

# 基于 Nios II 的井下静态随机存取存储器管理

付绍凯<sup>1,2</sup>, 卢俊强<sup>1,2</sup>, 门百永<sup>1,2</sup>, 鞠晓东<sup>1,2</sup>, 陈光建<sup>3</sup>

(1. 中国石油大学(北京)油气资源与探测国家重点实验室, 北京 102249;

2. 北京市地球探测与信息技术重点实验室, 北京 102249;

3. 中国石油集团测井有限公司生产测井中心, 河北 任丘 062552)

**摘要:** 针对测井仪器中高速数据缓存的问题, 提出了一种基于 Nios II 的静态随机存储器 (Static Random-Access Memory, SRAM) 控制器的 IP 核的设计方法, 详细介绍了 IP 核的设计和实现过程并基于一定的硬件平台对 SRAM 控制器 IP 核的稳定性进行了测试; 测试结果表明, 所设计的 SRAM 控制器 IP 核能够在高温环境下实现对 SRAM 的有效管理而且通过修改配置参数就可应用在不同位宽和容量的 SRAM 上且运行稳定; 该设计不仅实现测井仪器中高速数据缓存的管理, 也提供了一套 SRAM 选型测试系统; 另外设计介绍的可编程片上系统 (System-on-a-Programmable-Chip, SOPC) 开发流程对其它 SOPC 系统开发也具有一定的参考价值。

**关键词:** 测井仪器; 静态随机存储器; Nios II; IP 核; 可编程片上系统

## Static Random-Access Memory Management Based on Nios II

Fu Shaokai<sup>1,2</sup>, Lu Junqiang<sup>1,2</sup>, Men Baiyong<sup>1,2</sup>, Ju Xiaodong<sup>1,2</sup>, Chen Guangjian<sup>1,2</sup>

(1. State Key Laboratory of Petroleum Resources and Prospecting, China University of Petroleum (Beijing), Beijing 102249, China; 2. Earth Explorer and Information Technology Laboratory, Beijing 102249, China;

3. Production logging center of CNPC logging Co. Ltd, Renqiu 062552, China)

**Abstract:** Aiming at the problem of high speed data cache in logging tool, proposes a design method of IP core of SRAM controller based on Nios II, introduces the design and implementation process of IP core in detail, and tests the stability of IP core of SRAM controller based on certain hardware platform. The test results show that the designed SRAM controller IP core can effectively manage SRAM in high temperature environment and can be applied to SRAM with different bit widths and capacities by modifying configuration parameters and running stably. The design not only realizes the management of high speed data cache in logging tool, but also introduces a set of SRAM selection test system. In addition, the development process of SOPC system introduced in the design has certain reference value for the development of other SOPC systems.

**Keywords:** logging tool; SRAM; Nios II; IP core; SOPC

## 0 引言

随着大规模阵列化成像测井仪器的出现, 井下仪器采集的数据量大幅度增加, 对数据的实时性要求越来越高, 静态数据存储 SRAM 因具有较高的性能、较低的功耗和较快的速度被大量应用于井下电路设计中。在测井仪器电路设计和测试中需要用到不同类型的 SRAM 存储器, 设计一种用于井下仪器电路中可配置的 SRAM 控制器的解决方案有利于提高仪器开发和测试的效率, 并能够较好地适用于仪器功能的扩展和升级。

Altera 的 Nios II 核是用户可随意配置和构建的 32 位嵌入式处理器 IP 核, 采用 Avalon 总线结构的通信接口。利用 Altera 开发环境下的 Quartus II 和 SOPC Builder 工具可以根据设计要求对 Nios II 处理器内核及其外围设备进行搭建, 使该嵌入式系统全面满足用户系统设计的要求。由于处理器是软核形式, 具有很大的灵活性, 采用 Nios II 处理器, 用户将不再局限于 ASIC 处理器技术, 而是根据自己要求定制处理器, 按照需要选择合适的外设设备、存储器以及各种接口, 很受用户们欢迎。NiosII 开发包含一套通用的外设和接口库, 另外利用 SOPC Builder 软件中的用户逻辑接口向导可以生成自己的定制外设, 从而实现现有处理器无法实现的嵌入式处理器配置, 得到用户所需的结果<sup>[1]</sup>。

