Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Platform Flash XL High-Density Configuration and Storage Device

DS617 (v4.0) August 5, 2015 **Product Specification**

Features

- In-System Programmable Flash Memory Optimized for Virtex®-5 or Virtex-6 FPGA Configuration
- High-Performance FPGA Bitstream Transfer up to 800 Mb/s (50 MHz × 16-bits), Ideal for PCI Express® Endpoint Applications
- MultiBoot Bitstream, Design Revision Storage
- FPGA Configuration Synchronization (READY_WAIT) Handshake Signal
- ISE® Software Support for In-System Programming via Xilinx® JTAG Cables
- Standard NOR-Flash Interface for Access to Code or Data Storage
- Operation over Full Industrial Temperature Range (–40°C to +85°C)
- Common Flash Interface (CFI)
- Low-Power Advanced CMOS NOR-Flash Process
- Endurance of 10,000 Program/Erase Cycles Per Block
- Power Supplies
  - Industry-Standard Core Power Supply Voltage (V_{DD}) = 1.8V
  - 3.3V or 2.5V I/O (V_{DDQ}) Power Supply Voltage
- Memory Organization
  - 128-Mb Main Array Capacity
  - 16-bit Data Bus
  - Multiple 8-Mb Bank Architecture for Dual Erase/Program and Read Operation
  - 127 Regular 1-Mb Main Blocks
  - 4 Small 256-Kb Parameter Blocks
- Synchronous/Asynchronous Read Modes
  - Power-On in Synchronous Burst Read Mode
  - Asynchronous Random Access Mode
  - Accelerated Asynchronous Page Read Mode
- Protection
  - Default Block Protection at Power-Up
  - Hardware Write Protection (when $V_{PP} = V_{SS}$)
- Security
  - Unique Device Number (64-bits)
  - One-Time-Programmable (OTP) Registers
- Small-Footprint (10 mm × 13 mm) FT64 Packaging

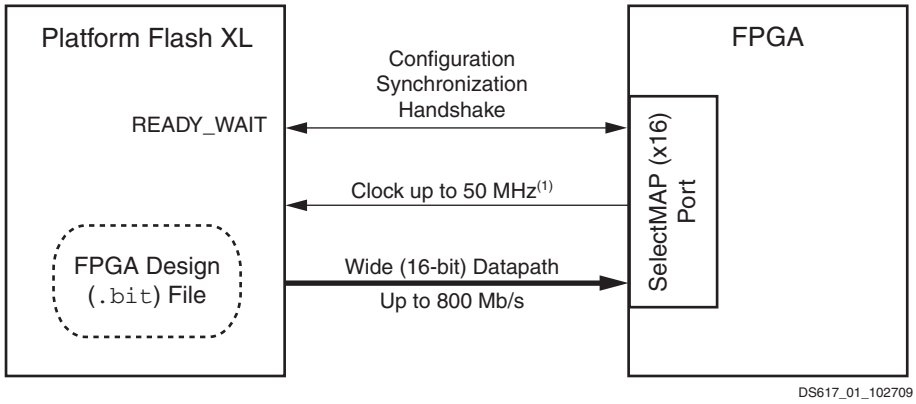
Description

A reliable compact high-performance configuration bitstream storage and delivery solution is essential for the high-density FPGAs. Platform Flash XL is the industry's highest performing configuration and storage device and is specially optimized for high-performance FPGA configuration. Platform Flash XL integrates 128 Mb of in-system programmable flash storage and performance features for configuration within a small-footprint FT64 package (Figure 5). Power-on burst read mode and dedicated I/O power supply enable Platform Flash XL to mate seamlessly with the native SelectMAP configuration interface. A wide, 16-bit data bus delivers the FPGA configuration bitstream at speeds up to 800 Mb/s without wait states. See UG438, *Platform Flash XL Configuration and Storage Device User Guide*, for system-level usage and performance considerations.

Platform Flash XL is supported for use with Virtex-5 or Virtex-6 FPGAs only. Use with older Virtex families, Spartan® families, or AES encrypted bitstreams is not supported.

Platform Flash XL is a non-volatile flash storage solution, optimized for FPGA configuration. The device provides a READY_WAIT signal that synchronizes the start of the FPGA configuration process, improving both system reliability and simplifying board design. Platform Flash XL can download an XC5VLX330 bitstream (79,704,832 bits) in less than 100 ms, making the configuration performance of Platform Flash XL ideal for PCI Express endpoints and other high-performance applications.

Platform Flash XL is a single-chip configuration solution with additional system-level capabilities. A standard NOR flash interface (Figure 2) and support for common flash interface (CFI) queries provide industry-standard access to the device memory space. The Platform Flash XL's 128 Mb capacity can typically hold one or more FPGA bitstreams. Any memory space not used for bitstream storage can be used to hold general purpose data or embedded processor code.



Notes:

1. System considerations can lower the configuration clock frequency below the maximum clock frequency for the device. To determine the maximum configuration clock frequency, check the minimum clock period (T_{KHKH}) for the chosen I/O voltage range (V_{DDQ}), the clock High-to-output valid time (T_{KHQV}), and the FPGA SelectMAP setup time.

Figure 1: **Platform Flash XL Delivers Reliable, High-Performance FPGA Configuration**

Platform Flash XL support is integrated with the Xilinx design and debug tool suite.The iMPACT application, included with the ISE software, supports indirect, in-system programming of Platform Flash XL via the IEEE Standard 1149.1 (JTAG) port on the FPGA for prototype programming (Figure 3).

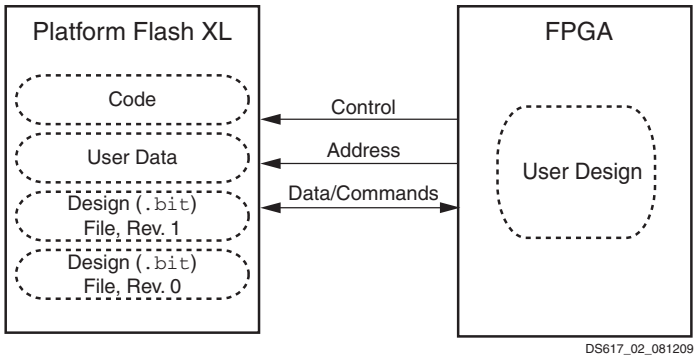


Figure 2: **Standard NOR Flash Interface for User Access to Memory**

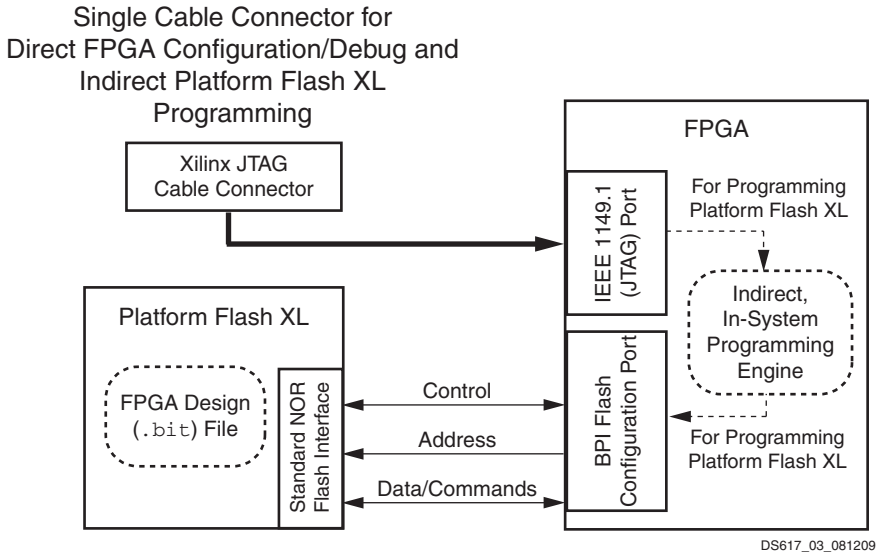


Figure 3: **Indirect Programming Solution for Platform Flash XL**

Flash Memory Architecture Overview

Platform Flash XL is a 128-Mb (8 Mb × 16) non-volatile flash memory. The device is in-system programmable with a 1.8V core (V_{DD}) power supply. A separate I/O (V_{DDQ}) power supply enables I/O operation at 3.3V or 2.5V. An optional 9V V_{PP} power supply can accelerate factory programming.

A common flash interface (CFI) provides access to device memory (Figure 3, page 3). Moreover, Platform Flash XL supports multiple read modes. A 23-bit address bus provides random read access to each 16-bit word. Four words occupy each page for accelerated page mode reads. The device powers-up in a synchronous burst read mode capable of sequential read rates up to 54 MHz.

Platform Flash XL has a multiple-bank architecture. An array of 131 individually erasable blocks are divided into 16, 8-Mb banks. Fifteen main banks contain uniform blocks of 64 Kwords, and one parameter bank contains seven main blocks of 64 Kwords, plus four parameter blocks of 16 Kwords.

Note: The device is electronically erasable at the block level and programmable on a word-by-word basis.

The multiple-bank architecture allows dual operations — read operations can occur on one bank while a program or erase operation occurs in a different bank. However, only one bank at a time is allowed to be in program or erase mode. Burst reads are allowed to cross bank boundaries.

Table 1 summarizes the bank architecture, and the memory map is shown in Figure 4, page 5. The parameter blocks are located at the top of the memory address space in Platform Flash XL.

Table 1: **Bank Architecture**

Number	Bank Size	Parameter Blocks	Main Blocks
Parameter Bank	8 Mbits	4 blocks of 16 Kwords	7 blocks of 64 Kwords
Bank 1	8 Mbits	–	8 blocks of 64 Kwords
Bank 2	8 Mbits	–	8 blocks of 64 Kwords
Bank 3	8 Mbits	–	8 blocks of 64 Kwords
⋮	⋮	⋮	⋮
Bank 14	8 Mbits	–	8 blocks of 64 Kwords
Bank 15	8 Mbits	–	8 blocks of 64 Kwords

Each block can be erased separately. Erase operations can be suspended in order to perform a program or read operation in any other block and then resumed. Program operations can be suspended to read data at any memory location except for the one being programmed, and then resumed.

Program and erase commands are written to the command interface of the memory. An internal program/erase controller takes care of the timing necessary for program and erase operations. The end of a program or erase operation can be detected and any error conditions identified in the status register. The command set required to control the memory is consistent with JEDEC standards.

The device supports synchronous burst read and asynchronous read from all blocks of the memory array. At power-up, the device is configured for synchronous read. In synchronous burst read mode, data is output on each clock cycle at frequencies of up to 54 MHz. The synchronous burst read operation can be suspended and resumed.

