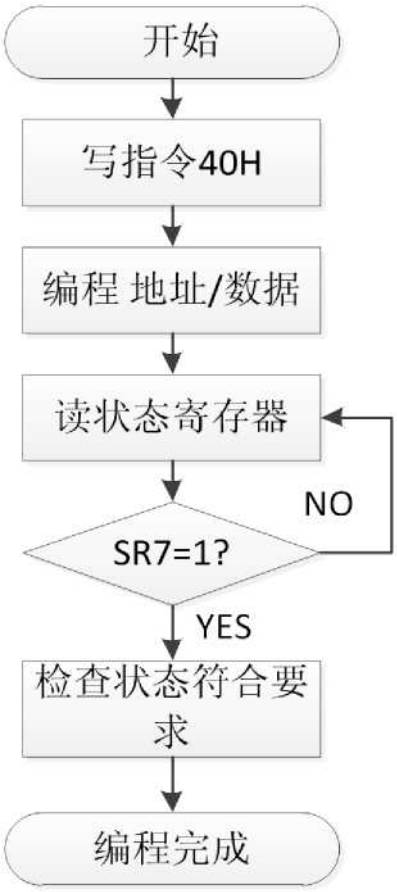


Figure 4.2 Programming flow chart

Bus Operation	Order	Notes
wait		View SR3 SR3=1 VPP is low level
wait		View SR4 SR4=1 Programming error
wait		View SR1 SR1=1 Attempt to program locked module - programming aborted
SR1, SR3, and SR4 can only be cleared by the Clear Status Register instruction. If an error is detected, the status register needs to be cleared before other operations.		

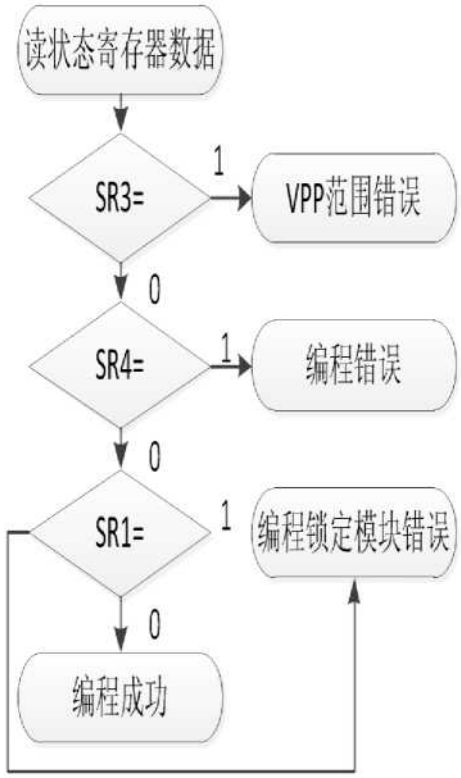


Figure 4.3. Reading status register flow chart

4.2.4.6. Suspend and resume programming

Program Suspend will terminate the programming operation so that data at another address can be read. Once the programming operation starts, the WSM will suspend the programming operation after the Program Suspend command is written through the CUI. After Program Suspend, the FLASH will continue to output the value of the status register. By checking the status registers SR7 and SR2, you can check whether the Program Suspend is completed (SR2 and SR7 will be set to "1"). tWHRH1/tEHRH1 indicates the Program Suspend delay.

The read array instruction can be written into CUI to read the array data of other modules except those that have been suspended. The valid instructions after programming suspension are: read status register, read ID and programming recovery. After the programming recovery instruction is written into FLASH, WSM will continue the programming process, and the status registers SR2 and SR7 will be cleared. After the programming recovery command is written, when reading, the device will automatically

output the value of the status register (see Figure 4.4 Programming Suspend and Programming Resume Flowchart). In programming suspension mode, VCCW must maintain the same voltage. RP# must remain high.

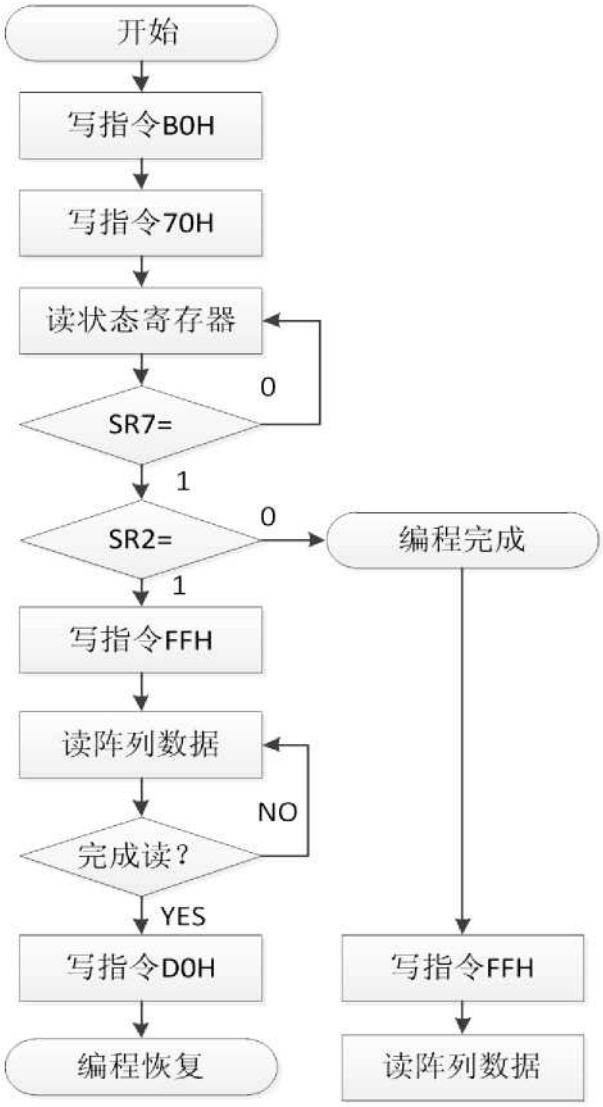


Figure 4.4. Programming Suspend and Resume Flowchart

4.2.4.7. Erase Mode

Send the erase setup and confirmation command to CUI, and send the module address you want to erase, then you can erase the module. When the erase confirmation command is executed, the module address will be latched internally. The result of module erasure is that all the used BIT bits of the module become "1". Only one module can be erased at a time. WSM will erase all BIT bits to "1". When the erase is executed, the status

register bit SR7 is "0".

When the status register SR7 indicates that the erase is complete, check the erase status bit to confirm that the erase operation was successful. If the erase fails, SR5 will become "1", indicating that the erase failed. If VPP is not within the working range after the erase confirmation instruction, the WSM will not perform the erase operation, and SR5 will become "1" to prove that the erase failed, and SR3 will become "1" to indicate that VCC is not within the working range. After the erase operation, the status register should be cleared before starting the next operation. After the erase, many CUI instructions can follow; however, in order to protect the status register from unwanted reads, it is recommended to put the FLASH in read array mode after the erase is completed.

4.2.4.8. Suspend and resume programming

Since the erase operation requires two steps to complete, the erase suspend instruction can be used to interrupt the erase operation and then read data or program other modules. Once the erase operation starts, the erase suspend instruction can be written through the CUI to let the WSM suspend the erase operation through a certain algorithm. The status register can indicate whether the erase suspend has been completed.

Read array/program instructions can be written to CUI after Erase Suspend to read data/program addresses of other modules. Programming can also be suspended subsequently to read data from other addresses. When Erase Suspend is complete, valid instructions are Erase Resume, Program, Read Array, Read Status Register and Read ID.

After the erase is complete, the status register should be read and cleared before starting the next instruction.

Table 4.3. (1,4)

Order	Note	First bus cycle			Second bus cycle		
		operate	address	data	operate	address	data
Read Array		Write	X	FFH			
Read ID	2	Write	X	90H	read	I A	ID
Read Status Register		Write	X	70H	read	X	SRD
Clear Status Register		Write	X	50H			

asd

Device							

programming	3	Write	X	40/10H	Write	PA	PD
Module Erase/Confirm		Write	X	20H	Write	BA	DOH
Program/Erase Suspend		Write	X	B0H			
Program/erase recovery		Write	X	D0H			

Note：

PA:Programming Address PD:Programming Data BA: module address

IA:IDaddress ID:Identification code SRD: Status Register Data

- The bus operation definition is shown in Table 4.1.
- A0=0, the output is the manufacturer identification code, A0=1, the output is the device identification code.
- 40H or 10H instructions are both acceptable, but the standard programming instruction is 40H.
- 4When writing instructions to FLASH, the upper 8 bits DQ[15:8] of the data bus should be high or low to reduce power consumption.

Table 4.4. Status register definition

WSMS	ESS	ES	PS	VPPS	PSS	BLS	R
7	6	5	4	3	2	1	0

	Note:
SR7 = Write State Machine WSM Status (WSMS) 1 = Complete 0 = Busy	Before checking the programming or erasing status bits, check the write status machine bits to see if programming or erasing has completed.
SR6 = Erase Suspend State (ESS) 1 = Erase Suspend 0 = Erase in progress/completed	When the Erase Suspend command is executed, WSM will suspend the erase operation and change the WSMS and ESS bits to "1". The ESS bit will remain "1" until the Erase Resume command is executed.
SR5 = Erase Status (ES) 1 = Module erase error 0 = Module erased successfully	When this bit is "1", it indicates that an error occurred in module erasure.
SR4 = Programming State (PS)	When this bit is "1", the WSM has attempted but

1 = Word programming error	failed to program a word.