本文结合 Altera 公司的 Cyclone II 系列 FPGA 芯片, 以 SRAM (CY7C1011DV33) 为例, 设计一款应用于测井仪器中 SRAM 控制器 IP 核, 并基于一定硬件平台通过构建 SOPC (System On Programmable Chip, 可编程的片上系统) 系统对设计的 IP 核进行了相关测试。该测试平台也可应用在对 SRAM 的选型中<sup>[2-3]</sup>。

收稿日期: 2019-03-18; 修回日期: 2019-04-13。

基金项目: 国家科技重大专项 (2017ZX05019006、2017ZX05019001); 国家自然科学基金项目 (11734017); 中国石油科技创新基金项目 (2016D-5007-0303); 中国石油大学(北京)基金 (2462016YJRC020)。

作者简介: 付绍凯(1994-), 男, 山东潍坊人, 硕士研究生, 主要从事声波测井仪器研发方向的研究。

通讯作者: 卢俊强(1978-), 男, 甘肃临洮人, 博士, 副教授, 主要从事石油测井方法和装备方面的教学方向的研究。

## 1 SRAM 应用

以声波测井仪器为例，常规声波测井仪器的电子系统由主控电路、声激励电路、声信号采集电路组成<sup>[4]</sup>，各电路采用自定义高速总线 TMB 互联。主控电路作为整个系统的控制中心和数据处理中心，通常采用 DSP+FPGA/CPLD+存储器的架构实现，如图 1 所示。在 DSP 中运行主控制程序，控制仪器发射和数据采集、处理和上传等。井下多通道数据采集使用 FPGA 作为接口控制逻辑，能够实现高速总线的节点协议和数据缓存。采集数据没有立即经过 DSP 处理后写入存储器中而是通过 SRAM 进行数据缓存，这样能够避免 DSP 频繁干预采集过程从而降低数据处理速度。另外井下高温高压的恶劣环境对大容量耐高温的 SRAM 选型要求难度越来越大。DSP 可以利用专用外部存储器接口来直接访问 SRAM，而 FPGA 对 SRAM 需要自行设计实现<sup>[5]</sup>。

一般 FPGA 对 SRAM 管理采用 Verilog/VHDL 硬件语言设计 SRAM 管理模块实现，过程繁琐且灵活性差，如果仪器更换其他型号或容量的芯片，需要重新设计该模块，不利于仪器升级改造。设计一款可灵活配置的 SRAM 控制器，实现对不同数据宽度和容量的 SRAM 的控制管理，可提高控制器的复用性，灵活访问 SRAM 空间。

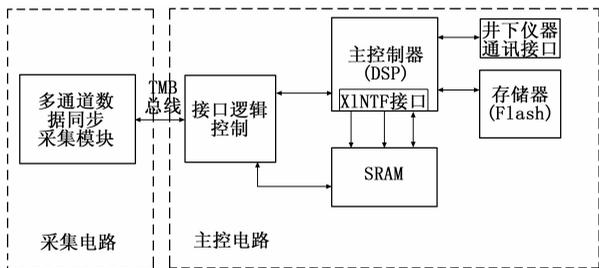


图 1 声波测井仪器电子系统

## 2 SRAM 控制器 IP 核设计

### 2.1 SRAM 控制时序设计

以 CY7C1011DV33 为例，对该 SRAM 芯片进行分析，SRAM 芯片包含  $A_0 \sim A_{16}$  共 17 根地址线， $IO_0 \sim IO_{15}$  共 16 根数据线，片选信号/CE，读使能信号/OE，写使能信号/WE 和高低字节选择信号/BHE 和/BLE<sup>[6]</sup>。SRAM 操作主要包括写操作和读操作：通过片选和读/写控制信号可实现 SRAM 写入和读出数据，读写时序如图 2 所示（前半部分为写： $t_{WC_{min}}=10\text{ ns}$ ，后半部分为读： $t_{RC_{min}}=10\text{ ns}$ ）。