When the bus is inactive during asynchronous read operations, the device automatically switches to an automatic standby mode. In this condition the power consumption is reduced to the standby value, and the outputs are still driven.

Platform Flash XL features an instant, individual block-locking scheme, allowing any block to be locked or unlocked with no latency, and enabling instant code and data protection. All blocks have three levels of protection. Blocks can be locked and locked-down individually preventing any accidental programming or erasure. There is an additional hardware protection against program and erase: when $V_{PP} = V_{PPLK}$ all blocks are protected against program or erase. All blocks are locked at power-up.

The device features a separate region of 17 programmable registers whose values can be protected against further programming changes. Sixteen of these registers are each 128-bits in size, with the 17th register subdivided into two 64-bit registers. One of the 64-bit registers contains a factory preprogrammed, unique device number, permanently protected against modification. The second 64-bit register is user-programmable.

All bits within these registers (except for the permanently-protected unique number register) are one-time-programmable (OTP) — each bit can be programmed only once from a one-value to a zero-value.

Two protection lock registers can be programmed to lock any of the 17 protectable registers against further changes. One protection lock register contains bits that determine the protection state of the two special 64-bit registers. The bit corresponding to the unique device number register is pre-programmed to ensure the unique device number register is permanently protected against modification. The second protection lock register contains OTP bits that correspond the protection state each of the remaining 16 registers.

Platform Flash XL is available in a 10 × 13 mm, 1.0 mm-pitch FT64 package and supplied with all the bits erased (set to '1').

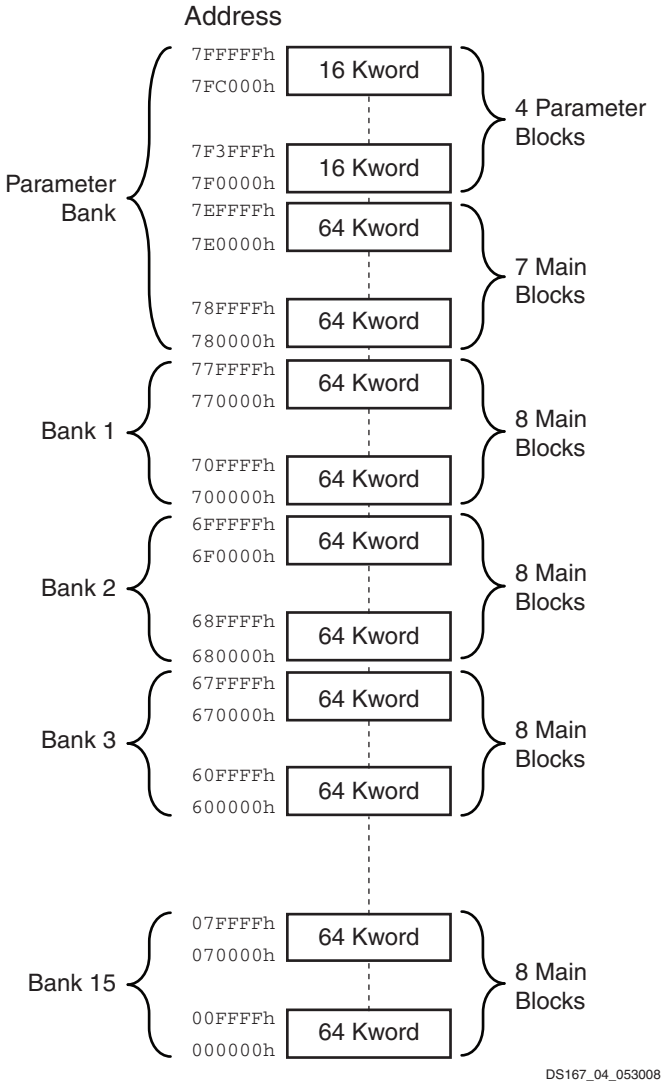


Figure 4: **Platform Flash XL Memory Map (Address Lines A22 – A0)**

Pinout and Signal Descriptions

See Figure 5 and Table 2 for a logic diagram and brief overview of the signals connected to this device.

Table 2: **Signal Names**

Signal Name	Function	Direction
A22-A0	Address Inputs	Inputs
DQ15-DQ0	Data Input/Outputs, Command Inputs	I/O
$\overline{E}$	Chip Enable	Input
$\overline{G}$	Output Enable	Input
$\overline{W}$	Write Enable	Input
$\overline{RP}$	Reset	Input
$\overline{WP}$	Write Protect	Input
K	Clock	Input
$\overline{L}$	Latch Enable	Input
READY_WAIT	Ready/Wait	I/O
V_{DD}	Supply Voltage	–
V_{DDQ}	Supply Voltage for Input/Output Buffers	–
V_{PP}	Optional(1) Supply Voltage for Fast Program and Erase	–
V_{SS}	Ground	–
V_{SSQ}	Ground Input/output Supply	–
NC	Not Connected Internally	–

Notes:

1. Typically, V_{PP} is tied to the V_{DDQ} supply on a board. See the V_{PP} Program Supply Voltage section for alternate options.

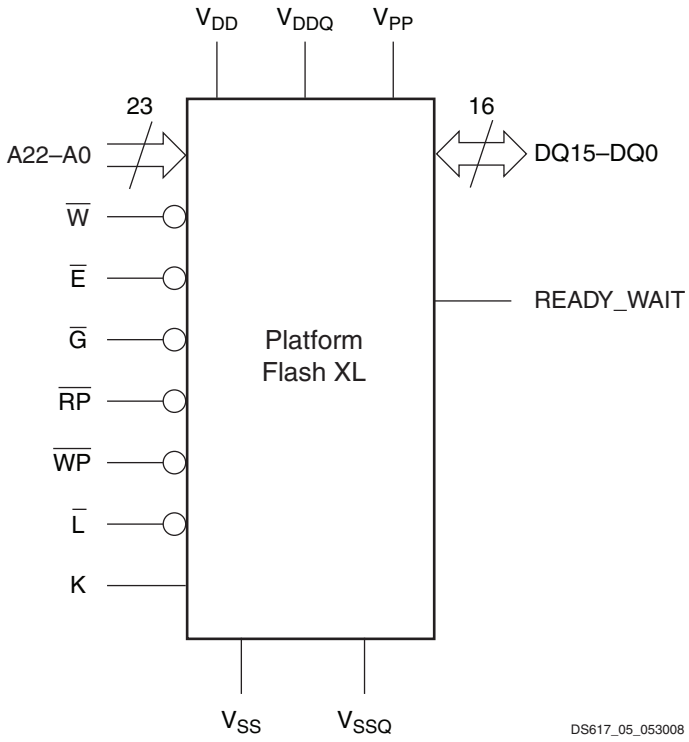


Figure 5: **Logic Diagram**

Address Inputs (A22-A0)

The Address inputs select the words in the memory array to access during Bus Read operations. During Bus Write operations they control the commands sent to the Command Interface of the Program/Erase Controller.

Data Inputs/Outputs (DQ15-DQ0)

The Data I/O output the data stored at the selected address during a Bus Read operation or input a command or the data to be programmed during a Bus Write operation.

Chip Enable ($\overline{E}$)

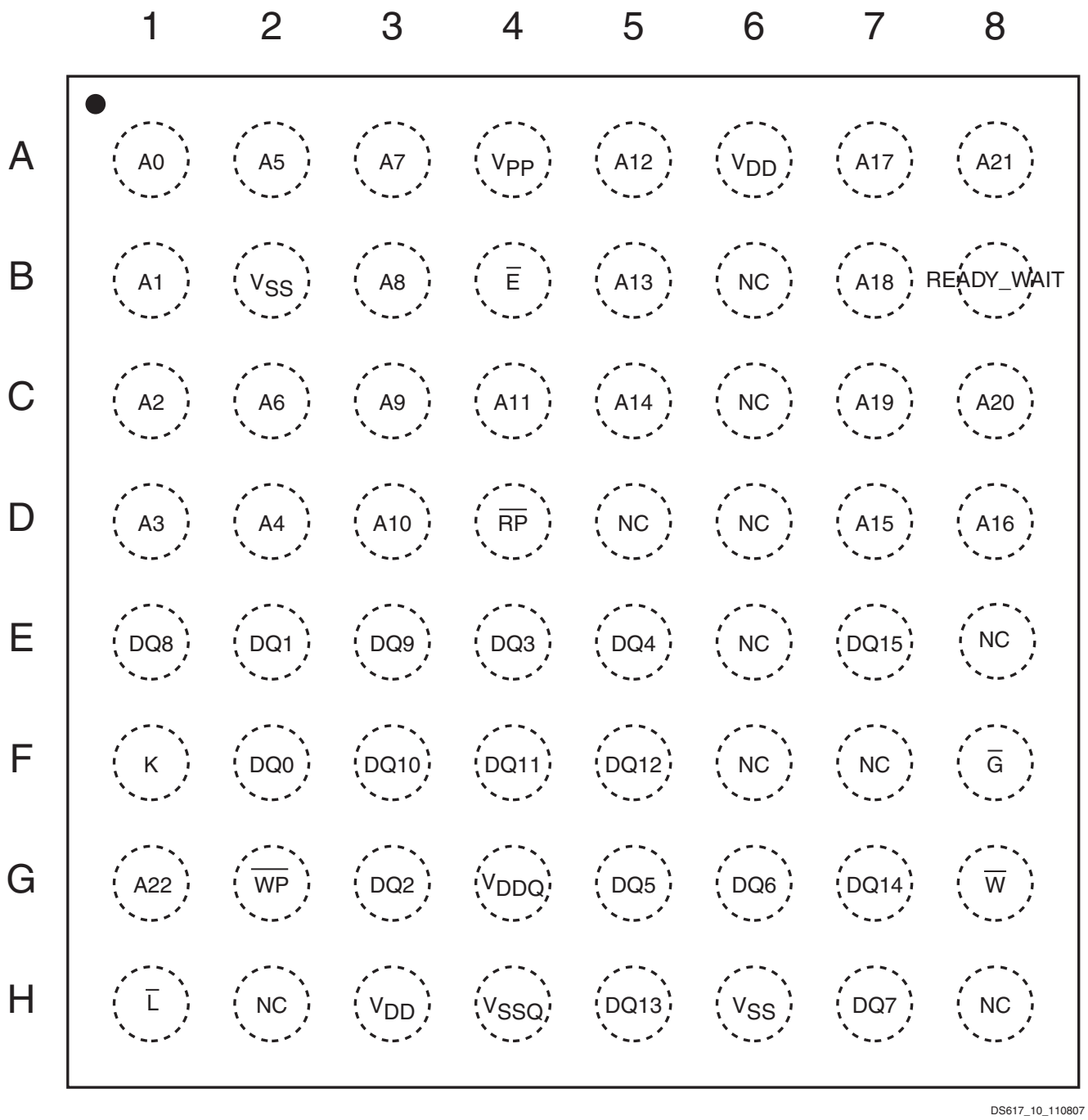
The Chip Enable input activates the memory control logic, input buffers, decoders and sense amplifiers. When Chip Enable is at V_{IL} and Reset is at V_{IH}, the device is in active mode. When Chip Enable is at V_{IH}, the memory is deselected, the outputs are high impedance, and the power consumption is reduced to the standby level.