0 = word programming successful	
SR3 = VPP state (VPPS) 1 = VPP is low, operation is terminated 0=VPP normal	When this bit is "1", it indicates that VPP is not in the normal operating range and the erase or program operation cannot be performed normally.
SR2 = Program Suspend State (PSS) 1 = Programming paused 0 = Programming in progress/completed	When the Program Suspend command is executed, WSM will terminate the programming operation and change the WSMS and PSS bits to "1". Before the Program Resume command is executed, the PSS bit is always "1".
SR1 = module lock status 1 = Program/Erase Lock Module Abort 0 = No action on locked module	When erasing or programming the locked module, this bit will be set to "1" by WSM. The corresponding operation will be aborted and the device will return to the mode of reading the status register.
SR0 = reserved for future function expansion	No function, this is reserved for future application expansion.

4.2.5. FLASH module lock

There are two ways to lock the module in FLASH.

4.2.5.1. WP#=VIL locks the module

When WP#=VIL, the 8 lockable parameter modules of FLASH can be locked. Programming or erasing these locked modules will cause errors, which will be reflected in the corresponding status registers. Other modules that are not locked can be erased or programmed normally (unless VPP is not within the normal working range).

4.2.5.2. WP#=VIH module is not locked

WP#=VIH will not lock the lockable modules. At this time, these 8 parameter modules can be erased and programmed.

WP# provides protection for the parameter module, and RP# avoids unwanted write operations.

Table 4.5. Write protection truth table

VPP	WP#	RP#	Write protection function
X	X	VIL	All modules locked
VIL	X	VIH	All modules locked

High level	VIL	VIH	Parameter module lock
High level	VIH	VIH	All modules are unlocked

4.2.6. FLASH AC parameters

4.2.6.1 读操作

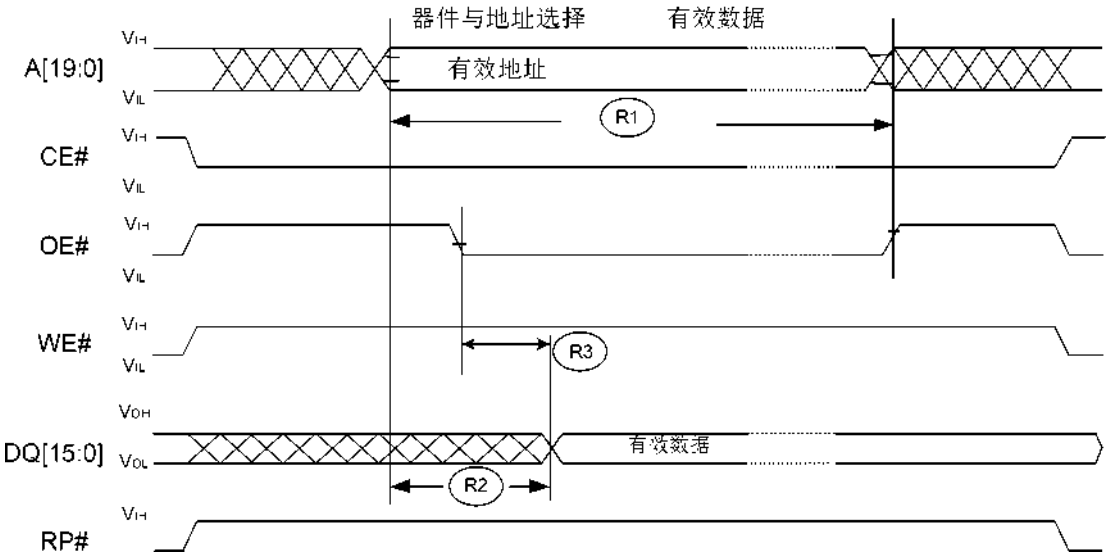


Figure 4.5. Read operation timing diagram

Table 4.6. Read operation AC parameters

#	symbol	describe	Minimum	maximum	unit
R1	tAV	Read cycle time	90	-	ns
R2	QUR	Delay time from address setup to output validity	-	90	ns
R3 ASD	QUR	Delay time from OE establishment to output validity	-	90	ns

4.2.6.2. Write Operation

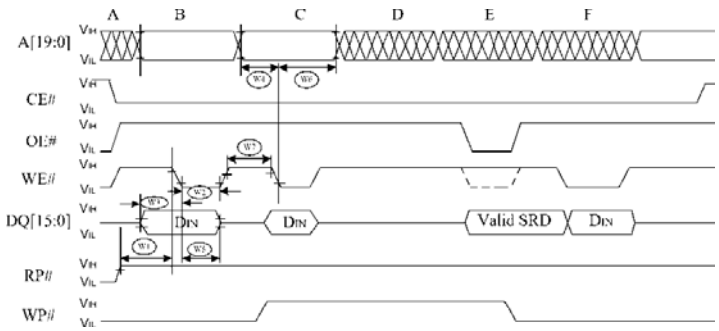


Figure 4.6. Write

Table 4.7. Write operation AC parameters

#	Symbol	Describe	most Small	one Bit
W1	oeLh	The time from RP changing to high level to WE changing to low level	600	ns
W2	oeLh	WE pulse width	70	ns
W3	oeLh	The time from when data is established to when WE becomes low	0	ns
W4	oeLh	The time from address establishment to WE becoming low	0	ns
W5	wxya	Data retention time after WE becomes low level	70	ns
W6	WX	The address holding time after WE becomes low	70	ns
W7	wxya	WE high level pulse width	30	ns

4.2.6.3. Programming and Erase Time

Table 4.8 Programming and erasing time

Symbol	Describe	Typical	Maximum	Unit
tWHQV1/ tEHQV1	Programming time for one word	70	200	US
tWHQV2/ tEHQV2	4-KW parameter module erase time	0.5	4	S
tWHQV3/ tEHQV3	32-KW Master Module Erase Time	1	5	S
tWHRH1/ oeLh	Programming Pause Delay Time	70	200	US
tWHRH1 oeLh	Erase Suspend Delay Time	1	5	S

4.2.6.4 Reset Parameters

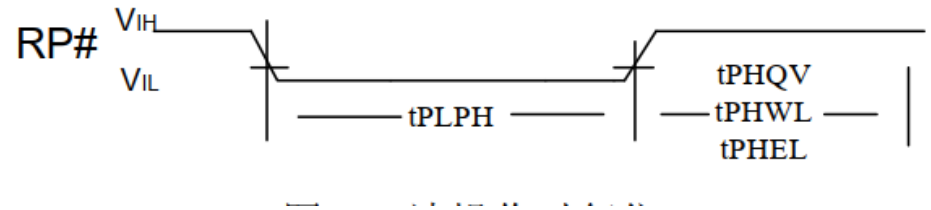


Figure 4.7 Reset during read operation

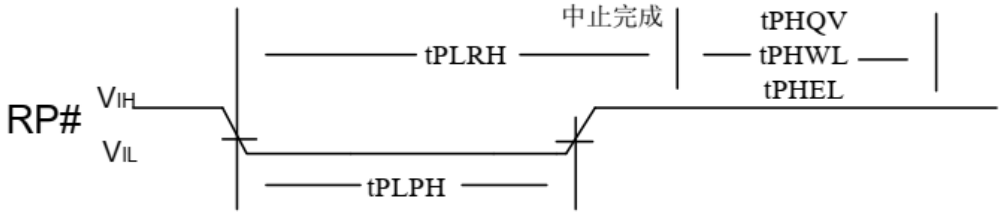


Figure 4.8. Reset during programming or erasing, tPLPH≤tPLRH

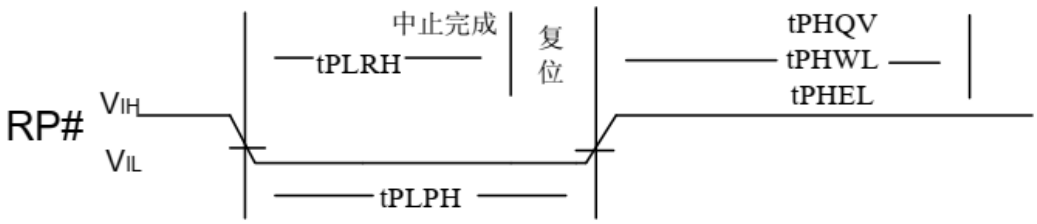


Table 4.9. Reset parameters

Symbol	Describe	Minimum	Maximum	Unit
tPH	RP resets the low level during the read operation. Bit Time	100	-	ns
tPLRH	RP during module erasing and programming	-	twenty two	us
	Low level reset time			

4.2.6. FLASH module address distribution

Table 7.1. FLASH module address distribution

Size (KW)	Address
32	F8000-FFFF
32	F0000-F7FFF
32	E8000-EFFFF
32	E0000-E7FFF
32	D8000-DFFFF
32	D0000-D7FFF
32	C8000-CFFFF
32	C0000-C7FFF
32	B8000-BFFFF
32	B0000-B7FFF
32	A8000-AFFFF
32	A0000-A7FFF
32	98000-9FFFF
32	90000-97FFF
32	88000-8FFFF
32	80000-87FFF
32	78000-7FFFF
32	70000-77FFF
32	68000-6FFFF
32	60000-67FFF
32	58000-5FFFF
32	50000-57FFF
32	48000-4FFFF
32	40000-47FFF
32	38000-3FFFF
32	30000-37FFF
32	28000-2FFFF
32	20000-27FFF
32	18000-1FFFF
32	10000-17FFF
32	08000-0FFFF
4	07000-07FFF

4	06000-06FFF
4	05000-05FFF
4	04000-04FFF
4	03000-03FFF
4	02000-02FFF
4	01000-01FFF
4	00000-00FFF

V. Pin Description

Figure 5.1 shows the bottom schematic diagram of the BSTEPC16 package UFBGA88. Table 5.1, Table 5.2 and Table 5.3 are the pin descriptions of BSTEPC16, including configuration interface pins, external FLASH interface pins, JTAG interface pins and other pins.

