Modern Applied Science
www.ccsenet.org/mas.html
Vol. 1, No. 2
July 2007

Interface between the Embedded Processor Nios Ⅱ and the TDC Module and its Application

Jiawei Zhang, Shengjun Zhai, Jiuling Lin
College of Science, Tianjin Polytechnic University, Tianjin 300160, China
Email: zjw0128 @tom.com

Abstract

This paper introduces a design method for SOPC (System on Programmable Chip) based on embedded Nios II soft-core processor. The interface between the soft-core processor Nios II and the TDC module is given, and the initial programming technique and Nios II application programme is explained at the same time.

Keywords: Nios II, FPGA, SOPC, TDC

Introduction

High-resolution time interval measurement has been widely applied in the aerospace, communications, digital television broadband radar, logic analyzer, Digital Storage Oscilloscope modern testing equipment etc. With public transportation, technology, defense of the rapid development and awareness of precision measuring instruments growing, the demanding of ultra short optical pulses between narrow time interval and pulses high-precision measurement have become a remarkable research projects .TDC exclusive use of time-to-digital converter chip with Nios II soft-core processor time interval is a measurement system to achieve high precision measurement of the effective methods. The advantage is: High accuracy, debugging simple, low power consumption, real-time data processing, human-machine interface humanity, remote operation and owning embedded operating system. This work focuses on the following embedded Nios II soft-core processor and TDC–GP1 time-to-digital converter chip interface circuit and programming technology.

1. NIOSII Soft-Core Processor

A Nios II processor system is equivalent to a microcontroller or “computer on a chip” that includes a processor and a combination of peripherals and memory on a single chip. The term “Nios II processor system” refers to a Nios II processor core, a set of on-chip peripherals, onchip memory, and interfaces to off-chip memory, all implemented on a single Altera device. (Altera, 2005)Like a microcontroller family, all Nios II processor systems use a consistent instruction set and programming model.

1.1Nios II Processor System Basics

The Nios II processor is a general-purpose RISC processor core providing:

Full 32-bit instruction set, data path, and address space

32 general-purpose registers

32 external interrupt sources

Single-instruction 32×32 multiply and divide producing a 32-bit result

Dedicated instructions for computing 64-bit and 128-bit products of multiplication

Floating-point instructions for single-precision floating-point operations

Single-instruction barrel shifter

Access to a variety of on-chip peripherals, and interfaces to off-chip memories and peripherals

Hardware-assisted debug module enabling processor start, stop, step and trace under integrated development environment (IDE)control

Software development environment based on the GNU C/C++ tool chain and Eclipse IDE

Integration with Altera®'s Signal Tap® II logic analyzer, enabling realtime analysis of instructions and data along with other signals in the FPGA design

Instruction set architecture (ISA) compatible across all Nios II processor systems

Performance up to 250 DMIPS

Altera's SOPC Builder design tool fully automates the process of configuring processor features and generating a hardware design that you program into an FPGA. The SOPC Builder graphical user interface (GUI) enables you to configure Nios II processor systems with any number of peripherals and memory interfaces. You can create entire processor systems without performing any schematic or hardware description-language (HDL) design entry. SOPC Builder can also import HDL design files, providing an easy mechanism to integrate custom logic into a Nios II processor system. After system generation, you can download the design onto a board, and debug software executing on the board. To the software developer, the processor architecture of the design is set. Software development proceeds in the same manner as for traditional, on-configurable processors.

2 NiosII and TDC–GP1 Module Interface

2.1 TDC-GP1 chip brief Introduction

TDC - GP1 chip is a dual-channel Universal time-to-digital converter chip, with the largest range of 200ms and accuracy of 125 ps. Chip supports two working range and kinds of work patterns and it has flexible work practices. TDC-GP1 use 44 pin TQFP package with TDC measurement unit, 16-ALU, RLC measurement unit and eight processor interface modules. The pin names and functions are listed in Table 1(Acam, 2001)

Table 1. TDC-GP1 pin description

Symbol	Pin	I/O	Description	Symbol	Pin	I/O	Description
RST_N	1	I	Reset(low active)	VCC_CORE	5,28,44		Core supply voltage
TEST	2	I	Test pin, must be connected to GND	GND_CORE	6,27,43		Core Ground
CLK_REF	3	I	Input external reference clock	VCC_IO	17,35,39		Supply voltage IO ports
CHARGE	4	O	Charge pin for RLC measurements	GND_IO	12		Ground of IO Ports
SENSE	7	I	Schmitt trigger input RLC-measurements	GND_IO	18,40		Ground IO ports
RLC_P4	8	O	Port 4 for RLC-measurement	DATA0-DATA3	13-16	I/O	Data bus
RLC_P3	9	O	Port 3 for RLC-measurement	DATA4-DATA7	19-22	I/O	Data bus
RLC_P2	10	O	Port2 for RLC-measurement	ADR0-ADR4	23-26	I	Address bus
RLC_P1	11	O	Port 1 for RLC-measurement	INTFLAG	34	O	Interrupt flag (high active)
ALE	29	I	Address latch enable	START	36	I	Start input
RDN	30	I	Read (low active)	EN_STOP2	37	I	enable stop input 2