Output Enable ($\overline{G}$)

The Output Enable input controls data outputs during the Bus Read operation of the memory. Before the start of the first address latching sequence (FALS), the Output Enable input must be held Low before the clock starts toggling.

Write Enable ($\overline{W}$)

The Write Enable input controls the Bus Write operation of the memory's Command Interface. The data and address inputs are latched on the rising edge of Chip Enable or Write Enable, whichever occurs first.



Notes:

1. See the FT64/FTG64 package specifications at http://www.xilinx.com/support/documentation/package_specifications.htm.

Figure 6: **FT64 Package Connections (Top View through Package)**

Write Protect ($\overline{WP}$)

Write Protect is an input that gives an additional hardware protection for each block. When Write Protect is at V_{IL}, the Lock-Down is enabled, and the protection status of the Locked-Down blocks cannot be changed. When Write Protect is at V_{IH}, the Lock-Down is disabled, and the Locked-Down blocks can be locked or unlocked.

Reset ($\overline{RP}$)

The Reset input provides a hardware reset of the memory. When Reset is at V_{IL}, the memory is in reset mode: the outputs are high impedance, and the current consumption is reduced to the Reset supply current I_{DD2}. After Reset all blocks are in the Locked state, and the Configuration Register is reset. When Reset is at V_{IH}, the device is in normal operation. Exiting reset mode the device enters the synchronous read mode and the FALS is executed.

Latch Enable ($\overline{L}$)

Latch Enable latches the address bits on its rising edge. The address latch is transparent when Latch Enable is at V_{IL} and inhibited when Latch Enable is at V_{IH}.

The Latch Enable ($\overline{L}$) signal must be held at V_{IH} during the power-up phase, during the FALS restart phase and through the entire FALS.

In asynchronous mode, the address is latched on $\overline{L}$ going High. or addresses are sent continuously if $\overline{L}$ is held Low. During Write operations, $\overline{L}$ can be tied Low (V_{IL}) to allow the addresses to flow through.

Table 3: **Latch Enable Logic Levels in Synchronous and Asynchronous Modes**

Operation	Asynchronous	Synchronous
Bus Read	X	V_{IH}
Bus Write	X or toggling	X or toggling
Address Latch	Toggling	Toggling
Standby	X	X
Reset	V_{IH}	V_{IH}
FALS	V_{IH}	V_{IH}
Power-up	V_{IH}	V_{IH}

Notes:

1. See waveforms in the "DC and AC Parameters" section for details.

Clock (K)

The Clock input synchronizes the memory to the FPGA during synchronous read operations. The address is latched on a Clock edge (rising or falling, according to the configuration settings) when Latch Enable is at V_{IL}. Clock is ignored during asynchronous read and in write operations.

Ready/Wait (READY_WAIT)

> ***Caution!*** The READY_WAIT requires an external pull-up resistor to V_{DDQ}. The external pull-up resistor must be sufficiently strong to ensure a clean, Low-to-High transition within less than one microsecond (T_{RWRT}) when the READY_WAIT pin is released to a high-impedance state.

READY_WAIT can perform one of two functions. By default, READY_WAIT is an input/open-drain ready signal coordinating the initiation of the device's synchronous read operation with the start of an FPGA configuration sequence. Optionally, READY_WAIT can be dynamically configured as an output wait signal, indicating a wait condition during a synchronous read operation.

Upon a power-on reset (POR) or $\overline{RP}$-pin reset event, the device drives READY_WAIT to V_{IL} until the device is ready to initiate a synchronous read or receive a command. When the device reaches an internal ready state from a reset condition, READY_WAIT is released to a high-impedance state (an external pull-up resistor to V_{DDQ} is required to externally pull the READY_WAIT signal to a valid input High). The device waits until the READY_WAIT input becomes a valid input High before permitting a synchronous read or accepting a command. Connecting the READY_WAIT to the FPGA INIT_B pin in a wired-and circuit creates a handshake coordinating the initiation of the device synchronous read with the start of the FPGA configuration sequence.

When READY_WAIT is an input/open-drain ready signal, the system can drive READY_WAIT to V_{IL} to reinitiate a synchronous read operation. A valid address must be provided to the device for a reinitiated synchronous read operation.

Optionally, READY_WAIT can be configured as an output signaling a wait condition during a synchronous read operation. The wait condition indicates a clock cycle during which the output data is not valid. When configured as an output wait signal, READY_WAIT is high impedance when Chip Enable is at V_{IH} or Output Enable is at V_{IH}. Only when configured as a wait signal, READY_WAIT can be configured to be active during the wait cycle or one clock cycle in advance, and the READY_WAIT polarity can be configured.

V_{DD} Supply Voltage

V_{DD} provides the power supply to the internal core of the memory device and is the main power supply for all operations (Read, Program and Erase).

V_{DDQ} Supply Voltage

V_{DDQ} provides the power supply to the I/O pins and enables all outputs to be powered independently of V_{DD}.

V_{PP} Program Supply Voltage

V_{PP} is either a control input or a power supply pin, selected by the voltage range applied to the pin.

If V_{PP} is kept in a low voltage range (0V to V_{DDQ}), V_{PP} is seen as a control input. In this case a voltage lower than V_{PPLK} gives absolute protection against program or erase, while V_{PP} in the V_{PP1} range enables these functions. V_{PP} is only sampled at the beginning of a program or erase — a change in its value after the operation starts does not have any effect, and all program or erase operations continue.

If V_{PP} is in the range of V_{PPH}, the signal acts as a power supply pin. In this condition V_{PP} must be stable until the Program/Erase algorithm is completed.

V_{SS} Ground

V_{SS} Ground is the reference for the core supply and must be connected to the system ground.

V_{SSQ} Ground

V_{SSQ} Ground is the reference for the input/output circuitry driven by V_{DDQ}. V_{SSQ} must be connected to V_{SS}.

Note: Each device in a system should have V_{DD}, V_{DDQ} and V_{PP} decoupled with a 0.1 µF ceramic capacitor close to the pin (high-frequency, inherently low-inductance capacitors should be placed

as close as possible to the package). The PCB track widths should be sufficient to carry the required V_{PP} program and erase currents.

FPGA Configuration Overview

Platform Flash XL enables the rich set of FPGA configuration features without additional glue logic. The device delivers the FPGA bitstream at power-on through a 16-bit data bus at data rates up to 800 Mb/s. The FPGA can also be configured from one of many design/revision bitstreams stored in the device. These revision bitstreams are accessed through the FPGA's MultiBoot addressing and fallback features available in specific system configurations with Platform Flash XL. For detailed descriptions of the FPGA configuration features and configuration procedure, refer to the respective FPGA configuration user guide.

At a high level, the general procedure for FPGA configuration from Platform Flash XL is as follows:

1. A system event, such as power-up, initiates the FPGA configuration process. The FPGA drives its INIT_B pin Low while it clears its configuration memory. The Platform Flash XL drives its READY_WAIT pin Low during its reset period.
2. When ready, the FPGA and Platform Flash XL release their respective INIT_B and READY_WAIT pins. An external resistor pulls the connected INIT_B-READY_WAIT signal from Low to High, synchronizing the start of the FPGA configuration process.
3. At the start of the configuration process, the FPGA samples its mode pins to determine its configuration mode. For Master BPI-Up mode, the FPGA outputs an address to read from the flash. For Slave SelectMAP mode, onboard resistors set the initial flash read address.
4. The Platform Flash XL latches the initial address from the FPGA or from onboard resistor settings into its internal address counter and the Platform Flash XL outputs the first 16-bit word.
5. The bitstream is synchronously transferred from the Platform Flash XL to the FPGA. During each successive FPGA CCLK period, the Platform Flash XL increments its internal address counter and outputs the next 16-bit word of the bitstream for the FPGA to consume.
6. At the end of the configuration process, the FPGA starts operation of the loaded bitstream and either drives DONE High or releases DONE to High, indicating the completion of the configuration procedure.

Platform Flash XL can configure the FPGA in Slave SelectMAP (x16) (recommended for maximum performance), Master SelectMAP (x16), or Master BPI-Up (x16) configuration mode. See Table 4 for a summary of attributes for different configuration modes and memories.

Table 4: **Overview of FPGA Configuration from Platform Flash XL and Standard BPI Flash**

	Platform Flash XL		Third-Party Standard BPI Flash (110-ns Access Time)
	High-Performance Configuration Mode	**Standard BPI Flash Compatibility Mode**	
FPGA Configuration Mode	Slave SelectMAP mode (x16 data bus width)	Master BPI-Up mode (x16 data bus width)	Master BPI-Up mode (x16 data bus width)
Guaranteed Bitstream Transfer Bandwidth at Best Clock Setting	800 Mb/s(1)	248 Mb/s(2)	78 Mb/s(3)
Virtex-5 FPGA Support	✓	✓	✓
Virtex-6 FPGA Support	✓	✓	✓
ISE Software Programming Support	✓	✓	For limited setups(4)
MultiBoot Capable	✓	✓	✓

Notes:

1. The 800 Mb/s rate is achieved using a Virtex-5 FPGA with an external 50 MHz configuration clock source. Specific speed grades of the Virtex-6 FPGA or system-level considerations can limit the configuration performance to less than 800 Mb/s.
2. Bandwidth is based on an example Virtex-5 FPGA considering $F_{MCCKTOL}$ and BitGen ConfigRate = 31 MHz (nominal frequency).
3. Bandwidth is based on an example Virtex-5 FPGA considering $F_{MCCKTOL}$ and BitGen ConfigRate = 17 MHz (nominal frequency), bpi_page_size = 4, and bpi_1st_read_cycle = 4. First word access time = 110 ns; Page word access time = 25 ns.
4. See XAPP973, *Indirect Programming of BPI PROMs with Virtex-5 FPGAs.*

Slave SelectMAP Configuration Mode

Platform Flash XL achieves maximum configuration performance when the FPGA is in Slave SelectMAP configuration mode. In the Slave SelectMAP mode, a stable, external clock source can drive the synchronous bitstream transfer from the device to the FPGA up to the maximum burst read frequency (T_{CLK}). See the SelectMAP Configuration Interface section in the respective FPGA Configuration User Guide for details of the Slave SelectMAP mode.