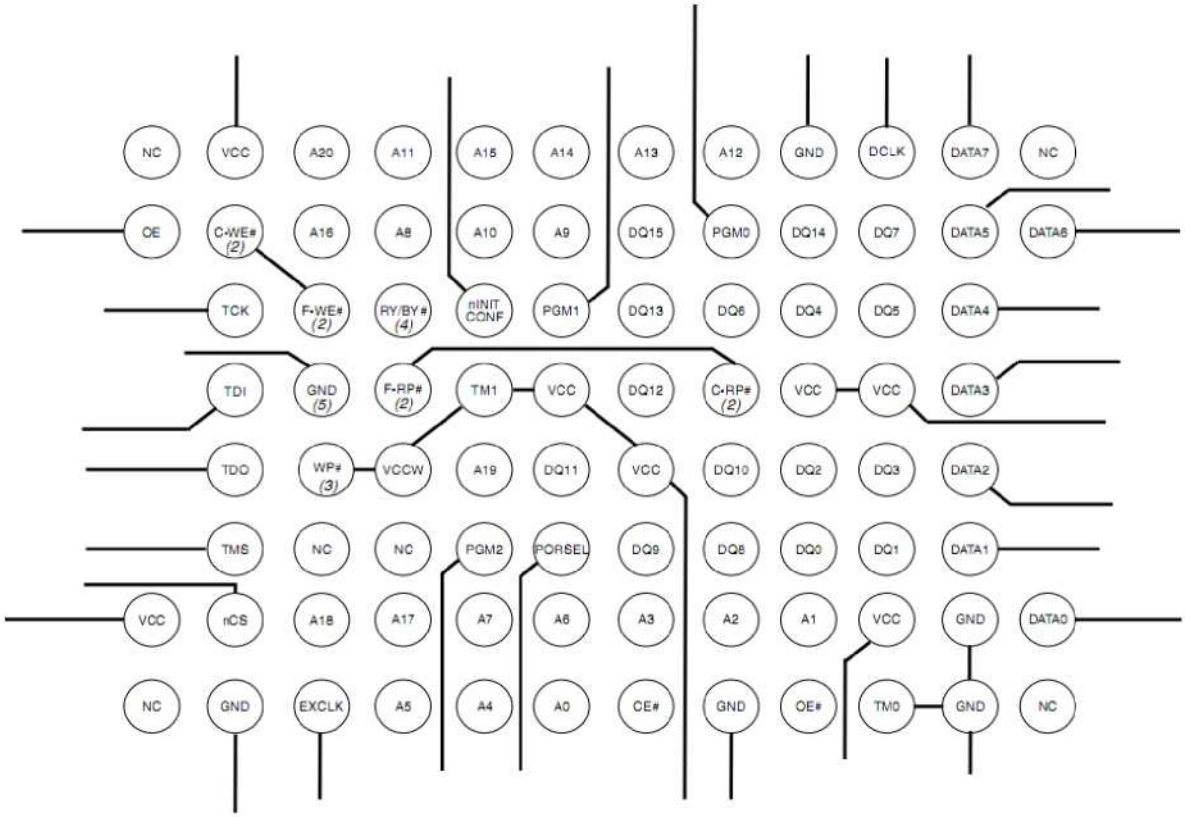


Figure 5.1 Terminal arrangement

Table 5.1. Configuration interface pins

Pin Name	Pin Type	Describe
DATA[7:0]	Output	Configure the data output bus. Data changes on every falling edge of DCLK. Data is latched into the FPGA on the rising edge of DCLK.
DCLK	Output	As the configuration clock of the FPGA, data is latched into the FPGA on the rising edge of DCLK.
nCS	enter	This pin is the input pin of BSTEPC16 and is connected to the CONF_DONE pin of the FPGA. It can be used for error detection after all configuration data has been transferred to the FPGA. When nCONFIG is pulled low, the FPGA will drive nCS and
		OE is low. nCS has a programmable pull-up resistor inside, which can be disabled or enabled in Quartus II by using the Disable nCS and OE pull-ups on configuration device option.
nINIT_CONF	Open-drain output	nINIT_CONF can be connected to the nCONFIG pin of the FPGA. BSTEPC16 can be used to initialize and configure the FPGA via private JTAG instructions. This pin has an internal pull-up resistor that is always valid. If this function is not used, the INIT_CONF pin can be left floating. If nINIT_CONF is not used, nCONFIG must be pulled up to Vcc directly or through a pull-up resistor.
OE	Open drain bidirectional	This pin is driven low when power-up is not complete. A user-selectable 2ms or 100ms delay keeps OE low during initial power-up, allowing for more stable voltages. The power-on reset delay can be extended by holding OE low. OE is connected to the nSTATUS signal of the FPGA. After the controller releases OE, it waits for nSTATUS-OE to go high before the FPGA begins configuration. This pin has a programmable internal pull-up resistor that can be disabled or enabled in the Quartus II software by using the Disable nCS and OE pull-ups on configuration device option.

Table 5.2. External FLASH interface pins

Pin Name	Pin Type	describe
A[19:0]	enter	FLASH address bus. These pins are used for The address line when the FLASH performs read and

		write operations. The address line will be internally latched during a write cycle. If the external FLASH interface is not used, this pin is left floating on the board.
		The address, data and control pins of these FLASH are internally connected to the controller.
DQ[15:0]	Bidirectional	FLASH data bus. These pins are the data bus that connects the FLASH to the controller. The controller or external FLASH interface drives DQ[15:0] during the FLASH command and data write cycle. If the external FLASH interface is not used, this pin is left floating on the board.
CE#	enter	FLASH chip select signal, low effective. This FLASH input pin is internally connected to the controller.
RP#	enter	The reset pin of FLASH is valid at low level. When this pin is high, normal operation is realized. When it is low, the FLASH cannot be written. It can protect the data when the power supply changes. The FLASH RP is not connected to the controller internally. When the external FLASH interface is not used, the F-RP and C-RP must be connected together on the board. If using the external FLASH interface, the external device is connected to RP#.
OE#	enter	FLASH output enable, low effective, when low, enable FLASH output. If the external FLASH interface is not used, this pin is left floating on the board.
WE#	enter	FLASH write enable signal, low effective, data and address are latched into FLASH at the falling edge of WE. This pin is not connected to the internal FLASH and controller. When the external FLASH interface is not used, F-WE and C-WE must be connected together on the board. If using the external FLASH interface, the external device is connected to WE#.
WP#	enter	This pin is usually connected to power or ground on the circuit board. When it is high, it allows erasing and programming of the FLASH parameter module. The controller does not drive this pin, even when the external FLASH interface is not used, this pin should be connected to VCC.
Vx	power supply	Module erase, chip erase, write operation power supply, connected to 3.3v power supply.
Vcc	power supply	Core and I/O Power Supply

RY/BY#	Open-drain output	Displays the status of internal FLASH writing or erasing. It is high when writing or erasing is completed, and low when writing or erasing is in progress. If this pin is used, an external resistor needs to be connected, otherwise it should be left floating.

Table 5.3. JTAG interface pins and other required control pins

Pin Name	Pin Type	Describe
TDI	enter	JTAG data input pin. Connect to VCC when the JTAG circuit is not used.
TDO	Output	JTAG data output pin. When JTAG is not used, leave this pin unconnected.
TCK	enter	JTAG clock input pin. Connect to GND when the JTAG circuit is not used.
TMS	enter	JTAG mode selection pin. Connect to VCC when JTAG is not used.
PGM[2:0]	enter	Input pin selects one of eight pages of configuration information for FPGA configuration. Configuration page information in Quartus II
		Generate. PGM[2] is the highest bit. By default, Page 0 is selected (PGM[2..0]==000). These pins cannot be left floating.
EXCLK	enter	Optional external clock input pin that can be used to generate the configuration clock DCLK. Select whether to use this function in the Quartus II software. When this function is not used, connect EXCLK to a fixed level.
PORSEL	enter	This pin selects 2ms or 100ms power-on reset counter delay during power-up. When PORSEL is low, the power-on reset time is 100ms; when PORSEL is high, the power-on reset time is 2ms. This pin must be connected to a fixed level.
TM0	enter	Test pin. When used by the user, this pin must be connected to ground.
TM1	enter	Test pin. This pin must be connected to the power supply when used by the user.
TEST_MODE	enter	Test pin. When used by the user, this pin must be connected to ground.

VI. Power-on reset circuit and power-on sequence

5.1. Power-On Reset

The power-on reset circuit includes: a power-on reset circuit and a delay circuit. The power-on reset circuit will put the system in a reset state before VCC reaches the operating voltage during the power-on process. The time of power-on reset and delay depends on the time it takes for VCC to linearly climb to the voltage range where the device can work normally within a certain range of rising slopes and the user-programmable power-on delay circuit. The delay circuit delays the global reset signal by 2ms (fast power-on) or 100ms (slow power-on) according to the external pin PORSEL. The power-on reset delay time can be extended by driving OE low.

During the VCC power-on process, when the VCC voltage is greater than VCCPOR, the chip starts to work and enters the reset state. During the power-on reset delay process, the chip drives the OE pin to a low level. After the power-on reset is completed (VCC reaches the operating voltage and the delay circuit is calculated), the chip will release the OE pin (OE has an internal programmable pull-up resistor, which needs to be set in the Quartus II software. The default is to have a pull-up resistor. If the internal pull-up resistor is canceled, it will be invalid. A 10KΩ pull-up resistor needs to be added externally). The following figure shows the power-on reset requirements, which includes the delay of OE during fast power-on and slow power-on.