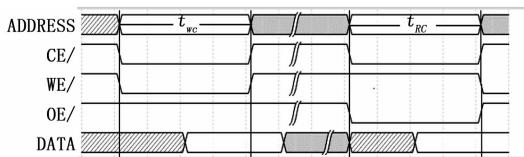


图 2 SRAM 读写时序图

用户定制外设有两种可行的办法：一种是利用 SOPC Builder 提供的元件编辑器在图形化用户界面下将硬件描述语言（Verilog 或者 VHDL）描述的用户逻辑封装为一个

SOPC Builder 元件（也可称为 IP 核）；另一种是在 Altera 提供的 IP 核的基础上来修改（主要是通过修改 Class. ptf 文件）满足用户实现其功能<sup>[1,7]</sup>。本设计采用第一种方式<sup>[8]</sup>，根据 SRAM 读写操作要求采用 Verilog 语言编写元件逻辑的硬件描述逻辑（HDL 文件）。

下面附上是硬件描述逻辑的 verilog 代码。由于篇幅限制，详细代码略去，这里只附上一些重要部分：

——以下是端口部分说明

```
module SRAM_xBit_xK(
// Host Data
oDATA,iDATA,iADDR,
iWE_N,iOE_N,
iCE_N,iCLK,
iBE_N,
// SRAM
SRAM_DQ,
SRAM_ADDR,
SRAM_UB_N,
SRAM_LB_N,
SRAM_WE_N,
SRAM_CE_N,
SRAM_OE_N
);
```

### 2.2 SRAM 控制器 IP 核的设计

将前面设计好的 SRAM 控制逻辑封装为 SOPC Builder 元件主要包括以下几个步骤<sup>[1]</sup>：

1) 在 SOPC Builder 中，单击菜单选项 File 下的 New Component，打开创建新元件向导，从 Introduction 选项卡页面点击 next 进入创建过程。

2) 在 HDL Files 选项卡中，单击 Add HDL Files 按钮，加入设计好的硬件描述语言文件模块（HDL 文件），点击 next 进入 Signals 选项卡。

3) 在 Signals 选项卡，按照图 3 设置信号类型：Name 一栏是用户定义的信号类型；Interface 是接口类型名称；Signal Type 一栏是信号类型，其中“\_n”表示低电平有效；Width 为信号的宽度，这里在设计 HDL 文件时将数据宽度和地址宽度分别宏定义为 DATA\_LEN 和 ADDR\_LEN 是为了在使用时能够灵活配置 SRAM 控制器 IP 核；Direction 一栏为信号的方向选择，注意这是对外部器件而言的。

4) 选择 Interface 选项卡。对于 avalon\_slave\_0 选项卡选择 Associated Clock 为 clock\_sink，Timing 选项卡（如图 4 所示）可以根据具体器件的读写参数进行配置，单位可以是时钟周期也可以为 ns。读写时序参数关乎 SRAM 读写速度，可通过不断调整选择最佳参数。设计使用的 SRAM 芯片的读/写时序图如图 2 所示，为了能够在高温环境中实现对 SRAM 的稳定读写，本设计将此控制器的读写时间设定为 2 个时钟周期。在 Deprecated 选项下，因为 SRAM 是易失性存储器，所以一定要勾选 Memory device 选项，Slave addressing 选择 NATIVE 即静态地址对齐方

Name	Interface	Signal Type	Width	Direction
oDATA	avalon_slave_0	readdata	DATA_LEN	output
iDATA	avalon_slave_0	writedata	DATA_LEN	input
iADDR	avalon_slave_0	address	ADDR_LEN	input
iWE_N	avalon_slave_0	write_n	1	input
iOE_N	avalon_slave_0	read_n	1	input
iCE_N	avalon_slave_0	chipselct_n	1	input
iBE_N	avalon_slave_0	byteenable_n	DATA_LEN/8	input
SRAM_ADDR	conduit_end	export	ADDR_LEN	output
SRAM_UB_N	conduit_end	export	1	output
SRAM_LB_N	conduit_end	export	1	output
SRAM_WE_N	conduit_end	export	1	output
SRAM_CE_N	conduit_end	export	1	output
SRAM_OE_N	conduit_end	export	1	output
SRAM_DQ	conduit_end	export	DATA_LEN	bidir
iCLK	clock_sink	clk	1	input