Note: The FPGA fallback feature is disabled in the Slave SelectMAP mode.

See UG438, *Platform Flash XL Configuration and Storage Device User Guide*, for guidance and examples of FPGAs connected to a Platform Flash XL for Slave SelectMAP configuration mode.

Alternate Configuration Modes

Platform Flash XL is optimized for the Slave SelectMAP configuration mode. Alternatively, Platform Flash XL can configure an FPGA via the Master SelectMAP or Master BPI-Up mode—albeit with compromises in configuration speed. See the respective FPGA configuration user guide for details regarding the Master SelectMAP mode or Master BPI-Up mode.

See UG438, *Platform Flash XL Configuration and Storage Device User Guide*, for additional information on using the Platform Flash XL with the FPGA in Master SelectMAP or Master BPI-Up mode.

Programming Overview

Programming solutions satisfying the requirements for each product phase are available for Platform Flash XL. ISE software provides integrated programming support for the FPGA design engineer in the prototyping environment. Third-party programming support is also available for the demands of the manufacturing environments.

iMPACT Programming Solution for Prototype FPGA Designs

Xilinx ISE software has integral support for in-system programming enabling rapid develop-program-and-test cycles for prototype FPGA designs. The software can compile the FPGA design into a configuration bitstream and program the bitstream into a Platform Flash XL in-system via a Xilinx JTAG cable (Figure 7).

The iMPACT software tool within the ISE software suite formats the FPGA user design bitstream into a flash memory image file and programs the device via a Xilinx JTAG cable connection to the JTAG port of the FPGA. For the programming process, the iMPACT software first downloads a pre-built bitstream containing an in-system programming engine into the FPGA. Then, the iMPACT software indirectly programs the FPGA user design bitstream into a Platform Flash XL via the downloaded in-system programming engine in the FPGA.

Note: For iMPACT software indirect in-system programming support, a specific set of connections is required between the FPGA and Platform Flash XL. See UG438, *Platform Flash XL Configuration and Storage Device User Guide,* for recommended connections. iMPACT supports reading and writing of only the main memory array. iMPACT does not support reading or writing of special data registers, for example, electronic signature codes, protection registers, or OTP registers.

Production Programming Solutions

For the requirements of manufacturing environments, multiple solutions exist for programming Platform Flash XL. Programming support is available for the common production programming platforms.

Note: Check with the third-party vendor for the availability of Platform Flash XL programming support.

Device Programmers

Device programmers can gang program a high volume of Platform Flash XL in an minimum of time. Third-party device programmer vendors, such as BPM Microsystems, support programming of Platform Flash XL.

See http://www.xilinx.com/support/programr/dev_sup.htm for a sample list of third-party programmer vendors supporting Platform Flash XL.

Device programmers require the array data in the form of a standard PROM formatted data file, such as MCS. The FPGA `.bit` file is not a valid data input format for third-party device programmers. See the *Platform Flash XL Configuration and Storage Device User Guide* for instructions on preparing a programming file.

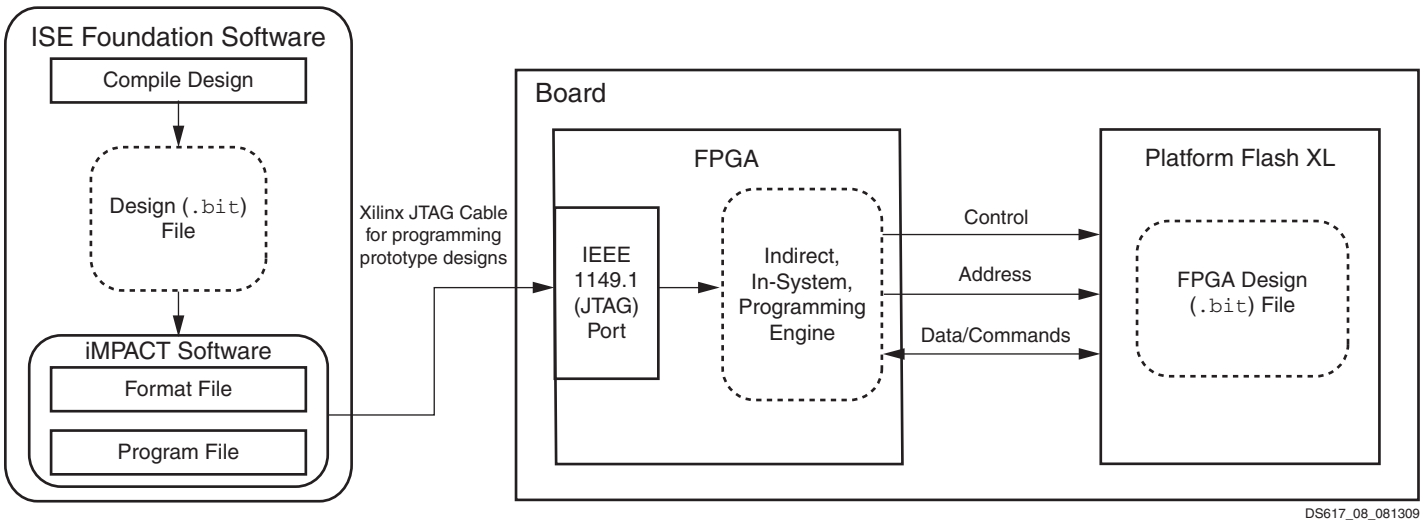


Figure 7: **Integrated FPGA Design and In-System Programming Solution for Platform Flash XL**

Bus Operations

There are six standard bus operations that control the device: Bus Read, Bus Write, Address Latch, Output Disable, Standby and Reset (Table 5).

Bus Read

Bus Read operations are used to output the contents of the Memory Array, Electronic Signature, Status Register and Common Flash Interface. Both Chip Enable and Output Enable must be at V_{IL} in order to perform a read operation. The Chip Enable input should be used to enable the device. Output Enable should be used to gate data onto the output. The data read depends on the previous command written to the memory (see "Command Interface," page 14).

Bus Write

Bus Write operations write commands to the memory or latch Input Data to be programmed. A Bus Write operation is initiated when Chip Enable and Write Enable are at V_{IL} with Output Enable at V_{IH}. Commands, Input Data and Addresses are latched on the rising edge of Write Enable or Chip Enable, whichever occurs first. The addresses can be latched prior to the write operation by toggling Latch Enable (when Chip Enable is at V_{IL}).

The Latch Enable signal can also be held at V_{IL} by the system, but then the system must guarantee that the address lines remain stable for at least T_{WHAX}.

Note: Typically glitches of less than 5 ns on Chip Enable or Write Enable are ignored by the memory and do not affect Bus Write operations.

Address Latch

Address latch operations input valid addresses. Both Chip enable and Latch Enable must be at V_{IL} during address latch operations. Addresses are latched on the rising edge of Latch Enable.

Output Disable

The outputs are held at high impedance when Output Enable is at V_{IH}.

Standby

Standby disables most of the internal circuitry allowing a substantial reduction of the current consumption. The memory is in standby when Chip Enable and Reset are at V_{IH}. Power consumption is reduced to the standby level I_{DD3}, and the outputs are set to high impedance independently from Output Enable or Write Enable. If Chip Enable switches to V_{IH} during a program or erase operation, the device enters Standby mode when finished with the program or erase operation.

Reset

During Reset mode, the memory is deselected and the outputs are high impedance. The memory is in Reset mode when Reset is at V_{IL}. Power consumption is reduced to the Reset level independently from Chip Enable, Output Enable or Write Enable. If Reset is pulled to V_{SS} during a Program or Erase, this operation is aborted and the memory content is no longer valid.

Table 5: **Bus Operations(1)**

Operation	$\overline{E}$	$\overline{G}$	$\overline{W}$	$\overline{L}$	$\overline{RP}$	READY_WAIT(2,3)		DQ15-DQ0
						CR4 = 1	CR4 = 0	
Bus Read	V_{IL}	V_{IL}	V_{IH}	V_{IL}(4)	V_{IH}	Hi-Z	–	Data output
Bus Write	V_{IL}	V_{IH}	V_{IL}	V_{IL}(4)	V_{IH}	Hi-Z	–	Data input
Address Latch	V_{IL}	X	V_{IH}	V_{IL}	V_{IH}	Hi-Z	–	Data output or Hi-Z(5)
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{IH}	Hi-Z	Hi-Z	Hi-Z
Standby	V_{IH}	X	X	X	V_{IH}	Hi-Z	Hi-Z	Hi-Z
Reset	X	X	X	X	V_{IL}(6)	V_{IL}(7)	–	Hi-Z
FALS	V_{IL}	V_{IL}	V_{IH}	V_{IH}	V_{IH}	Hi-Z	–	Data output

Notes:

1. X = Don't care.
2. If READY_WAIT is configured as an output wait signal (CR4 = 0), then the CR10 Configuration Register bit defines the signal polarity.
3. READY_WAIT is configured using the CR4 Configuration Register bit.
4. $\overline{L}$ can be tied to V_{IH} if the valid address was previously latched.
5. Depends on $\overline{G}$.
6. The Configuration Register reverts to its default value after a Low logic level (V_{IL}) is detected on the $\overline{RP}$ pin.
7. READY_WAIT pin used as an output. READY_WAIT goes Low T_{PLRWL} after $\overline{RP}$ goes Low.

Command Interface

All Bus Write operations to the memory are interpreted by the Command Interface. Commands consist of one or more sequential Bus Write operations. An internal Program/Erase Controller handles all timings and verifies the correct execution of the program and erase commands. The Program/Erase Controller provides a Status Register whose output can be read at any time to monitor the progress or the result of the operation.

The Command Interface is set to synchronous read mode when power is first applied, when exiting from Reset, or whenever V_{DD} falls below its power-down threshold. Command sequences must be followed exactly — any invalid combination of commands are ignored.

Table 6 provides a summary of the Command Interface codes.

Table 6: **Command Codes**

Hex Code	Command
01h	Block Lock Confirm
03h	Set Configuration Register Confirm
10h	Alternative Program Setup
20h	Block Erase Setup
2Fh	Block Lock-Down Confirm
40h	Program Setup
50h	Clear Status Register
60h	Block Lock Setup, Block Unlock Setup, Block Lock Down Setup and Set Configuration Register Setup
70h	Read Status Register
80h	Buffer Enhanced Factory Program Setup
90h	Read Electronic Signature
98h	Read CFI Query
B0h	Program/Erase Suspend
BCh	Blank Check Setup
C0h	Protection Register Program
CBh	Blank Check Confirm
D0h	Program/Erase Resume, Block Erase Confirm, Block Unlock Confirm, Buffer Program or Buffer Enhanced Factory Program Confirm
E8h	Buffer Program
FFh	Read Array

Read Array Command

The Read Array command returns the addressed bank to Read Array mode. One Bus Write cycle is required to issue the Read Array command. After a bank is in Read Array mode, subsequent read operations output data from the memory array.