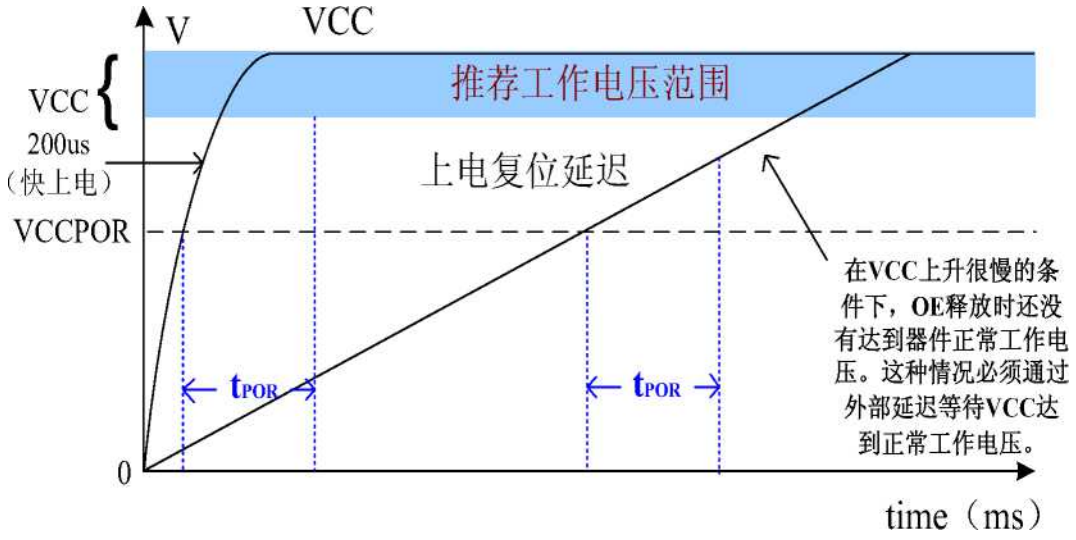


Figure 6.1. Power-on reset requirements

Table 6.1. Power-on reset parameters

Signal	Describe	Typical Value		unit
VCCPOR	VCC Power-On Reset Threshold Pressure	1.6		V
tPOR Asasd	After VCC reaches VCCPOR OE release delay time	PORSEL=0	100	ms
		PORSEL=1	3	

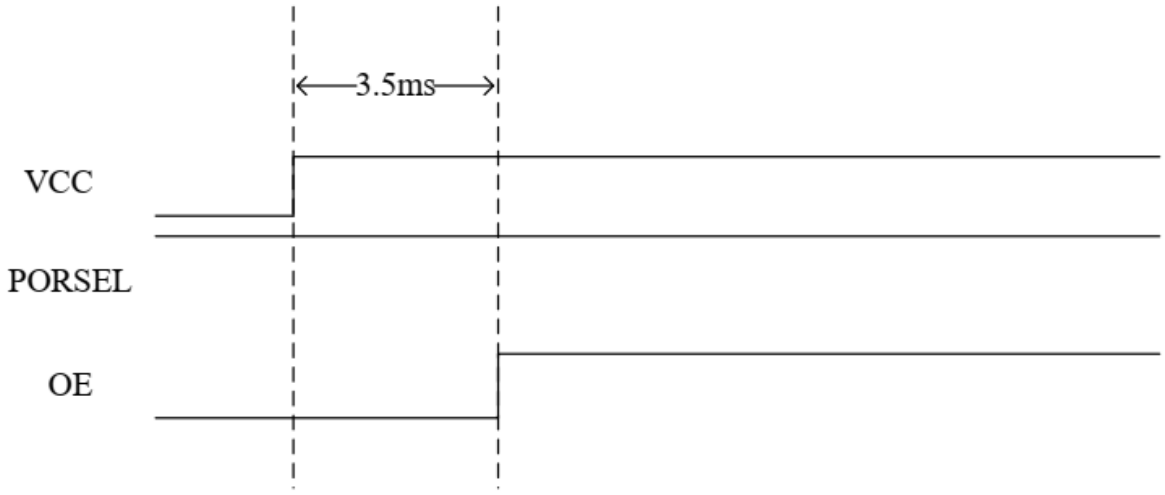


Figure 6.2. Changes in the power-on reset OE pin (when PORSEL is 1)

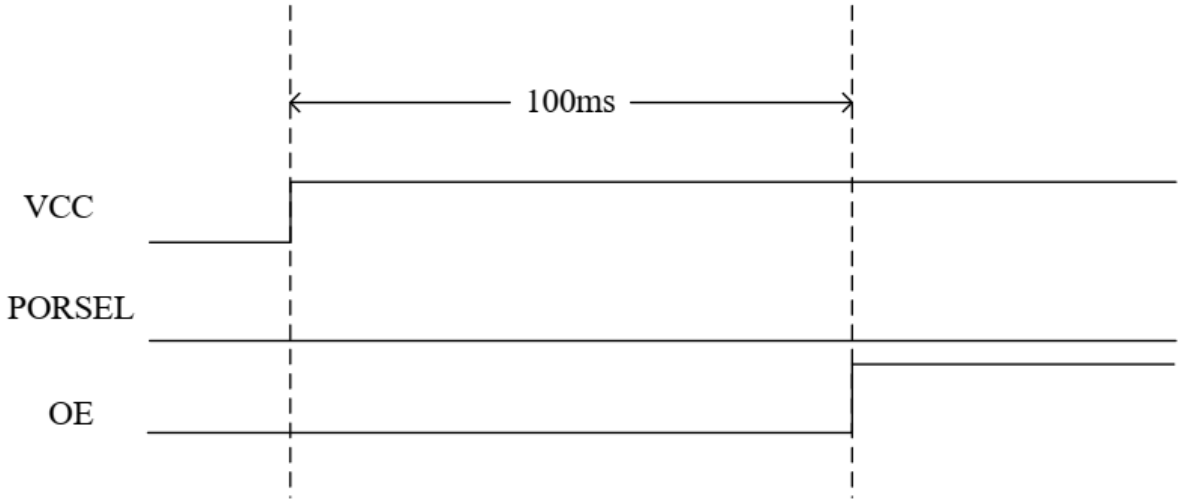


Figure 6.3. Changes in the power-on reset OE pin (when PORSEL is 0)

The power-on reset circuit is very important for BSTEPC16, because after the power-on reset, BSTEPC16 needs to configure its internal registers. If the voltage value of the VCC

power supply is low at the end of the power-on reset, it may cause abnormal configuration of the internal registers. Therefore, the power-on reset delay can delay the process of configuring the internal registers of the device, so that the internal registers are configured only when the VCC power supply voltage reaches a higher voltage value, thereby widening the device's adaptability to the power-on timing. In addition, during the power-on reset process, the OE pin is always driven to a low level, which can effectively cooperate with the FPGA device, because the OE pin is connected to the nSTATUS pin of the ALTERA FPGA, and its low level can also delay the FPGA startup configuration, that is, as long as the power-on reset is not completed, the FPGA will not initiate the configuration process. The configuration process may not start until the BSTEPC16 completes the power-on reset, which can ensure that the correct configuration data is provided to the FPGA.

5.2. Power-on sequence

The recommended power-on sequence is: the FPGA's VCCINT power-on should be completed before the BSTEPC16 power-on reset delay ends.

It is necessary to control the power-on so that the OE signal of BSTEPC16 becomes high after the CONF_DONE signal of FPGA becomes low. If BSTEPC16 completes the power-on reset before the FPGA is powered on, the CONF_DONE signal of FPGA will be pulled high by the pull-up resistor. When BSTEPC16 completes the power-on reset, the OE signal will be released and pulled high by the pull-up resistor. Since BSTEPC16 samples the nCS signal at the rising edge of OE, if a high level on nCS is sampled, there will be no output of the configuration port DATA and DCLK, and thus the configuration will not start. BSTEPC16 can only exit this incorrect configuration state by powering off and then powering on correctly.

In order to ensure that BSTEPC16 can enter the configuration mode correctly, it is necessary to ensure that the FPGA has been powered on before BSTEPC16 completes the power-on reset.

Another option is to hold the FPGA's nCONFIG pin low externally before the FPGA and BSTEPC16 are powered on. This ensures the correct power-on sequence and thus the correct configuration.

VII. Working conditions and DC characteristics

Table 7.1 BSTEPC16 absolute maximum ratings

Symbol	Parameter	Condition	Minimum	Maximum	Unit
VCC	power supply	---	-0.2	4.6	V
VI	DC input voltage	---	-0.5	3.6	V
IMAX	DC VCC or ground current	---	---	100	mA
IOUT	Each pin has a DC output Current	---	-25	25	mA
PD	Power consumption	---	---	360	mW
TSTG	Storage temperature	No bias	-65	150	℃
TJ	Temperature section	Under bias	---	175	℃

Table 7.2. BSTEPC16 recommended operating conditions

Symbol	Parameter	Condition	Minimum	Maximum	Unit
VCC	Supply voltage	---	3.0	3.6	V
VI	Input voltage	---	-0.3	VCC+0.3	V
VO	Output voltage	---	0	VCC	V
TA	Operating temperature	---	-55	125	℃
TR	Input rise time	---	---	20	ns
TF	Input Fall Time	---	---	20	ns

Table 7.3. BSTEPC16 DC operating conditions

			EPC16 (original)			BSTEPC16			
Symbol	Parameter	Condition	Minimum	Typical	Maximum	Minimum	Typical	Maximum	Unit
VCC	power supply	---	3.0	3.3	3.6	3.0	3.3	3.6	V
VIH	Input high level	---	2.0	---	VCC +0.3	2.0	---	VCC +0.3	V
VIL	Input low level	---	---	---	0.8	---	---	0.8	V
VOH	High level output Voltage	IOH=－4mA	2.4	---	---	2.4	---	---	V
VOL	Low level	IOL= -4mA	---	---	0.45	---	---	0.4	V

	output Voltage								
II	Input drain current flow	VI= VCCor	-10	---	10	-10	---	10	uA
ICCO	Power Static flow	---	---	---	---	---	---	6	mA
ICC1	Power Dynamics flow	---	---	60	90	---	50	90	mA