图 3 SRAM 控制器 Signals 选项卡

式。对于“clock\_sink”和“conduit\_end”选项栏可默认设置不用修改。

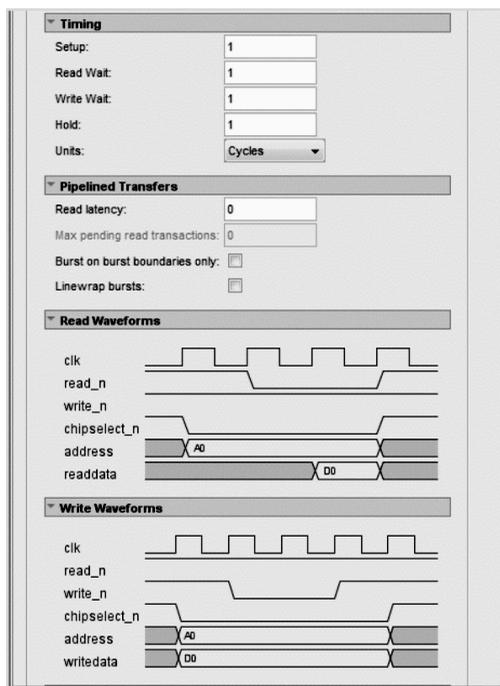


图 4 SRAM 读写时序配置

5) 最后进入 Component Wizard 选项卡, 该选项卡主要设置 SRAM 控制器 IP 核的名称、版本等说明信息, 可根据实际情况进行填写。单击 Finish, 至此完成对 SRAM 控制器 IP 核的创建, 可在 Component Library 一栏中找到刚刚创建的 IP 核, 设计完成的 SRAM 控制器 IP 核可以灵活配置数据和地址宽度。

### 3 SRAM 控制器 IP 核测试

测试在以 Altera Cyclone II 系列的 EP2C8T144I8N FPGA 为核心的最小系统板的硬件平台上实现的。Cyclone II 系列 FPGA 具有很高的性能和极低的功耗, 而且性价比较高, 能够提供多种功能, 还可以扩展外围设备、存储器、I/O 以及嵌入式处理器。同时提供了丰富的可编程资源和可编程 I/O 资源。EP2C8T144I8N 系列 FPGA 提供高达 8256

个逻辑单元, 36 块 M4KRAM 块即 165888bits 的 RAM 资源, 2 个 PLLS 及 144 个引脚。此款 FPGA 支持多种配置模式: 包括主动串行, 被动方式和 JTAG 方式<sup>[9]</sup>。Cyclone II 系列 FPGA 为了固化程序需要选择合适的配置器件, EP2C8 系列支持 EPCS4、EPCS16 和 EPCS64。

#### 3.1 硬件设计

底层板通过 FPGA+晶振+配置芯片+电源搭建形成最小系统板。以 EP2C8T144I8N 为核心, 外围设备包括: 1.2 V 的核心电压和 3.3 V 的 I/O 电压, 外部直流电源经过滤波电路可直接给芯片供电, 地端统一共地; 20 MHz 的外部有源晶振 CSTOSC, 其中引脚 2 接地, 引脚 3 为时钟输出信号, 引脚 4 接 3.3 V 电源。引脚 3 的时钟输出和 FPGA 的时钟输入引脚之间串联了一个电阻用于保证时钟波形稳定; 配置芯片采用 Altera 公司生产的 EPCS4N, 可用于存储程序代码、非易失性程序代码和 FPAG 的配置数据; JTAG 配置电路, 按照设计规范, 数据输入信号和模式选择信号分别接 1 kΩ 的上拉电阻, 时钟信号 TCK 需要接 1 kΩ 的下拉电阻, JTAG 插座型号为 DC3-10P; UART 接口, 直接扩展两个 I/O 引脚为串口通信接口。设计没有采用硬件复位电路, 复位是通过软件实现。将 FPAG 余下的 I/O 引脚进行外扩, 扩展为两排 12x2 的直插插座。