A Read Array command can be issued to any bank while programming or erasing in another bank. If the Read Array command is issued to a bank currently executing a program or erase operation, the bank returns to Read Array mode but the program or erase operation continues; however the data output from the bank is not guaranteed until the program or erase operation finishes. The read modes of other banks are not affected.

Read Status Register Command

The device contains a Status Register used to monitor program or erase operations.

The Read Status Register command is used to read the contents of the Status Register for the addressed bank. One Bus Write cycle is required to issue the Read Status Register command. After a bank is in Read Status Register mode, subsequent read operations output the contents of the Status Register.

The Status Register data is latched on the falling edge of Chip Enable or Output Enable. Either Chip Enable or Output Enable must be toggled to update the Status Register data.

The Read Status Register command can be issued at any time, even during program or erase operations. The Read Status Register command only changes the read mode of the addressed bank. The read modes of other banks are not affected. Only Asynchronous Read and Single Synchronous Read operations should be used to read the Status Register.

A Read Array command is required to return the bank to Read Array mode.

See Table 11, page 23 for the description of the Status Register Bits.

Read Electronic Signature Command

The Read Electronic Signature command is used to read the Manufacturer and Device Codes, Lock Status of the addressed bank, Protection Register, and Configuration Register. One Bus Write cycle is required to issue the Read Electronic Signature command. After a bank is in Read Electronic Signature mode, subsequent read operations in the same bank output the Manufacturer Code, Device Code, Lock Status of the addressed bank, Protection Register, or Configuration Register (see Table 10, page 22).

The Read Electronic Signature command can be issued at any time, even during program or erase operations, except during Protection Register Program operations. Dual operations between the Parameter bank and the Electronic Signature location are not allowed (see Table 17, page 36 for details).

If a Read Electronic Signature command is issued to a bank executing a program or erase operation, the bank enters

into Read Electronic Signature mode. Subsequent Bus Read cycles output Electronic Signature data, and the Program/Erase controller continues to program or erase in the background.

The Read Electronic Signature command only changes the read mode of the addressed bank. The read modes of other banks are not affected. Only Asynchronous Read and Single Synchronous Read operations should be used to read the Electronic Signature. A Read Array command is required to return the bank to Read Array mode.

Read CFI Query Command

The Read CFI Query command is used to read data from the Common Flash Interface (CFI). One Bus Write cycle is required to issue the Read CFI Query command. After a bank is in Read CFI Query mode, subsequent Bus Read operations in the same bank read from the Common Flash Interface. The Read CFI Query command can be issued at any time, even during program or erase operations.

If a Read CFI Query command is issued to a bank executing a program or erase operation, the bank enters into Read CFI Query mode. Subsequent Bus Read cycles output CFI data, and the Program/Erase controller continues to program or erase in the background.

The Read CFI Query command only changes the read mode of the addressed bank. The read modes of other banks are not affected. Only Asynchronous Read and Single Synchronous Read operations should be used to read from the CFI. A Read Array command is required to return the bank to Read Array mode. Dual operations between the Parameter Bank and the CFI memory space are not allowed (see Table 17, page 36 for details).

See "Appendix B: Common Flash Interface," page 65, Table 36, page 65, through Table 45, page 70, Table 38, Table 38 for details on the information contained in the Common Flash Interface memory area.

Clear Status Register Command

The Clear Status Register command can be used to reset (set to '0') all error bits (SR1, 3, 4 and 5) in the Status Register. One Bus Write cycle is required to issue the Clear Status Register command. The Clear Status Register command does not affect the read mode of the bank.

The error bits in the Status Register do not automatically return to '0' when a new command is issued. The error bits in the Status Register should be cleared before attempting a new program or erase command.

Block Erase Command

The Block Erase command is used to erase a block. It sets all the bits within the selected block to '1'. All previous data in the block is lost.

If the block is protected, then the erase operation aborts, data in the block is not changed, and the Status Register outputs the error.

Two Bus Write cycles are required to issue the command.

- The first bus cycle sets up the Block Erase command.
- The second latches the block address and starts the Program/Erase Controller.

If the second bus cycle is not the Block Erase Confirm code, Status Register bits SR4 and SR5 are set and the command is aborted.

After the command is issued, the bank enters Read Status Register mode, and any read operation within the addressed bank outputs the contents of the Status Register. A Read Array command is required to return the bank to Read Array mode.

During Block Erase operations, the bank containing the block being erased only accepts the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query, and Program/Erase Suspend command; all other commands are ignored.

The Block Erase operation aborts if Reset ($\overline{RP}$) goes to V_{IL}. As data integrity cannot be guaranteed when the Block Erase operation is aborted, the block must be erased again.

Refer to "Dual Operations and Multiple Bank Architecture," page 35 for detailed information about simultaneous operations allowed in banks not being erased.

Typical Erase times are given in Table 21, page 44.

See Figure 41, page 75, for a suggested flowchart for using the Block Erase command.

Blank Check Command

The Blank Check command is used to check whether a Block is completely erased. Only one block at a time can be checked. To use the Blank Check command, V_{PP} must be equal to V_{PPH}. If V_{PP} is not equal to V_{PPH}, the device ignores the command and no error is shown in the Status Register.

Two bus cycles are required to issue the Blank Check command:

- The first bus cycle writes the Blank Check command (BCh) to any address in the block to be checked.
- The second bus cycle writes the Blank Check Confirm command (CBh) to any address in the block to be checked and starts the Blank Check operation.

If the second bus cycle is not Blank Check Confirm, Status Register bits SR4 and SR5 are set to '1', and the command aborts.

After the command is issued, the addressed bank automatically enters the Status Register mode and further reads within the bank output the Status Register contents.

The only operation permitted during Blank Check is Read Status Register. Dual Operations are not supported while a Blank Check operation is in progress. Blank Check operations cannot be suspended and are not allowed while the device is in Program/Erase Suspend.

The SR7 Status Register bit indicates the status of the Blank Check operation in progress:

- SR7 = '0' indicates that the Blank Check operation is still ongoing.
- SR7 = '1' indicates that the operation is complete.

The SR5 Status Register bit goes High (SR5 = '1') to indicate that the Blank Check operation has failed.

At the end of the operation the bank remains in the Read Status Register mode until another command is written to the Command Interface.

See Figure 38, page 72, for a suggested flowchart for using the Blank Check command.

Typical Blank Check times are given in Table 21, page 44.

Program Command

The program command is used to program a single word to the memory array. If the block being programmed is protected, then the Program operation aborts, data in the block is not changed, and the Status Register outputs the error.

Two Bus Write cycles are required to issue the Program Command.

- The first bus cycle sets up the Program command.
- The second latches the address and data to be programmed and starts the Program/Erase Controller.

After the programming starts, read operations in the bank being programmed output the Status Register content.

During a Program operation, the bank containing the word being programmed only accepts the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query and Program/Erase Suspend command; all other commands are ignored. A Read Array command is required to return the bank to Read Array mode.

Refer to "Dual Operations and Multiple Bank Architecture," page 35 for detailed information about simultaneous operations allowed in banks not being programmed.

Typical Program times are given in Table 21, page 44.

The Program operation aborts if Reset ($\overline{RP}$) goes to V_{IL}. As data integrity cannot be guaranteed when the Program operation is aborted, the word must be reprogrammed.

See Figure 37, page 71, for the flowchart for using the Program command.

Buffer Program Command

The Buffer Program Command makes use of the device's 32-word Write Buffer to speed up programming. Up to 32 words can be loaded into the Write Buffer. The Buffer Program command dramatically reduces in-system programming time compared to the standard non-buffered Program command.

Four successive steps are required to issue the Buffer Program command:

1. The first Bus Write cycle sets up the Buffer Program command. The setup code can be addressed to any location within the targeted block.

   After the first Bus Write cycle, read operations in the bank output the contents of the Status Register. Status Register bit SR7 should be read to check that the buffer is available (SR7 = 1). If the buffer is not available (SR7 = 0), the Buffer Program command must be re-issued to update the Status Register contents.
2. The second Bus Write cycle sets up the number of words to be programmed. Value n is written to the same block address, where n + 1 is the number of words to be programmed.
3. A total of n + 1 Bus Write cycles are used to load the address and data for each word into the Write Buffer. Addresses must lie within the range from the start address to the start address + n, where the start address is the location of the first data to be programmed. Optimum performance is obtained when the start address corresponds to a 32-word boundary.
4. The final Bus Write cycle confirms the Buffer Program command and starts the program operation.

All the addresses used in the Buffer Program operation must lie within the same block. Invalid address combinations or failing to follow the correct sequence of Bus Write cycles sets an error in the Status Register and aborts the operation without affecting the data in the memory array.

If the block being programmed is protected, an error is set in the Status Register, and the operation aborts without affecting the data in the memory array.

During Buffer Program operations, the bank being programmed only accepts the Read Array, Read Status Register, Read Electronic Signature, Read CFI Query and Program/Erase Suspend command; all other commands are ignored.

Refer to "Dual Operations and Multiple Bank Architecture," page 35 for detailed information about simultaneous operations allowed in banks not being programmed.

See Figure 39, page 73, for a suggested flowchart on using the Buffer Program command.

Buffer Enhanced Factory Program Command

The Buffer Enhanced Factory Program command has been specially developed to speed up programming in manufacturing environments where the programming time is critical. The command is used to program one or more Write Buffer(s) of 32 words to a block. After the device enters Buffer Enhanced Factory Program mode, the Write Buffer can be reloaded any number of times as long as the address remains within the same block. Only one block can be programmed at a time.

If the block being programmed is protected, then the Program operation aborts, data in the block is not changed, and the Status Register outputs the error.

The use of the Buffer Enhanced Factory Program command requires certain operating conditions:

- V_{PP} must be set to V_{PPH}.
- V_{DD} must be within operating range.
- Ambient temperature T_A must be 30°C ±10°C.
- The targeted block must be unlocked.
- The start address must be aligned with the start of a 32- word buffer boundary.
- The address must remain the Start Address throughout programming.

Dual operations are not supported during the Buffer Enhanced Factory Program operation, and the command cannot be suspended.