VIII. JTAG and AC parameters

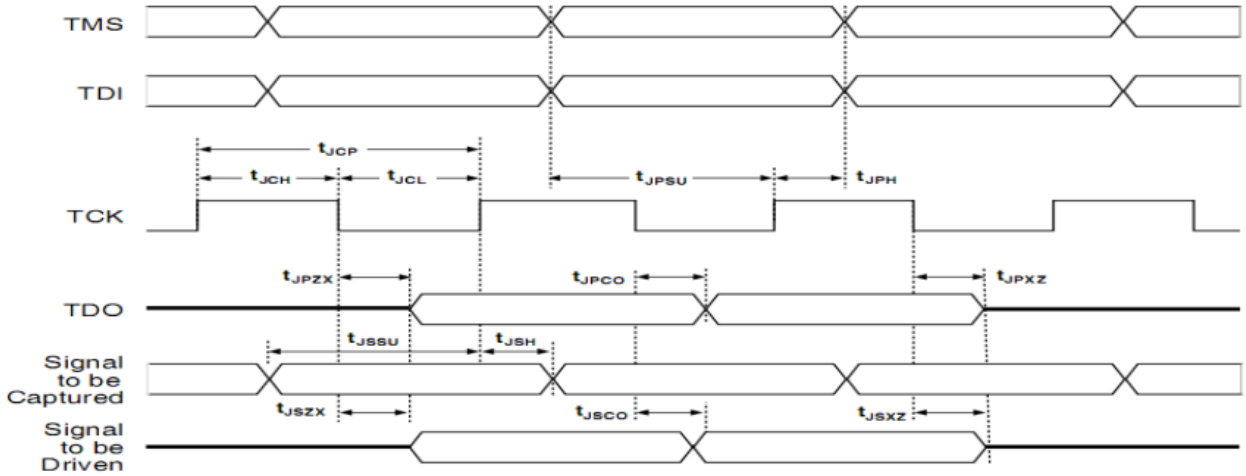


Figure 8.1. JTAG timing

Table 8.1. BSTEPC16 JTAG configuration timing

Symbol	Characteristic	Epc16 (Original)		BSTEPC16		unit
		Minimum	Maximum	Minimum	Maximum	
tJCP	TCK clock period	100	—	100	—	ns
tJCH	TCK clock high time	50	—	50	—	ns
tJCL	TCK clock low time	50	—	50	—	ns

tJPSU	JTAG port setup time	20	—	20	—	ns
tJPH	JTAG port hold time	45	—	45	—	ns
tJPCO	JTAG port clock output	—	25	—	25	ns
tJPZX	JTAG port TDO high impedance to effective value output	—	25	—	25	ns
tJPXZ	The effective value of JTAG port TDO is output to high impedance	—	25	—	25	ns
tJSSU	Capture register setup time	20	—	20	—	ns
tJSH	Capture register hold time	45	—	45	—	ns
tJSCO	Update register clock to output	—	25	—	25	ns
tJSZX	Update register high impedance to effective value output	—	25	—	25	ns
tJSXZ	Update register effective value output to high impedance	—	25	—	25	ns

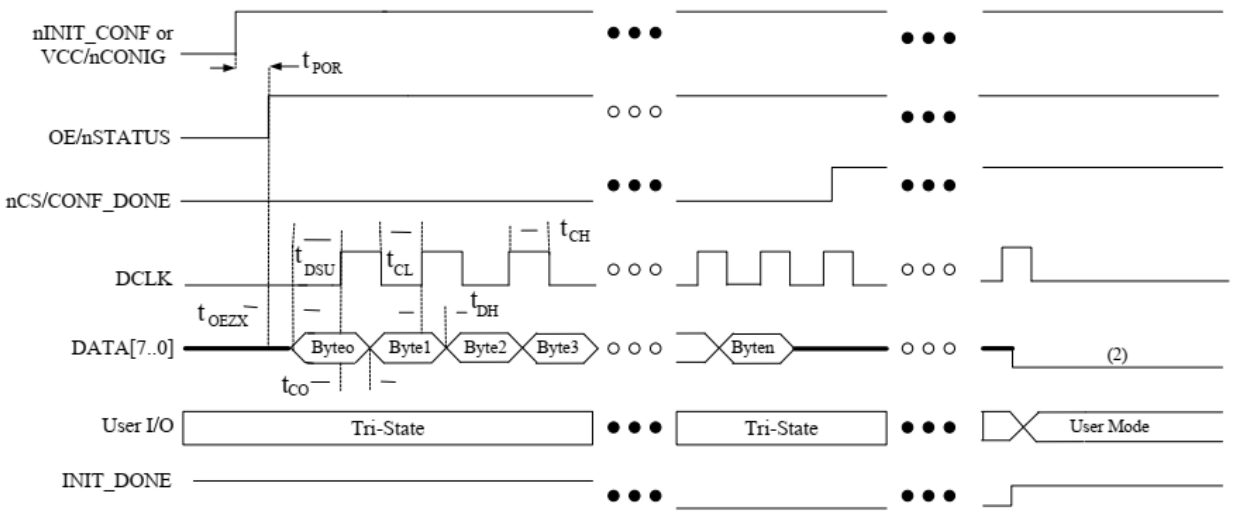


Figure 8.2 Configuration sequence using BSTEPC16

Notes to Figure 4:

- The configuration device drives DCLK low after configuration is complete.
- The configuration device will drive DATA to a high level after configuration is completed.

Table 8.2. BSTEPC16 configuration parameters

Symbol	Parameter	Condition	Epc16 (Original)			BSTEPC16			Unit
			Minimum	Typical	Maximum	Minimum	Typical	Maximum	
fDCLK	DCLK Frequency Rate	40% Empty ratio			66.7			66.7	MHz
tDCLK	DCLK Week Expect		15			15			ns
tHC	DCLK high level time	40% Empty ratio	6			6			ns
tLC	DCLK low level time	40% Empty ratio	6	—	—	6	—	—	ns
tCE	OE to one Delay of DCLK	—	40	—	—	40	—	—	ns
Asd									
tOE	OE toAn effectiveDATA	—	40	—	—	40	—	—	ns
tOH	DCLK rising edge to count According to changes			(1)			(2)		ns
tLOE	OE Invalid Until reset is completed become		60			60			ns
fECLK	EXCLK Input frequency	40% Empty ratio	—	—	100	—	—	100	MHz

tECLK	EXCLK Input cycle		10			10			ns
tECLKH	EXCLK Input high voltage Flat time	40% Empty ratio	4	—	—	4	—	—	ns
tECLKL	EXCLK Input low voltage Flat time	40% Empty ratio	4			4		—	ns
tECLKR	EXCLK Input Rising time	100MHz			3			3	ns
tECLKF	EXCLK Input drop time	100MHz			3			3	ns
tPOR	Power-On Reset Delay time	2ms	1	2	3	1	3	4	ms
		100ms	70	100	120	70	100	120	ms

Note:

(1) tOH=0.5(DCLK period)-2.5;

(2) tOH = 0.5 (DCLK period);

IX. Detailed Functional Description

9.1. FPGA Configuration

The FPGA configuration is controlled by the controller. The entire operation process includes reading the configuration data from the FLASH memory, decompressing (if necessary), transmitting the data through the pin DATA[], (if an error occurs during configuration) and handling the configuration error.

After power-on, the controller reads 32 words of user-defined option information from FLASH, including configuration scheme, configuration clock speed, whether to decompress, and configuration page selection.

After reading these option bits, the controller detects whether the FPGA is ready to receive

data by monitoring nSTATUS and CONF_DONE. When the controller detects that the FPGA is ready to receive (nSTATUS is high and CONF_DONE is low), the controller transmits data using pins DCLK and DATA[]. After power-on reset or reset, the controller selects the configuration page data to transmit to the FPGA by sampling PGM[2:0].

Depending on the configuration scheme, the configuration unit transfers uncompressed data to the FPGA. BSTEPC16 supports four simultaneous configurations with n=1, 2, 4 or 8. (n bits are transferred on DATA[] per DCLK cycle). n=1 is the same as the traditional PS configuration scheme. n=2, 4 or 8 are PS modes with different configuration chains. BSTEPC16 also supports FPP fast parallel slave mode, which means that 8 bits of data are transferred to the FPGA per DCLK. Depending on the different configuration bus widths (n), the circuit transfers uncompressed configuration data to the valid DATA[n] pins. Unused DATA[] pins are driven low.

The controller configuration section handles errors that occur during the configuration process. If the CONF_DONE signal of the FPGA does not go high after the configuration data is transferred, a configuration error occurs. The controller generates a low pulse on the OE signal and pulls nSTATUS low, triggering another configuration cycle.