SRAM 测试底座是将芯片引脚进行扩展, 其中片选、读/写使能信号串联 22.9 Ω 的电阻后与接口引脚相连。SRAM 电源采用单独供电。利用底层板预留的扩展接口, 采用搭积木方式, 将 SRAM 测试插座通过插针直插到底层最小系统板上进行测试。

#### 3.2 SOPC 系统搭建

SOPC 是 Altera 公司提出的一种灵活、高效的片上系统解决方案。它将处理器、存储器、I/O 口、各种接口等系统设计所需要的功能模块集成到 FPGA 上, 构成一个可编程的片上系统。SOPC 系统开发流程主要包括硬件开发和软件开发两部分, 其中硬件开发是在 QuartusII 中利用 SOPCBuilder 搭建 NiosII 系统, 然后在 QuartusII 中进行综合布线、引脚分配等形成完整系统; 软件开发是根据设计需求在 NiosII 的 SBT 环境中利用 C/C++ 语言和系统自带 API 函数编写程序<sup>[10-11]</sup>, 图 5 为设计的 SOPC 系统框图。

##### 3.2.1 硬件开发

本文设计的系统硬件部分主要包括: NiosII 嵌入式系统、PLL 模块、复位模块等。首先需要构建 NiosII 嵌入式系统: 在 Quartus II 中新建工程并建立顶层模块后, 利用 Quartus II 的组件 SOPC Builder 创建完整的 Nios II 系统。启动 SOPC Builder 后, 首先指定目标 FPGA 和时钟设置为 50 MHz, 然后选择合适的 Nios II 处理器内核: 在 AvalonComponents 目录下双击 Nios II Process or—Altera Coporation, 弹出 Altera Nios II 配置向导对话框, 选择 Nios II Core 选项卡, 可以看到 Nios II 系列支持 3 种类型的 CPU (Nios II/e、Nios II/s、Nios II/f), 本设计选择标准型 CPU 核 Nios II/s, 在 Cachesand-MemoryInterface 选项卡中指定 Instruction Cache 为 4 KB,

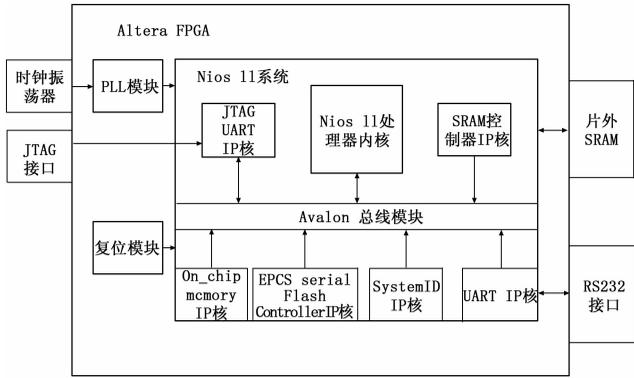


图 5 SOPC 系统框图

DATACache 为 2 KB, 在 JTAG Debug Module 选项卡中选择 Level 型 JTAG, 方便调试。完成 Nios II 处理器配置, 生成一个带有 JTAG 调试接口的 Nios II/s 型处理器核。添加 JTAG UART 控制器 IP 核可用于 PC 机对 FPGA 的程序下载和调试。然后添加 8 KB 片内 RAM 用于程序运行空间。另外因需要固化程序, 添加系统提供的 epcs\_flash\_controller 控制器 IP 核, EPCS 控制器自带 Boot-Loader 代码。因需要与上位机通讯, 添加系统提供的 UARTIP 核<sup>[12]</sup>。添加 System ID 外设 (当 SOPC Builder 生成 NiosII 系统时, 将为该 NiosII 系统生成一个标识符 (ID 号), 供 IDE 编译器和用户所设计的软件程序判断是否与目标系统匹配)。最后添加前面设计的 SRAMIP 核。将所需 IP 核都添加到 Nios II 系统后生成的界面如图 6 所示。