The Buffer Enhanced Factory Program Command consists of three phases: Setup, Program and Verify, and Exit (refer to Table 8, page 21 for detail information).

Setup Phase

The Buffer Enhanced Factory Program command requires two Bus Write cycles to initiate the command:

- The first Bus Write cycle sets up the Buffer Enhanced Factory Program command.
- The second Bus Write cycle confirms the command.

After the confirm command is issued, read operations output the contents of the Status Register.

> ***Caution!*** The read Status Register command must not be issued as it is interpreted as data to program.

The Status Register Program/Erase Controller (P/E.C). Bit SR7 should be read to check that the P/E.C. is ready to proceed to the next phase.

If an error is detected, SR4 goes High (set to ‘1’) and the Buffer Enhanced Factory Program operation is terminated. See "Status Register," page 23 for details on the error.

Program and Verify Phase

The Program and Verify Phase requires 32 cycles to program the 32 words to the Write Buffer. Data is stored sequentially, starting at the first address of the Write Buffer until the Write Buffer is full (32 words). To program less than 32 words, the remaining words should be programmed with `FFFFh`.

Four successive steps are required to issue and execute the Program and Verify Phase of the command.

1. One Bus Write operation is used to latch the Start Address and the first word to be programmed. The Status Register Bank Write Status bit SR0 should be read to check that the P/E.C. is ready for the next word.
2. Each subsequent word to be programmed is latched with a new Bus Write operation. The address must remain the Start Address as the P/E.C. increments the address location.If any address not in the same block as the Start Address is given, the Program and Verify Phase terminates. Status Register bit SR0 should be read between each Bus Write cycle to check that the P/E.C. is ready for the next word.
3. After the Write Buffer is full, the data is programmed sequentially to the memory array. After the program operation, the device automatically verifies the data and reprograms if necessary.

   The Program and Verify phase can be repeated without re-issuing the command to program an additional 32-word locations as long as the address remains in the same block.
4. Finally, after all words, or the entire block are programmed, one Bus Write operation must be written to any address outside the block containing the Start Address to terminate Program and Verify Phase.

Status Register bit SR0 must be checked to determine whether the program operation is finished. The Status Register can be checked for errors at any time but must be checked after the entire block is programmed.

Exit Phase

Status Register P/E.C. bit SR7 is set to ‘1’ when the device exits the Buffer Enhanced Factory Program operation and returns to Read Status Register mode. A full Status Register check should be done to ensure that the block is successfully programmed. See "Status Register," page 23 for more details.

For optimum performance, the Buffer Enhanced Factory Program command should be limited to a maximum of 100 program/erase cycles per block. If this limit is exceeded, the internal algorithm continues to work properly, but some degradation in performance is possible. Typical program times are given in Table 21, page 44.

See Figure 45, page 79, for a suggested flowchart on using the Buffer Enhanced Factory Program command.

Program/Erase Suspend Command

The Program/Erase Suspend command is used to pause a Program or Block Erase operation. The command can be addressed to any bank.

The Program/Erase Resume command is required to restart the suspended operation. One Bus Write cycle is required to issue the Program/Erase Suspend command. After the Program/Erase Controller pauses, bits SR7, SR6 and/or SR2 of the Status Register are set to '1'.

The following commands are accepted during Program/Erase Suspend:

- Program/Erase Resume
- Read Array (data from erase-suspended block or program-suspended word is not valid)
- Read Status Register
- Read Electronic Signature
- Read CFI Query

Additionally, if the suspended operation is a Block Erase, then the following commands are also accepted:

- Clear Status Register
- Program (except in erase-suspended block)
- Buffer Program (except in erase suspended blocks)
- Block Lock
- Block Lock-Down
- Block Unlock
- Set Configuration Register

During an erase suspend, the block being erased can be protected by issuing Block Lock or Block Lock-Down commands. When the Program/Erase Resume command is issued, the operation completes.

It is possible to accumulate multiple suspend operations. For example, suspend an erase operation, start a program operation, suspend the program operation, then read the array.

If a Program command is issued during a Block Erase Suspend, the erase operation cannot be resumed until the program operation is complete.

The Program/Erase Suspend command does not change the read mode of the banks. If the suspended bank is in Read Status Register, Read Electronic Signature or Read CFI Query mode, the bank remains in that mode and outputs the corresponding data.

Refer to "Dual Operations and Multiple Bank Architecture," page 35 for detailed information about simultaneous operations allowed during Program/Erase Suspend.

During a Program/Erase Suspend, the device can be placed in standby mode by taking Chip Enable to V_{IH}. Program/erase is aborted if Reset ($\overline{RP}$) goes to V_{IL}.

See Figure 40, page 74, and Figure 42, page 76, for flowcharts for using the Program/Erase Suspend command.

Program/Erase Resume Command

The Program/Erase Resume command is used to restart the program or erase operation suspended by the Program/Erase Suspend command. One Bus Write cycle is required to issue the command and can be issued to any address.

The Program/Erase Resume command does not change the read mode of the banks. If the suspended bank is in Read Status Register, Read Electronic Signature or Read CFI Query mode, the bank remains in that mode and outputs the corresponding data.

If a Program command is issued during a Block Erase Suspend, then the erase cannot be resumed until the program operation is complete.

See Figure 40, page 74, and Figure 42, page 76, for flowcharts for using the Program/Erase Resume command.

Protection Register Program Command

The Protection Register Program command is used to program the user one-time-programmable (OTP) segments of the Protection Register and the two Protection Register Locks.

The device features 16 OTP segments of 128 bits and one OTP segment of 64 bits (Figure 8, page 22). The segments are programmed one word at a time. When shipped, all bits in the segment are set to '1'. The user can only program the bits to '0'.

Two Bus Write cycles are required to issue the Protection Register Program command:

- The first bus cycle sets up the Protection Register Program command.
- The second latches the address and data to be programmed to the Protection Register and starts the Program/Erase Controller.

Read operations to the bank being programmed output the Status Register content after the program operation starts. Attempting to program a previously protected Protection Register results in a Status Register error.

The Protection Register Program cannot be suspended. Dual operations between the Parameter Bank and the Protection Register memory space are not allowed (see Table 17, page 36, for details).

The two Protection Register Locks are used to protect the OTP segments from further modification. The protection of the OTP segments is not reversible. Refer to Figure 8, page 22, and Table 10, page 22, for details on the Lock bits.

See Figure 44, page 78, for a flowchart for using the Protection Register Program command.

Set Configuration Register Command

The Set Configuration Register command is used to write a new value to the Configuration Register. Two Bus Write cycles are required to issue the Set Configuration Register command:

- The first cycle sets up the Set Configuration Register command and the address corresponding to the Configuration Register content.
- The second cycle writes the Configuration Register data and the confirm command.

The Configuration Register data must be written as an address during the bus write cycles, that is A0 = CR0, A1 = CR1, …, A15 = CR15. Addresses A16–A22 are ignored. Read operations output the array content after the Set Configuration Register command is issued.

The Read Electronic Signature command is required to read the updated contents of the Configuration Register.

Block Lock Command

The Block Lock command is used to lock a block and prevent program or erase operations from changing the contents. All blocks are locked after power-up or reset.

Two Bus Write cycles are required to issue the Block Lock command:

- The first bus cycle sets up the Block Lock command.
- The second Bus Write cycle latches the block address and locks the block.

The lock status can be monitored for each block using the Read Electronic Signature command. Table 18, page 38 shows the Lock Status after issuing a Block Lock command.

After being set, the Block Lock bits remain set even after a hardware reset or power-down/power-up. They are cleared by a Block Unlock command.

Refer to "Block Locking," page 37 for a detailed explanation. See Figure 43, page 77, for a flowchart for using the Lock command.

Block Unlock Command

The Block Unlock command is used to unlock a block, allowing the block to be programmed or erased.

Two Bus Write cycles are required to issue the Block Unlock command:

- The first bus cycle sets up the Block Unlock command.
- The second Bus Write cycle latches the block address and unlocks the block.

The lock status can be monitored for each block using the Read Electronic Signature command. Table 18, page 38 shows the protection status after issuing a Block Unlock command.

Refer to the "Block Locking," page 37 for a detailed explanation and Figure 43, page 77, for a flowchart for using the Block Unlock command.

Block Lock-Down Command

The Block Lock-Down command is used to lock down a locked or unlocked block.

A locked-down block cannot be programmed or erased. The lock status of a locked-down block cannot be changed when $\overline{WP}$ is Low (at V_{IL}). When $\overline{WP}$ is High (at V_{IH}), the Lock-Down function is disabled, and the locked blocks can be individually unlocked by the Block Unlock command.

Two Bus Write cycles are required to issue the Block Lock-Down command:

- The first bus cycle sets up the Block Lock-Down command.
- The second Bus Write cycle latches the block address and locks-down the block.

The lock status can be monitored for each block using the Read Electronic Signature command.

Locked-Down blocks revert to the Locked (and not Locked-Down) state when the device is reset on power-down. Table 18 shows the Lock Status after issuing a Block Lock-Down command.

Refer to "Block Locking", for a detailed explanation and Figure 43, for a flowchart for using the Lock-Down command.

Table 7: **Standard Commands(1)**

Commands	Cycles	Bus Operations					
		First Cycle			Second Cycle		
		Op.	Add.	Data	Op.	Add.	Data
Read Array	1+	Write	BKA	FFh	Read	WA	RD
Read Status Register	1+	Write	BKA	70h	Read	BKA(2)	SRD
Read Electronic Signature	1+	Write	BKA	90h	Read	BKA(2)	ESD
Read CFI Query	1+	Write	BKA	98h	Read	BKA(2)	QD
Clear Status Register	1	Write	X	50h	–	–	–
Block Erase	2	Write	BKA or BA(3)	20h	Write	BA	D0h
Program	2	Write	BKA or WA(3)	40h or 10h	Write	WA	PD
Buffer Program(4)	n+4	Write	BA	E8h	Write	BA	n
		Write	PA_1	PD_1	Write	PA_2	PD_2
		Write	PA_{n+1}	PD_{n+1}	Write	X	D0h
Program/Erase Suspend	1	Write	X	B0h	–	–	–
Program/Erase Resume	1	Write	X	D0h	–	–	–
Protection Register Program	2	Write	PRA	C0h	Write	PRA	PRD
Set Configuration Register	2	Write	CRD	60h	Write	CRD	03h
Block Lock	2	Write	BKA or BA(3)	60h	Write	BA	01h
Block Unlock	2	Write	BKA or BA(3)	60h	Write	BA	D0h
Block Lock-Down	2	Write	BKA or BA(3)	60h	Write	BA	2FH

Notes:

1. X = Don't Care, WA = Word Address in targeted bank, RD =Read Data, SRD =Status Register Data, ESD = Electronic Signature Data, QD =Query Data, BA =Block Address, BKA = Bank Address, PD = Program Data, PA = Program address, PRA = Protection Register Address, PRD = Protection Register Data, CRD = Configuration Register Data.
2. Must be same bank as in the first cycle. The signature addresses are listed in Table 9, page 21.
3. Any address within the bank can be used.
4. n+1 is the number of words to be programmed.