WEN	31	I	WRN Write (low active)	STOP2	38	I	Stop input 2
CSN	32	I	Chip select (low active)	STOP1	41	I	Stop 1 input
PHASE	33	O	Phase Phase out for regulation	EN_STOP1	42	I	enable stop input 1

Features:

2 measuring channels with a resolution of 250ps

4-fold multi-hit capability per channel, double pulse resolution 15ns, retriggerable

2 measurement ranges ---a: 2 ns -7.6 µs --- b: 60 ns-200 ms

The 8 events of the two channels can arbitrarily be measured against one another. Negative times can also be measured.

The resolution can be adjusted accurately via software in the 'resolution adjust' mode.

Ports to measure capacities, coils or resistors

Variable edge sensitivity of the measuring inputs

Internal ALU for the calibration of the measurement result. A 24-Bit multiplication unit enables the results to be scaled.

Wide range for the reference clock: 500 KHz - 35 MHz

Surface mount TQFP44 package.

Extremely low power consumption, fully battery operation possible

8-bit processor-interface

TDC-GPl provides with the external processor interface, including eight data buses, 4 bit can operate 16 Registers address line, read, write, film elections, and so on. Another, to simplify the interface design, also provided the address latches line (ALE). The internal chip has 7 only write registers, 4 can write value control registers, 8 only read results registers. External processor control to write control registers and value registers through the address line, writing, film elections, latches, etc. By controlling different addresses on the value of the control register, the chip can initialize, choose different working conditions and adjust the edge signal sensitivity. After measuring, the chip interruption activate the external processor to start reading the results register, the results are worth reading to further optimize the processor computation, storage and output display. Application of the range1 Register is set to: Reg0:0x44; Regl: 0x4D; Reg2:0x01; Reg3:0xXX: Reg4: OxXX; Reg5:0xXX; Reg6:Ox02; Reg7:0x01; Reg8:0x00; Reg9:0x00; Regl0:0x80.

2.2 NiosII and TDC-GPl interface circuit

Embedded Processor NiosII put TDC -GPl module interface equipment as PIO for the general operation. Therefore TDC data module |, address, the election unit, literacy signals are included in the PIO Bus. Its interface circuit shown in figure 1,which TDC_DB [7 .. 0] of the eight data lines, control will be responsible for order entry to the control registers and results data from the results registers were read out; TDC_AD [3 .. 0] to address data line; TDC_ALE to address latches; TDC_CEN signal for the election unit; TDC_WRN to enable writing; TDC_RDN to enable reading; INTFLAG signs of disruption; START signal to begin; STOP 1 received no signal delay; STOP 2 received signal delay.