Use	Conn...	Module Name	Description	Clock	Base	End	Tags	IRQ
✓		cpu	Nios II Processor					
		instruction_master	Avalon Memory Mapped Master	clk_0				
		data_master	Avalon Memory Mapped Master	clk_0				
		jtag_debug_module	Avalon Memory Mapped Slave	clk_0	0x00001000	0x000017ff	IRQ 31	
		jtag_uart	JTAG UART	clk_0	0x00001820	0x00001827		
		avalon_jtag_slave	Avalon Memory Mapped Slave	clk_0	0x00004000	0x00005fff		
		onchip_memory2_ram_s1	On-Chip Memory (RAM or ROM)	clk_0	0x00000000	0x000007ff		
		epcs_flash_controller	EPCS Serial Flash Controller	clk_0	0x00000000	0x000007ff		
		epcs_control_port	Avalon Memory Mapped Slave	clk_0	0x00001828	0x0000182f		
		sysaid	System ID Peripheral	clk_0	0x00001800	0x0000191f		
		control_slave	Avalon Memory Mapped Slave	clk_0	0x00001800	0x0000191f		
		uart_rs232_s1	UART (RS-232 Serial Port)	clk_0	0x00001800	0x0000191f		
		sram_new_0	Avalon Memory Mapped Slave	clk_0	0x00200000	0x003fffff		
		avalon_slave_0	Avalon Memory Mapped Slave	clk_0	0x00200000	0x003fffff		

图 6 SRAM 控制器 IP 核添加到 Nios II 系统后生成的界面

至此, 系统中已经添加完所需的元件, 接下来需要为每个外设分配基地址和中断请求优先级。可以利用 SOPC Builder 的 System 菜单提供的 Auto-Assign Base Address 命令进行自动基地址分配即可。本系统中所有外设有使用中断, 无需进行中断优先级分配。然后需要选中 Nios II Processor 中的 CORE Nios II 选项卡进行系统的复位地址和异常向量的设置。最后选择 System Generation 选项卡单击 Generate 进行系统生成编, 在系统生成过程中, SOPC Builder 会执行一系列操作, 包括: 生成所有部件 Verilog HDL 源文件、硬件部件、连接部件的片内总线以及为系统生成 IDE 软件开发所需要的 HAL 和 C 文件等。

生成过程的时间一般需要几分钟, 当选项卡中显示“SUCCESS; SYSTEM GENERATION COMPLETED”信

息时, 代表系统生成结束, 将会输出一个 .bsf 文件。通过在顶层模块中左键双击, 在出现的 Symbol 选项卡中的 Project 文件夹下将 SOPC Builder 搭建 Nios II 系统时生成的 Symbol 模块加入到工程创建的顶层模块 BDF 文件中。同样在 Symbol 选项卡中加入 input/output/bidir 引脚和所需的 PLL 模块 (PLL 备频到 50 MHz)。最后添加自行设计的自动复位模块, 将各功能模块用连接线连起来组成系统功能原理图, 并进行相应的引脚分配。这一过程类似于传统硬件电路设计中在 PCB 上连接各功能模块。完成硬件系统设计并编译无误后就可以将硬件设计下载到目标 FPGA 中。通过 USB-Blaster 下载电缆连接目标板和 PC 机, 接通电源后在 Quartus II 软件中选择 Tool-Programmer, 将配置文件 .sof 下载到 FPGA 中。完成硬件设计后, 接下来需要进行程序开发。

### 3.2.2 软件开发

Nios II 软件构建工具 Nios II Software Build Tools 是一组功能强大的集成开发环境, 利用它们可以管理应用程序、硬件抽象层 (HAL) (或者称为板级支持包 BSP) 和软件库的编译选项。硬件抽象层是底层硬件和应用程序开发之间的系统库, 用户可以利用 HAL 提供的各种函数编写应用程序。HAL 在新建 SBT 工程时自动生成, 与硬件开发紧密相关。所有的软件开发任务都是在 Nios II SBT 下完成的, 包括代码编写、编译和程序调试。调试无误的程序可以将生成的可执行文件 .elf 固化到配置芯片中。

打开 Nios IISBT。启动一个新的 Nios II Application and BSP from Template 工程时, Nios IISBT 需要使用 SOPC Builder 系统文件 (.sopcinfo) 进行创建。创建工程成功后, 在 SBT 主界面左侧会显示两个新的工程文件夹: sram 和 sram\_bsp, 其中 sram 是应用工程, sram\_bsp 是描述硬件的系统库。在 sram 工程文件夹下打开源文件进行 C 程序开发<sup>[13]</sup>。程序编写并编译成功后在硬件下载成功的前提下可以利用 Nios IISBT 的菜单栏 Nios II 中的 Flash Program 将工程文件下载到目标 FPGA 中, 下载成功后重新上电可以观察程序运行情况。