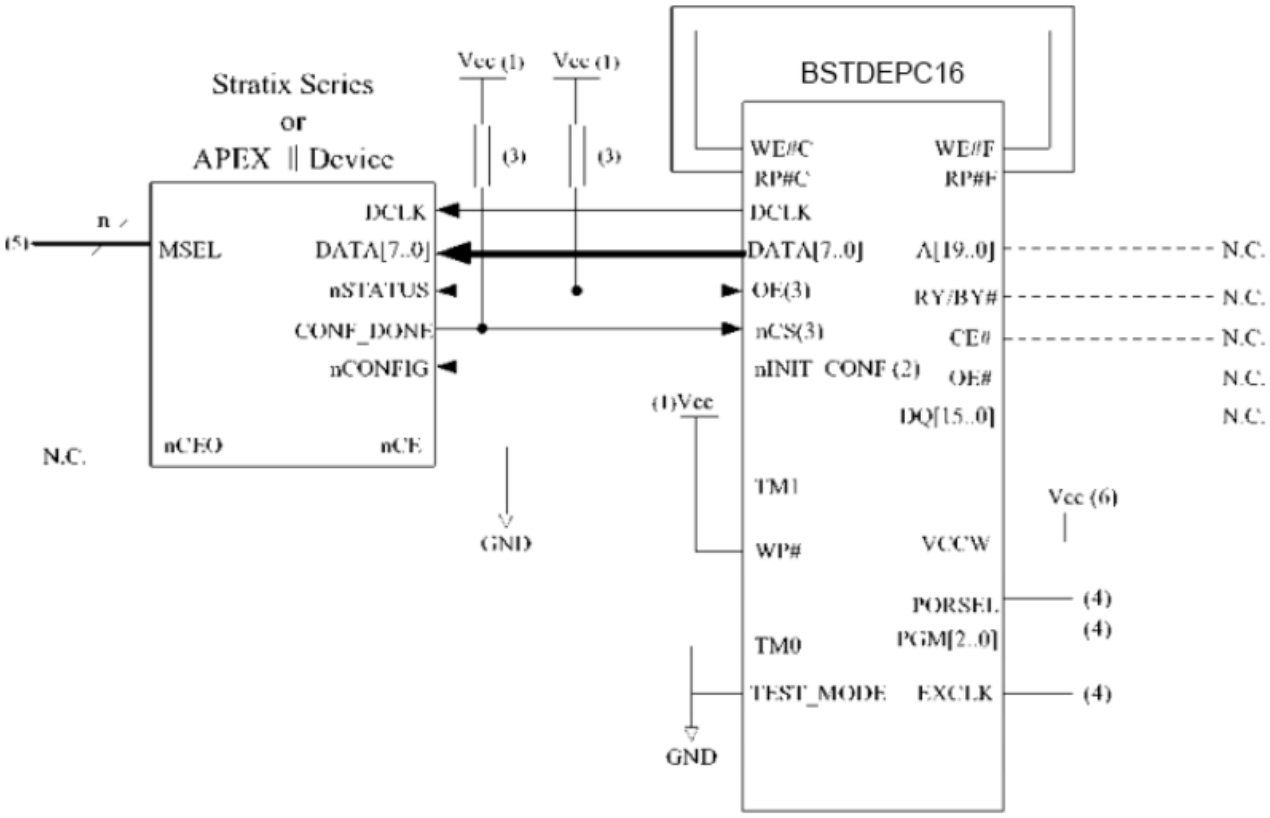
After configuration is completed, the controller drives DCLK low, DATA[] high, and releases the internal FLASH interface. The FLASH address and control signals have weak pull-ups inside, and there is a data retention circuit on the data line.

The following introduces the different configuration modes of this configuration device: FPP, PS, and concurrent configuration.

9.1.1. Fast Parallel Slave Mode Configuration (FPP)

Stratix series and APEXII devices can be configured in FPP mode. In this mode, the configuration device transmits one byte of data to the FPGA via DATA[7:0] at each DCLK.
Circuit connection method

As shown in Figure 9.1, in this figure, the external FLASH port is not used and most of the FLASH ports are not connected.



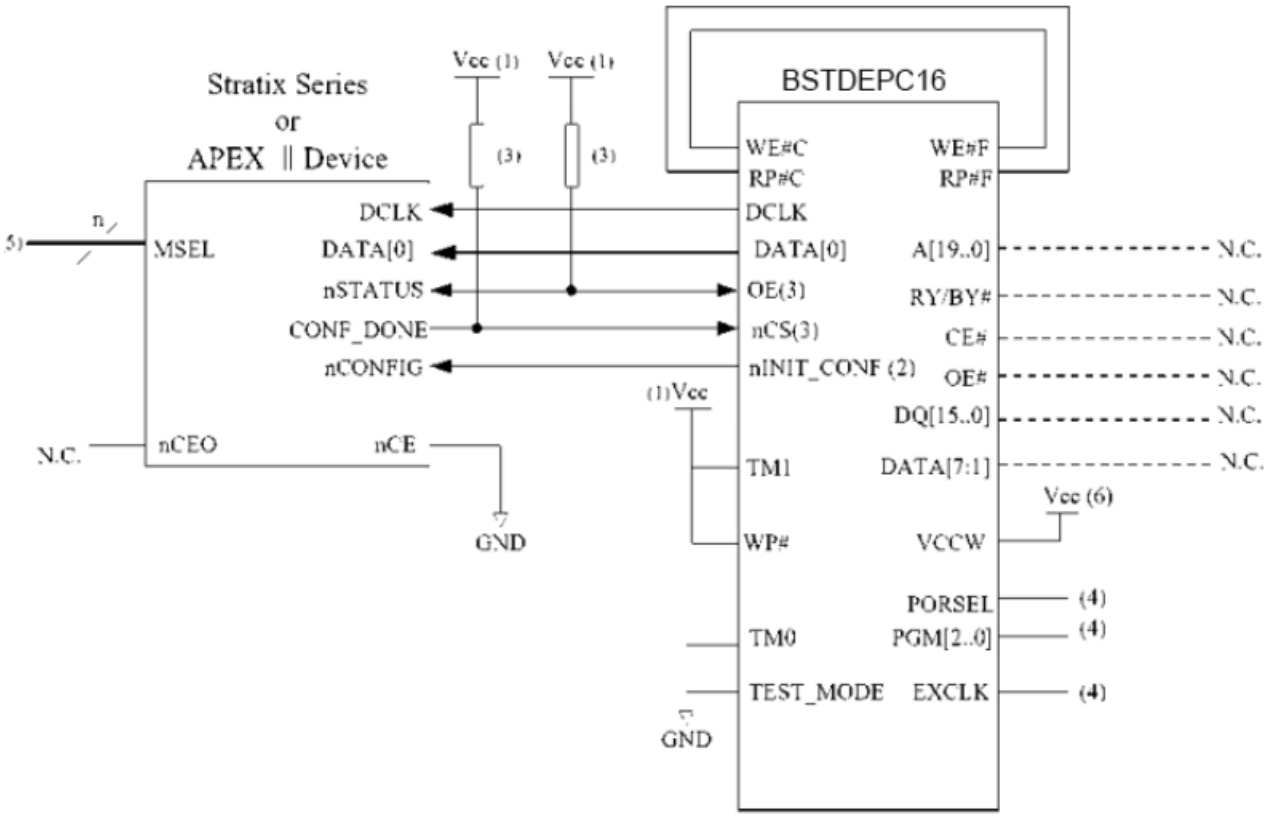
Note:

- Vcc is connected to the power supply of the configuration device.
- nINIT_CONF has an internal pull-up resistor that is always valid. If its function is not used, nINIT_CONF does not need to be connected to nCONFIG, but nCONFIG must be connected to Vcc directly or through a pull-up resistor.
- OE and nCS have internal programmable resistors. This function needs to be set in the Quartus II software. By default, there is a pull-up resistor. If the internal pull-up resistor is canceled, it will be invalid. You need to add a 10KΩ pull-up resistor externally.
- For the connection of PORSEL, PGM[2:0], and EXCLK, please refer to Examination form.
- Connect the MSEL[] pins of the FPGA to select the FPP configuration mode. For more configuration connection methods, refer to the corresponding FPGA's configuration_handbook.

- (6) In order to protect the contents of FLASH, the VCCW port and the VCC port can be powered separately. When VCCW is connected to a low level, the contents of FLASH cannot be rewritten.

9.1.2 Serial Slave Mode Configuration (PS)

APEX 20KC, APEX 20KE, APEX 20K, APEX II, Cyclone series, FLEX 10K, and Stratix series devices can be configured in PS mode, transferring 1 bit of uncompressed data to the FPGA via DATA[0] at every DCLK. DATA[7:1] is not used in this mode and is driven low.As shown in the picture Show.



Note:

- Vcc is connected to the power supply of the configuration device.
- nINIT_CONF has an internal pull-up resistor that is always valid. If its function is not used, nINIT_CONF does not need to be connected to nCONFIG, but nCONFIG must be connected to Vcc directly or through a pull-up resistor.

- OE and nCS have internal programmable resistors. This function needs to be set in the Quartus II software. By default, there is a pull-up resistor. If the internal pull-up resistor is canceled, it will be invalid. You need to add a 10KΩ pull-up resistor externally.
- For the connection of PORSEL, PGM[2:0], and EXCLK, please refer to Examination form.
- Connect the MSEL[] pins of the FPGA to select the PS configuration mode. For more configuration connection methods, refer to the corresponding FPGA's configuration_handbook.
- In order to protect the contents of FLASH, the VCCW port and the VCC port can be powered separately.

When connected to a low level, the contents of FLASH cannot be rewritten.