Table 8: **Factory Commands**

Command	Phase	Cycles	Bus Write Operations(1)										
			First		Second		Third		...	Final – 1		Final	
			Add.	Data	Add.	Data	Add.	Data		Add.	Data	Add.	Data
Blank Check		2	BA	BCh	BA	CBh	–	–	...	–	–	–	–
Buffer Enhanced Factory Program	Setup	2	BKA or WA(2)	80h	WA_1	D0h	–	–	...	–	–	–	–
	Program/ Verify(3)	≥32	WA^1	PD_1	WA_1	PD_2	WA_1	PD_3	...	WA_1	PD_{31}	WA_1	PD_{32}
	Exit	1	NOT BA_1(4)	X	–	–	–	–	...	–	–	–	–

Notes:

1. WA = Word Address in targeted bank, BKA = Bank Address, PD =Program Data, BA = Block Address, X = Don't Care.
2. Any address within the bank can be used.
3. The Program/Verify phase can be executed any number of times as long as the data is to be programmed to the same block.
4. WA_1 is the Start Address, NOT BA1 = Not Block Address of WA_1.

Table 9: **Electronic Signature Codes**

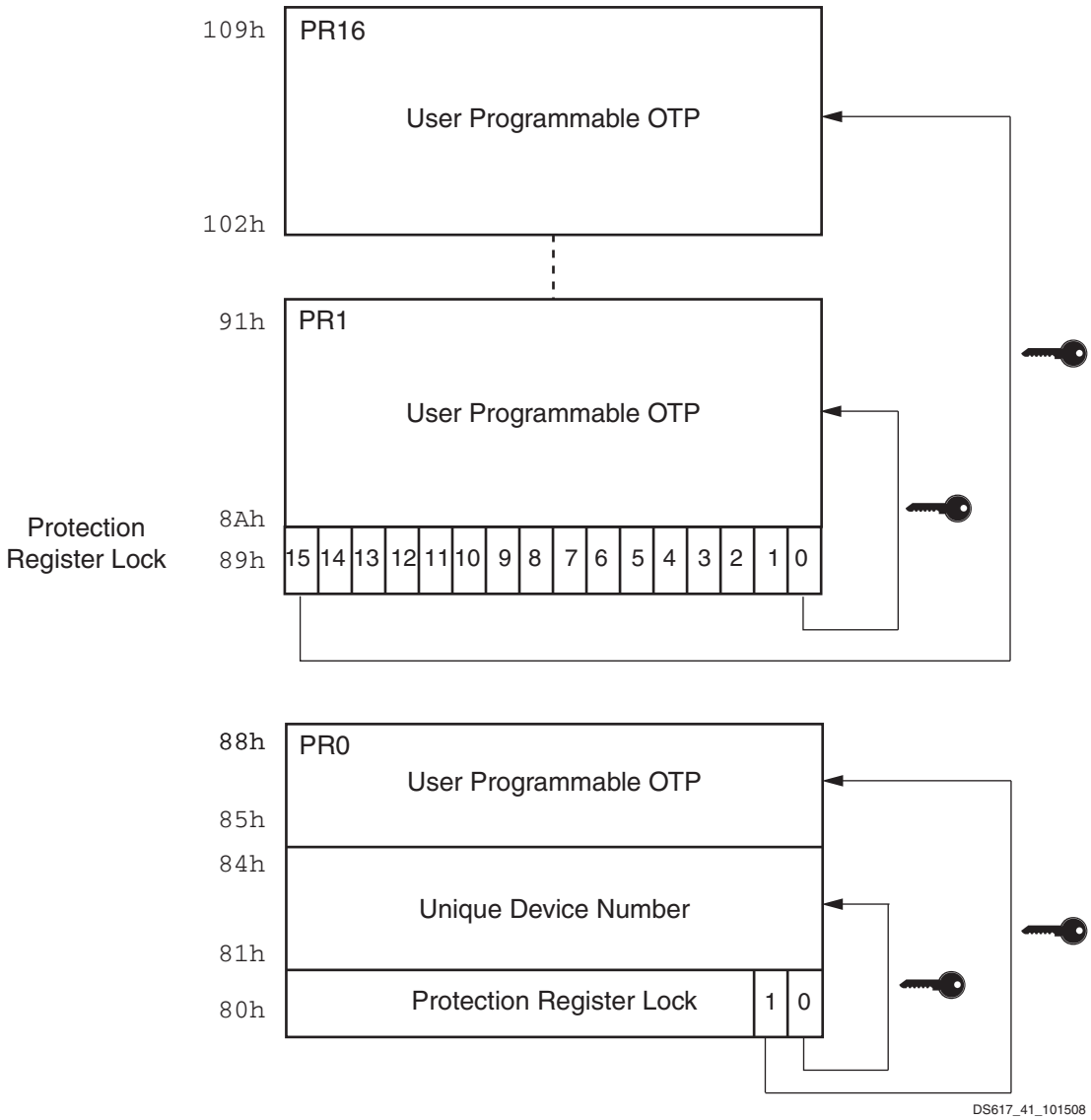
Code		Address (h)	Data (h)
Manufacturer Code		Bank Address + 000	0049
Device Code		Bank Address + 001	506B
Block Protection	Locked	Block Address + 002	0001
	Unlocked		0000
	Locked and Locked-Down		0003
	Unlocked and Locked-Down		0002
Configuration Register		Bank Address + 005	CR(1)
Protection Register PR0 Lock	Factory Default	Bank Address + 080	0002
	OTP Area Permanently Locked		0000
Protection Register PR0		Bank Address + 081 Bank Address + 084	Unique Device Number
		Bank Address + 085 Bank Address + 088	OTP Area
Protection Register PR1 through PR16 Lock		Bank Address + 089	PRLD(1)
Protection Registers PR1–PR16		Bank Address + 08A Bank Address + 109	OTP Area

Notes:

1. CR = Configuration Register, PRLD = Protection Register Lock Data.
2. The iMPACT software does not support reading of the electronic signature codes.

Table 10: **Protection Register Locks**

Lock			Description
Number	**Address**	**Bits**	
Lock 1	80h	bit 0	Preprogrammed to protect Unique Device Number, address 81h to 84h in PR0
		bit 1	Protects 64 bits of OTP segment, address 85h to 88h in PR0
		bits 2 to 15	Reserved
Lock 2	89h	bit 0	Protects 128 bits of OTP segment PR1
		bit 1	Protects 128 bits of OTP segment PR2
		bit 2	Protects 128 bits of OTP segment PR3
		–	–
		bit 13	Protects 128 bits of OTP segment PR14
		bit 14	Protects 128 bits of OTP segment PR15
		bit 15	Protects 128 bits of OTP segment PR16



Notes:

1. The iMPACT software does not support reading or writing of the protection register locks, OTP fields, or unique device number.

Figure 8: **Protection Register Memory Map**

Status Register

The Status Register provides information on the current or previous program or erase operations. A Read Status Register command is issued to read the contents of the Status Register, refer to "Read Status Register Command," page 14 for more details. To output the contents, the Status Register is latched and updated on the falling edge of the Chip Enable or Output Enable signals and can be read until Chip Enable or Output Enable returns to V_{IH}.

The Status Register can only be read using single asynchronous or synchronous reads. Bus Read operations from any address within the bank always read the Status Register during program and erase operations if no Read Array command is issued.

The various bits convey information about the status and any errors of the operation. Bits SR7, SR6, SR2 and SR0 give information on the status of the device and are set and reset by the device. Bits SR5, SR4, SR3 and SR1 give information on errors and are set by the device but must be reset by issuing a Clear Status Register command or a hardware reset.

If an error bit is set to ‘1’, the Status Register should be reset before issuing another command.

The bits in the Status Register are summarized in Table 11.

Table 11: **Status Register Bits**

Bit	Name	Type	Logic Level(1)	Definition	
SR7	P/E.C. Status	Status	'1'	Ready	
			'0'	Busy	
SR6	Erase Suspend Status	Status	'1'	Erase suspended	
			'0'	Erase In progress or completed	
SR5	Erase/Blank Check Status	Error	'1'	Erase/blank check error	
			'0'	Erase/blank check success	
SR4	Program Status	Error	'1'	Program error	
			'0'	Program success	
SR3	V_{PP} Status	Error	'1'	V_{PP} invalid, abort	
			'0'	V_{PP} OK	
SR2	Program Suspend Status	Status	'1'	Program suspended	
			'0'	Program In progress or completed	
SR1	Block Protection Status	Error	'1'	Program/erase on protected block, abort	
			'0'	No operation to protected block	
SR0	Bank Write Status	Status	'1'	SR7 = ‘1’	Not allowed
				SR7 = ‘0’	Program or erase operation in a bank other than the addressed bank
			'0'	SR7 = ‘1’	No program or erase operation in the device
				SR7 = ‘0’	Program or erase operation in addressed bank
	Multiple Word Program Status (Buffer Enhanced Factory Program mode)	Status	'1'	SR7 = ‘1’	Not allowed
				SR7 = ‘0’	The device is NOT ready for the next Buffer loading or is going to exit the BEFP mode
			'0'	SR7 = ‘1’	The device has exited the BEFP mode
				SR7 = ‘0’	The device is ready for the next Buffer loading

Notes:

1. Logic level '1' is High, '0' is Low.

Program/Erase Controller Status Bit (SR7)

The Program/Erase Controller Status bit indicates whether the Program/Erase Controller is active or inactive in any bank. When this bit is Low (set to ‘0’), the Program/Erase

Controller is active; when the bit is High (set to '1'), the controller is inactive, and the device is ready to process a new command.

The Program/Erase Controller Status bit is Low immediately after a Program/Erase Suspend command was issued until the controller pauses. After the Program/Erase Controller pauses the bit is High.

Erase Suspend Status Bit (SR6)

The Erase Suspend Status bit indicates that an erase operation is suspended. When this bit is High (set to '1'), a Program/Erase Suspend command was issued and the memory is waiting for a Program/Erase Resume command.

The Erase Suspend Status bit should only be considered valid when the Program/Erase Controller Status bit is High (Program/Erase Controller inactive). SR6 is set within the Erase Suspend Latency time of the Program/Erase Suspend command being issued; therefore, the memory can still complete the operation rather than entering the Suspend mode.