为了测试 SRAM 控制器 IP 核可用性, 软件设计是通过往 SRAM 中循环写入测井数据并将数据读出, 比较写入与读出数据的一致性, 将比较结果发送到 PC 机进行实时监测, 具体测试流程如图 7 所示。为了实现测试该 IP 的稳定性和 SRAM 耐温耐压特性, 该测试平台是放置在加温箱中进行连续测试。下位机 (测试平台) 可以通过串口接收上位机发出的命令, 通过对命令进行判别来进行测试模式切换, 分别进行连续地址写入和随机地址写入两种测试。

### 3.3 测试结果

将设计完成的测试平台放置于加温烘箱内, 底层板所需的 1.2 V 和 3.3 V 电源及 SRAM 测试板所需的 3.3 V 电源通过高温导线引出直接连接到直流电源上。UART 接口的 TX 和 RX 通过高温导线引出后连接到电平转换盒子再通过 USB 转串口线直接与 PC 机相连。将待测 SRAM 芯片放

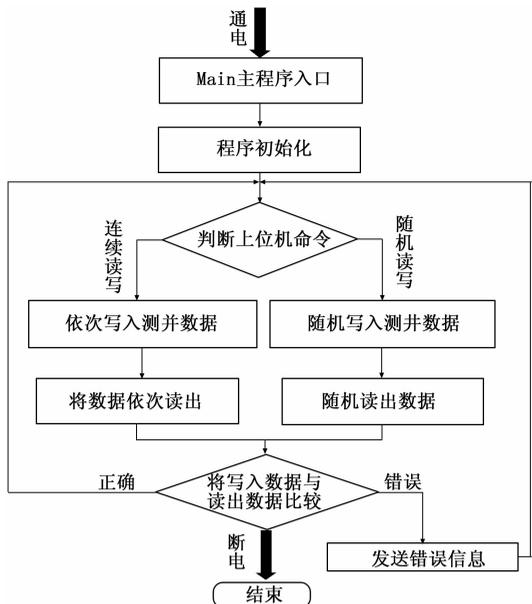


图 7 测试流程

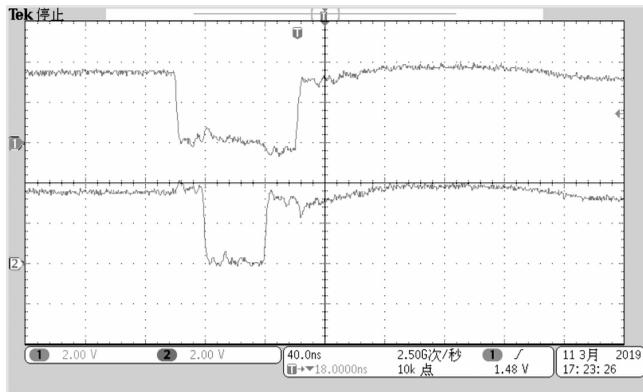
置于测试板上, 启动外部直流电源对设备进行供电, 通过 PC 机中的串口调试助手观察设备运行情况。

将 SRAM 测试系统放置在加温箱内进行连续高温测试, 测试温度梯度为 85 (30 min) - 125 (30 min) - 155 (30 min) - 175 (2 h) °C, 在 175 °C 连续工作 2 h 后测得的读写时序如图 8 所示 (其中第 1 道为使能信号, 第 2 道为写使能 (a) /读使能 (b) 信号), 从图中可以看出读写周期均为 2 个时钟周期, 读写时序平稳且毛刺少。通过 PC 机串口调试助手打印的数据写入、读出信息能够判断该控制器 IP 核实现了高温环境下 SRAM 数据写入、读出无误, SRAM 控制器 IP 核在高温环境下可以连续正常工作。