9.1.3. Simultaneous configuration

Enhanced configuration devices support simultaneous configuration of multiple FPGAs in slave mode. Simultaneous configuration means that the enhanced configuration device outputs n configuration data at the same time in DATA[n-1..0 (n=1,2,4,8), and each DATA[] line serially configures a different FPGA. Unused DATA[] is driven low. For the connection method of configuring multiple FPGAs or n = 1,2,4,8, etc., please refer to the configuration_handbook of the corresponding FPGA. Table 9.1 summarizes the simultaneous configuration mode. The connection relationship of configuring multiple FPGAs simultaneously in PS mode is shown in Figure 9.3

As shown:

Figure 9.3 Simultaneous

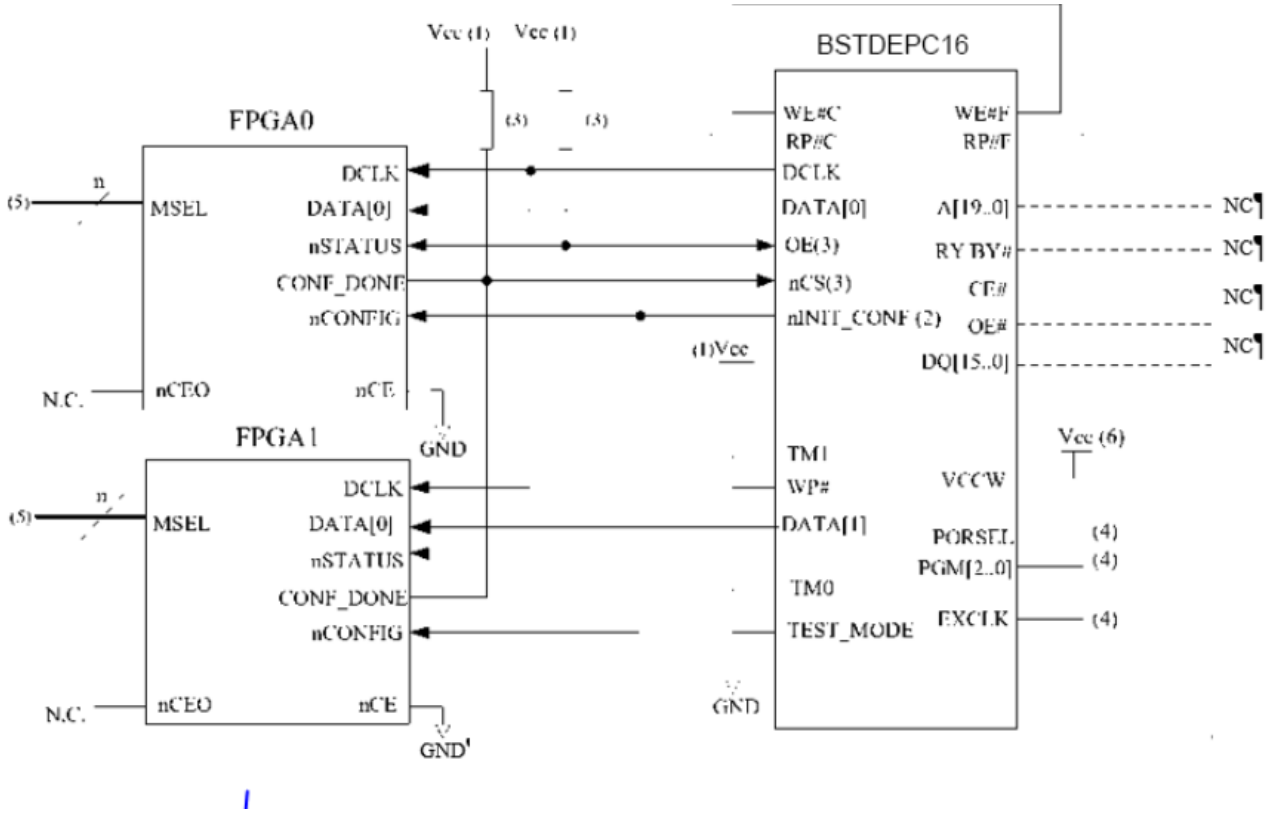


Figure 9.3. Simultaneous configuration of connection mode

Note:

- Vcc is connected to the power supply of the configuration device.
- nINIT_CONF has an internal pull-up resistor that is always valid.
- OE and nCS have internal programmable resistors. This function needs to be set in the Quartus II software. By default, there is a pull-up resistor. If the internal pull-up resistor is canceled, it will be invalid. You need to add a 10KΩ pull-up resistor externally.
- For the connection of PORSEL, PGM[2:0], and EXCLK, please refer to Examination form.
- Connect the MSEL[] pins of the FPGA to select the PS configuration mode. For more configuration connection methods, refer to the corresponding FPGA's configuration_handbook.
- In order to protect the contents of FLASH, the VCCW port and the VCC port can be powered separately. When VCCW is connected to a low level, the contents of FLASH cannot be rewritten.

Table 9.1. Summary of simultaneous configuration modes

Mode Name	Mode (n=) (1)	Available outputs	Unused Output
Fast Serial Mode	1	DATA[0]	DATA[7..1] is low level
Multi-device fast serial mode	2	DATA[1..0]	DATA[7..2] is low level
Multi-device fast serial mode	4	DATA[3..0]	DATA[7..4] is low level
Multi-device fast serial mode	8	DATA[7..0]	---

Note:

- Represents the number of valid output bits of the DATA pin of the configuration device

9.2. Dynamic configuration (page mode)

The dynamic configuration feature allows the configuration device in a system to store up to eight pages of configuration information for configuring the FPGA. When OE goes high to start configuration, the corresponding page is selected for configuration by sampling PGM[2:0]. The number of pages of the configuration device is affected by the size of the data on each page. To generate a programming file for page mode, you need to convert the programming file in the QuartusII software (convert the SOF file to a POF file). Page mode allows you to dynamically reconfigure the FPGA by selecting PGM[2:0] and setting the nCONFIG signal. PGM[2:0] cannot be left floating on the board. If the dynamic configuration function is not used, PGM[2:0] is tied low and the default page 000 is selected. For specific implementation, please refer to the configuration manual of the FPGA with this function.

9.3. Real-time decompression

BSTEPC16 can decompress compressed data on-chip, reducing the time to read configuration data from FLASH, thereby increasing the configuration rate. In Quartus II, the configuration data can be compressed and programmed into BSTEPC16. The compression option is selected in Quartus II. During configuration, the configuration device

will first decompress the compressed data, and then use the decompressed data to configure the FPGA according to the configuration scheme.

9.4. Programmable configuration clock

The user can program the configuration clock DCLK. The clock source is divided into external clock (EXCLK pin input) and internal oscillator clock. The external clock can reach up to 100MHz, and the internal clock has four options: 10, 33, 50, and 66MHz. The default is the internal 10MHz clock. There is a frequency division circuit inside that can divide the configuration clock as follows: 1~16 times integer division, 1.5, 2.5 fractional division. All of the above configuration clock settings are set in Quartus II. The schematic diagram is shown in Figure 9.4.

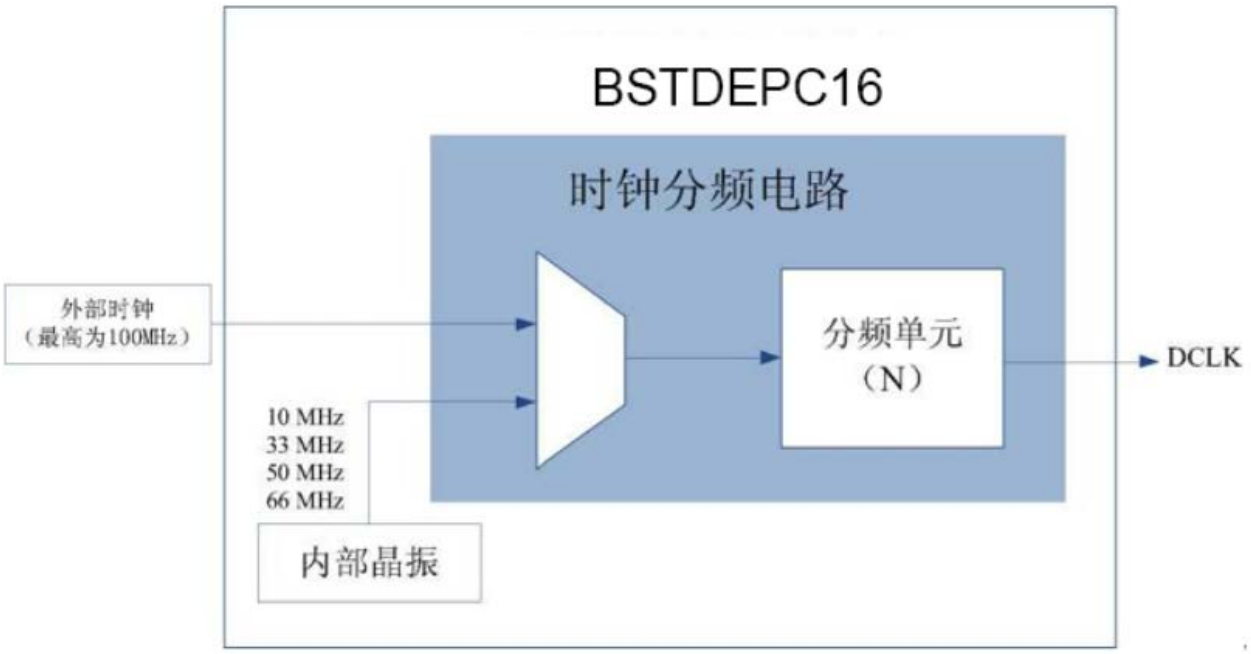


Figure 9.4. Configuration clock processing schematic

The internal oscillator frequency is 10MHz, 33MHz, 50MHz, 66Mhz. Its minimum, maximum and Typical values are shown in Table 9.2.

Table 9.2. Internal oscillator frequency

Frequency Setting (MHz)	Minimum value (MHz)	Typical value (MHz)	Maximum value (MHz)
10	6.4	8	10
33	twenty one	26.5	33
50	32	40	50
66	42	53	66

9.5. FLASH online programming

BSTEPC16 can be programmed using JTAG ISP and external FLASH interface. JTAG programming is online programming of FLASH through the internal controller. External FLASH interface programming requires an external processor or FPGA to program FLASH online.

9.5.1. JTAG Programming

IEEE Std.1149.1 and IEEE Std.1532 are used to implement JTAG programming.