When a Program/Erase Resume command is issued, the Erase Suspend Status bit returns Low.

Erase/Blank Check Status Bit (SR5)

The Erase/Blank Check Status bit is used to identify if an error occurred during a Block Erase operation. When this bit is High (set to '1'), the Program/Erase Controller applied the maximum number of pulses to the block and still failed to verify that it erased correctly.

The Erase/Blank Check Status bit should be read after the Program/Erase Controller Status bit is High (Program/Erase Controller inactive).

The Erase/Blank Check Status bit is also used to indicate whether an error occurred during the Blank Check operation. If the data at one or more locations in the block where the Blank Check command was issued is different from `FFFFh`, SR5 is set to '1'.

After set High, the Erase/Blank Check Status bit must be set Low by a Clear Status Register command or a hardware reset before a new erase command is issued; otherwise, the new command appears to fail.

Program Status Bit (SR4)

The Program Status bit is used to identify if there is an error during a program operation. This bit should be read after the Program/Erase Controller Status bit is High (Program/Erase Controller inactive).

When the Program Status bit is High (set to '1'), the Program/Erase Controller applied the maximum number of pulses to the word and still failed to verify that it programmed correctly.

Attempting to program a '1' to an already programmed bit while $V_{PP} = V_{PPH}$ also sets the Program Status bit High. If V_{PP} is different from V_{PPH}, SR4 remains Low (set to '0'), and the attempt is not shown.

After set High, the Program Status bit must be set Low by a Clear Status Register command or a hardware reset before a new program command is issued; otherwise, the new command appears to fail.

V_{PP} Status Bit (SR3)

The V_{PP} Status bit is used to identify an invalid voltage on the V_{PP} pin during program and erase operations. The V_{PP} pin is only sampled at the beginning of a program or erase operation. Program and erase operations are not guaranteed if V_{PP} becomes invalid during an operation.

When the V_{PP} Status bit is Low (set to '0'), the voltage on the V_{PP} pin was sampled at a valid voltage.

When the V_{PP} Status bit is High (set to '1'), the V_{PP} pin has a voltage below the V_{PP} Lockout Voltage (V_{PPLK}). the memory is protected and program and erase operations cannot be performed.

After set High, the V_{PP} Status bit must be set Low by a Clear Status Register command or a hardware reset before a new program or erase command is issued; otherwise, the new command appears to fail.

Program Suspend Status Bit (SR2)

The Program Suspend Status bit indicates that a program operation is suspended. This bit should only be considered valid when the Program/Erase Controller Status bit is High (Program/Erase Controller inactive).

When the Program Suspend Status bit is High (set to '1'), a Program/Erase Suspend command was issued, and the memory is waiting for a Program/Erase Resume command.

SR2 is set within the Program Suspend Latency time of the Program/Erase Suspend command being issued; therefore, the memory can still complete the operation rather than entering the Suspend mode.

When a Program/Erase Resume command is issued, the Program Suspend Status bit returns Low.

Block Protection Status Bit (SR1)

The Block Protection Status bit is used to identify if a Program or Block Erase operation tried to modify the contents of a locked or locked-down block. When this bit is High (set to ‘1’), a program or erase operation was attempted on a locked or locked-down block.

After set High, the Block Protection Status bit must be set Low by a Clear Status Register command or a hardware reset before a new program or erase command is issued; otherwise, the new command appears to fail.

Bank Write/Multiple Word Program Status Bit (SR0)

The Bank Write Status bit indicates whether the addressed bank is busy performing a write or is ready to accept a new write command (a program or erase command). In Buffer Enhanced Factory Program mode, the Multiple Word Program bit shows if the device is ready to accept a new word to be programmed to the memory array.

The Bank Write Status bit should only be considered valid when the Program/Erase Controller Status SR7 is Low (set to ‘0’).

When both the Program/Erase Controller Status bit and the Bank Write Status bit are Low (set to ‘0’), the addressed bank is executing a program or erase operation. When the Program/Erase Controller Status bit is Low (set to ‘0’) and the Bank Write Status bit is High (set to ‘1’), a program or erase operation is being executed in a bank other than the one being addressed.

In Buffer Enhanced Factory Program mode, if Multiple Word Program Status bit is Low (set to ‘0’), the device is ready for the next word; if the Multiple Word Program Status bit is High (set to ‘1’) the device is not ready for the next word.

For further details on how to use the Status Register, see the Flowcharts and Pseudocodes provided in "Appendix C: Flowcharts and Pseudocodes," page 71.

Configuration Register

The Configuration Register is used to configure the type of bus access that the memory performs. Refer to "Read Modes," page 34 for details on read operations.

The Configuration Register is set through the Command Interface using the Set Configuration Register command. After a reset or power-up, the device is configured for Synchronous Read (CR15 = 0). The Configuration Register bits (Table 12, page 26) specify the selection of the burst length, burst type, burst X latency, and read operation. Refer to Figure 9, page 28 and Figure 10, page 30 for examples of synchronous burst configurations.

Table 12: **Configuration Register Bits**

Bits	Description	Value	Description
CR15	Read mode	0	Synchronous Read (default)
		1	Asynchronous Read
CR14	Reserved	0	
CR13–CR11	Clock Latency	010	2 clock latency(1)
		011	3 clock latency
		100	4 clock latency
		101	5 clock latency
		110	6 clock latency
		111	7 clock latency (default)
		Other configurations reserved	
CR10	Wait Polarity	0	READY_WAIT with Wait function (CR4 = 0) is active Low
		1	READY_WAIT with Wait function (CR4 = 0) is active High (default)
CR9	Data output configuration	0	Data held for 1 clock cycle (default)
		1	Data held for 2 clock cycles(1)
CR8	Wait Configuration	0	Wait active during wait state
		1	Wait active 1 clock cycle before wait state (default)
CR7	Burst Type	0	Reserved
		1	Sequential (default)
CR6	Valid Clock Edge	0	Falling clock edge
		1	Rising clock edge (default)
CR5	Reserved	0	–
CR4	Device_ready	0	READY_WAIT signal has the Wait function
		1	READY_WAIT signal has the Ready function (default)
CR3(2)	Wrap burst	0	Wrap
		1	No wrap (default)
CR2–CR0(2)	Burst Length	001	4 words
		010	8 words
		011	16 words
		111	Continuous (default)

Notes:

1. The combination X-Latency = 2, Data held for two clock cycles and Wait active one data cycle before the WAIT state is not supported.
2. CR3 (wrap/no wrap) bit has no effect when CR2-CR0 (burst length) bits are set to continuous burst mode. Platform Flash XL wraps to the first memory address after the device outputs the data from the last memory address.

Read Mode Select Bit (CR15)

The Read Select bit, CR15, is used to switch between Asynchronous and Synchronous Read operations. When this bit is set to ‘1’, read operations are asynchronous; when set to ‘0’, read operations are synchronous.

Synchronous Burst Read is supported in both parameter and main blocks and can be performed across banks.

On reset or power-up, the Read Select bit is set to ‘0’ for synchronous access.

X-Latency Bits (CR13-CR11)

The X-Latency bits are used during Synchronous Read operations to set the number of clock cycles between the address being latched and the first data becoming available (Figure 9). For correct operation the X-Latency bits can only assume the values listed in Table 12, page 26.

Table 13 shows how to set the X-Latency parameter, taking into account the speed class of the device and the frequency used to read the flash memory in synchronous mode.

Table 13: **X-latency Settings**

F_{MAX}	$T_K$min	X-Latency min
30 MHz	33 ns	3
40 MHz	25 ns	4
54 MHz	19 ns	5

Wait Polarity Bit (CR10)

The Wait Polarity bit is used to set the polarity of the READY_WAIT signal used in Synchronous Burst Read mode (with CR4 = 0). During this mode, the READY_WAIT signal indicates whether the data output is valid or a WAIT state must be inserted.

When the Wait Polarity bit is at '0', the READY_WAIT signal is active Low. When this bit is set to '1', the READY_WAIT signal is active High.

The CR10 Configuration Register bit becomes “don't care” if CR4 is set to ‘1’, in which case the READY_WAIT pin behaves like a READY pin (default value).

Data Output Configuration Bit (CR9)

The Data Output Configuration bit is used to configure the output to remain valid for either one or two clock cycles during synchronous mode. When this bit is ‘0’, the output data is valid for one clock cycle; when the bit is ‘1’, the output data is valid for two clock cycles.

The Data Output Configuration must be configured using the following condition:

$$t_K > t_{KQV} + t_{QVK\_CPU}$$

where:

t_K is the clock period

t_{QVK_CPU} is the data setup time required by the system CPU

t_{KQV} is the clock to data valid time.

If this condition is not satisfied, the Data Output Configuration bit should be set to ‘1’ for two clock cycles (Figure 9, page 28).

Wait Configuration Bit (CR8)

The Wait Configuration bit is used to control the timing of the READY_WAIT signal when configured as an output with the Wait function (in Synchronous Burst Read mode).

When READY_WAIT is asserted, data is not valid; when READY_WAIT is deasserted, data is valid.

When the Wait Configuration bit is Low (reset to ‘0’), the READY_WAIT signal (configured as an output with the Wait function) is asserted during the WAIT state. When the Wait Configuration bit is High (set to ‘1’), the READY_WAIT output pin is asserted one data cycle before the WAIT state.

Burst Type Bit (CR7)

The Burst Type bit determines the sequence of addresses read during Synchronous Burst Read operations. This bit is High (set to ‘1’) as the memory outputs from sequential addresses only.

See Table 14, page 29, for the sequence of addresses output from a given starting address in sequential mode.

Valid Clock Edge Bit (CR6)

The Valid Clock Edge bit (CR6) is used to configure the active edge of the Clock (K) during synchronous read operations. When this bit is Low (set to ‘0’), the falling edge of the Clock is the active edge; when High (set to ‘1’), the rising edge of the Clock is the active edge.

READY_WAIT Bit (CR4)

The READY_WAIT Configuration Register bit is a user-configurable bit. The default value is ‘1’, where the READY_WAIT signal is configured as an input with the Ready function (CR4 = '1'). This particular configuration allows the use of the READY_WAIT signal for handshaking during the configuration sequence and during a Reset ($\overline{RP}$) pulse as the device holds the pin Low until the entire internal configuration of the device finishes. With CR4 = 1, the external pin can also be used by the end user to retrigger the first address latching sequence (FALS), simply by applying a High, a Low, and then a High pulse on the READY_WAIT pin. See "First Address Latching Sequence," page 41.

When CR4 = '0', the READY_WAIT signal assumes the standard WAIT functionality.