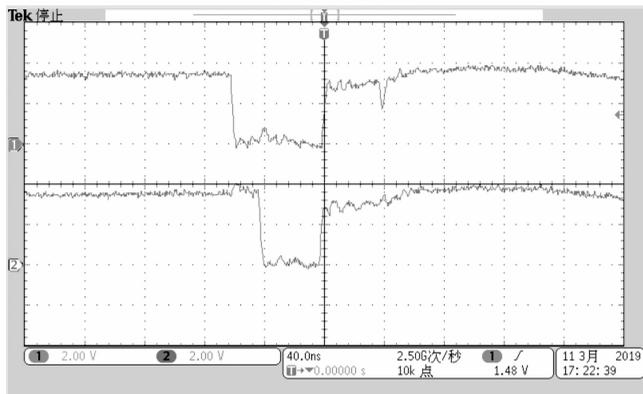
另外, 借助此测试平台可以实现测井仪器中 SRAM 芯片高温选型, 芯片插座支持相同封装的不同数据宽度和容量的 SRAM 芯片, 通过简单修改 IP 核配置即可进行不同类型 SRAM 读写测试。整个测试流程图可选择图 7 所示的流程。通过 UART 接口, 下位机把加温过程中的测试信息传输到 PC 机上进行显示, 可以观察芯片的工作状态, 进而判断芯片的耐温特性。

#### 4 结论

本文重点介绍了基于 Nios II 的 SRAM 控制器 IP 核的设计过程, 并将设计好的 SRAM 控制器 IP 核加入到完整的 NiosII 系统中进行了测试。测试结果表明该 IP 核能够很好地实现对 SRAM 的读写控制, 并能稳定的运行在高温环境中。对于不同位宽和容量的 SRAM 只需修改配置参数就可实现移植, 提高了 IP 核的重复利用性, 避免重复开发。该设计不仅解决了测井仪器中高速数据缓存的管理问题, 也提供了一套 SRAM 选型测试系统, 能够较好地适用于仪器功能的扩展和升级, 对其它 SOPC 系统开发也具有一定的参考价值。



(a) 写入过程的时序图



(b) 读出过程时序图  
图 8 读写时序图

#### 参考文献:

- [1] 周立功. SOPC 嵌入式系统基础教程 [M]. 北京: 北京航空航天大学出版社, 2006.
- [2] 王任重. 基于 SOPC 技术芯片自动测试系统的设计与实现 [D]. 湘潭: 湘潭大学, 2018.
- [3] 刘忠超. 基于 NIOS II 嵌入式系统的芯片自动测试系统设计 [J]. 电子质量, 2016 (12): 84 - 87.
- [4] 卢俊强, 鞠晓东, 乔文孝, 等. 数字信号处理器在随钻声波测井仪中的应用 [J]. 测井技术, 2013, 37 (05): 527 - 530.
- [5] 周新淳. 基于 DSP+FPGA 的实时信号采集系统设计与实现 [J]. 计算机测量与控制, 2017, 25 (08): 210 - 213.
- [6] CY1011DV33 datasheet [Z]. 2006 - 07 - 14.
- [7] 梁浩, 陈波, 良华. 基于 NIOSII 的 VGA 控制器 IP 核的设计 [J]. 工矿自动化, 2010 (3): 73 - 74.
- [8] 刘兆庆, 黄敏, 罗杰俊. SOPC 中定制功能实现方法比较及性能分析 [J]. 测控技术, 2010, 29 (10): 103 - 106.
- [9] Cyclone II Device Handbook, Volume 1 [Z]. <http://www.altera.com>, 2008.
- [10] 余毅, 马骋, 贾惠波. 基于 Nios II 的 PWM 控制电路设计 [J]. 微计算机信息 (测控自动化) 2008, 24: 7.
- [11] 林振营, 马刚. 基于 NiosII 的 SOPC 嵌入式系统开发解决方案 [J]. 现代电子技术, 2007 (6): 79 - 80.
- [12] 陈建, 张天宏, 刘冬冬. 基于 SOPC 的微型涡喷发动机控制器设计与验证 [J]. 测控技术, 2012, 31 (7): 50 - 52.
- [13] 潘宗树. 基于 Nios II 的 SOPC 系统设计与研究 [D]. 武汉: 武汉科技大学, 2007.