Table 9.3. JTAG instructions

JTAG Instructions	Coding	Functional Description
SAMPLE/ PRELOAD	00 0101 0101	Each pin of the configuration device is sampled, the device is tested during normal operation, and outputs are enabled through the device pins.
BYPASS	11 1111 1111	Placing the bypass register between TDI and TDO allows synchronous transfer of BST data from a selected device to adjacent devices.
IDCODE	00 0101 1001	Select the IDCODE register and place it between TDI and TDO, and allow the IDCODE to be serially shifted out through TDO. The IDCODE for all enhanced configuration devices is: 0100A0DDh
INIT_CONF	00 0110 0001	This function initiates FPGA reconfiguration by pulling down nINIT_CONF connected to FPGA pin nCONFIG. When this instruction is updated, the JTAG state machine enters the Run-Test/Idle state, and nINIT_CONF is pulled low. When the JTAG state machine exits Run-Test/Idle, nINIT_CONF is released. nCONFIG will be pulled high by the pull-up resistor, and then the FPGA will be reconfigured.
USERCODE	00 0111 1001	Select the USERCODE register and place it between TDI and TDO, and allow USERCODE to be serially shifted out through TDO. USERCODE is user-defined.
PENDCFG	00 0110 0101	This optional feature is used to pull nINIT_CONF low when JTAG ISP

		is programming the configuration device. It is very useful when the external FLASH interface is controlled by an FPGA/processor. This feature can avoid contention between the controller and the external device trying to control the FLASH pins.

Note:

For other JTAG instructions, refer to the BSDL file.

9.5.2. External FLASH interface

BSTEPC16 supports external FPGA or processor access to FLASH memory. Unused FLASH parts can be used by external devices to store data. Parallel (16bit) programming of FLASH can be realized, and programming, erasing and verification operations can be realized at the same time. Data can be updated through the external FLASH interface, and the system uses the new configuration information to reconfigure the FPGA and realize remote configuration.

The address, data and control ports of FLASH are connected to the internal controller and external ports of the configuration device. FLASH can be accessed through external resources when FLASH is allowed to be accessed. Since the external FLASH interface and the internal controller share the FLASH interface, there is no bus arbitration mechanism between the two. Therefore, FLASH can only be accessed through the external FLASH port when the controller tri-states (does not control) the internal FLASH interface. If the controller and the external device access FLASH at the same time, a conflict will occur, resulting in configuration or programming failure.

The controller that configures the device accesses the FLASH during these processes:

- FPGA configuration process - read configuration data from FLASH
- Programming FLASH via JTAG - storing configuration data in FLASH
- Power-on reset - read OPTION BITS (attribute bits) from FLASH

In the above processes, the external FLASH port must be tri-stated and not controlled by external devices. If you want to access FLASH during configuration, you can input a low level to the nCONFIG pin of the FPGA. You can keep the nSTATUS-OE line low to keep the configuration device in reset state, thereby allowing access to FLASH through the external interface.

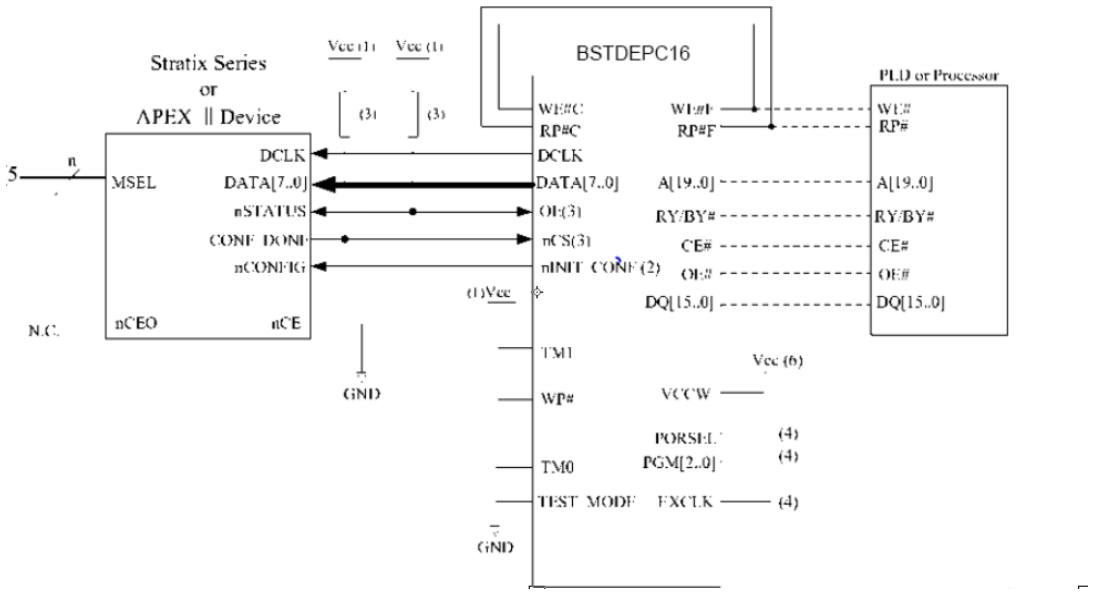


Figure 9.5. Connection method using external FLASH interface circuit

Note:

- Vcc is connected to the power supply of the configuration device.
- nINIT_CONF has an internal pull-up resistor that is always valid. If its function is not used, nINIT_CONF does not need to be connected to nCONFIG, but nCONFIG must be connected to Vcc directly or through a pull-up resistor.
- OE and nCS have internal programmable resistors. This function needs to be set in the Quartus II software. By default, there is a pull-up resistor. If the internal pull-up resistor is canceled, it will be invalid. You need to add a 10KΩ pull-up resistor externally.
- For the connection of PORSEL, PGM[2:0], and EXCLK, please refer to Examination form.
- Connect the MSEL[] pins of the FPGA to select the appropriate configuration mode. For more configuration connection methods, refer to the corresponding FPGA's configuration_handbook.
- VCCW is the erase and programming voltage of FLASH, so it must be connected to a power supply, otherwise errors will occur during the erase and programming operations.

For the timing and related parameters of erasing, programming and reading operations of FLASH through the BSTEPC16 external interface, please refer to the description of FLASH functions and parameters.

XX. Packaging Specifications

The device adopts ceramic CFBGA88 package. The shell shape and other related dimensions are shown in the figure below.

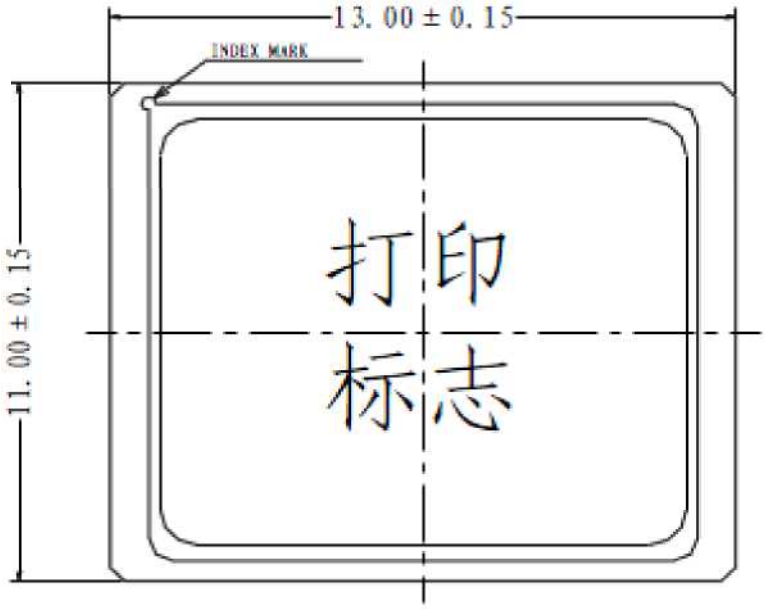


Figure 10.1. Appearance and dimensions

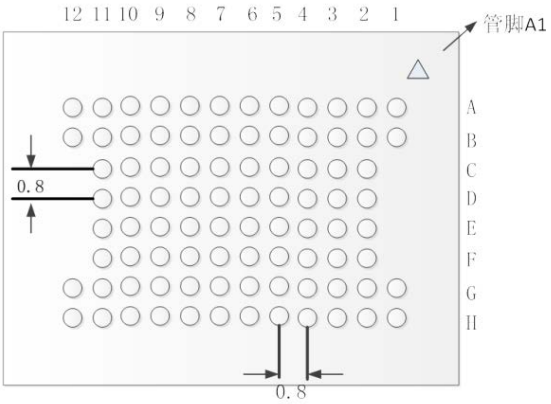


Figure 10.2. Bottom view and ball spacing

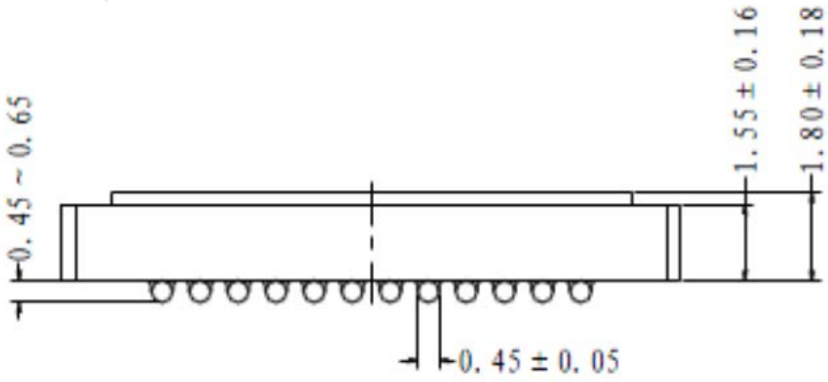


Figure 10.3. Tube thickness and ball diameter

XI. Odering Information

- Storage capacity: 16M Bit
- Package type: Ceramic Fine Line BGA package
- Number of pins: 88 pins
- Temperature range: -55~125℃