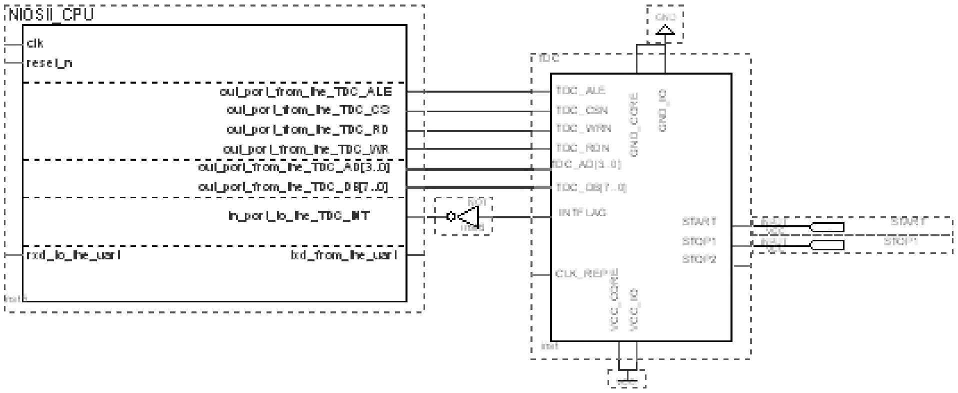

Figure 1. Interface between the Embedded Processor NiosⅡ and theTDC Module

3. NiosII and TDC interface chip

Nios II processor uses modular structure software, including;(1)TDC-GPl initialization,(2) setting TDC-GPl chip channel,(3) working mode setting, (4)measurement the results,(5)measurement data display,(6)measurement data storage. For example, TDC - GPl can work in precision adjustable model. After the measurement, the results will be shown as the following a series of manipulations:(1) TDC chip be set in the state of writing through the assignment Control Register 7 shielding all of the input signal STOP.(2) Write control register 11, initialize the TDC and ALU.(3) Pointer at the control register 0, the chip work in a range 1, self-calibration mode characterized by a number of functions. Then write control register 1, the chips achieve precision adjustable, adjustable work on the accuracy of mode l.(4) Write control register 2.The result is that, The first pulse of channel two rise time and the access an article a pulse rise time for the poor.(5) Write control register 7, the abolition of the two channels on the STOP input signal shielding. Read measurements, circuit is SCM inquiries TDC - GPl chip interrupted output pins, chip interrupted when issued, that is, enter the break in service procedures in the home TDC chip in the reading of the state, NiosII processor visits the results register and read the results. (Liu, 2004).Then niosII processor computate and show the results, the following is the main part of the NiosII interface procedures.

3.1 Initialization TDC–GPl

As Embedded Processor NioslI put TDC–GPl module interface as ordinary operate equipment PIO; blocks all the data through IOWR_ALTERA_AVALON_PIO_DIRECTIO N (PIO base address, data) order to register for the PIO write, IORD_ALTERA_AVALON_PIO_DATA (PIO base address) orders reading operation. Below is the initialization procedures:

```
Void Initial_TDC(   )                              // Initialization
        {IOWR_ALTERA_AVALON_PIO_DATA (TDC_ALE_BASE, 1); // ALE write for 1
IOWR_ALTERA_AVALON_PIO_DATA (TDC_CS_BASE, 0);           // Chip selected TDC
IOWR_ALTERA_AVALON_PIO_DATA (TDC_WR_BASE, 0); //Write effective signal
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x7); //Write control register 7
IOWR_ALTERA_AVALON_PIO_DATA (TDC_DB_BASE, 0x00); //Shield all stop importation
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0xb); // Write register 11
IOWR_ALTERA_AVALON_PIO_DATA (TDC_DB_BASE, 0x07); // Initialization of TDC and ALU
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x0); // Write register 0
IOWR_ALTERA_AVALON_PIO_DATA (TDC_DB_BASE, 0x44); // Choice range 1 to work
                                                   // Both Automatic Calibration
                                            Mode and Multiply fuction
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x3)       // Write register 3
IOWR_ALTERA_AVALON_PIO_DATA(TDC_DB_BASE,0x64)    // DIV_CLK_PLL=100
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x4)       // Write register 4
IOWR_ALTERA_AVALON_PIO_DATA(TDC_DB_BASE,0x16)
                                                 // SEL_CLK_PLL =64NEG_PH_PLL
                                                           // => Res = 50ns*64/120/100 = 1/3.75 ns
          Float resolution = 1.0/3.75;
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x1); // Write register 1
IOWR_ALTERA_AVALON_PIO_DATA (TDC_DB_BASE, 0xCD); // Work in the precision adjustable mode
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x2); // Write register 2
IOWR_ALTERA_AVALON_PIO_DATA (TDC_DB_BASE, 0x19); //
                                            Channel 2 first rising edge – Channe1 1 first rising
edge                                                                                          }
//Initialization finished
```

3.2 Measurement showed unit

```
     Void measure_TDC ()
        {Int valid = 0;
```

```
    Float nk0, nk1;
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x2);      // Write register 7
IOWR_ALTERA_AVALON_PIO_DATA (TDC_DB_BASE, 0x09); // open Stop1 and Stop2
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0xb); // Write register 11
IOWR_ALTERA_AVALON_PIO_DATA (TDC_DB_BASE, 0x07); // Initialization of TDC and ALU
                                                          // Start measuring signal
      While (valid==0)
        {
Valid=IORD_ALTERA_AVALON_PIO_DATA (TDC_INT_BASE); //Inquiries interrupted signs spaces
           }
        IOWR_ALTERA_AVALON_PIO_DATA (TDC_WR_BASE, 1);      // Write enable
        IOWR_ALTERA_AVALON_PIO_DATA (TDC_WR_BASE, 1);      // Read enable
IOWR_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x00); // Read addresses enable 0
        nk0= IORD_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x00); // Read low results
nk1= IORD_ALTERA_AVALON_PIO_DATA (TDC_AD_BASE, 0x00);// Read high results
         nk1 = nk1 * 256;                                   // Backspace
         nk1 = nk1 | nk0;                                 //Merger
result = resolution * (float)(nk1);                // Calculate the final results
printf("result = %6.3fns\n",result);         // Standard output results displayed on the LCD
                                         }
```

3.3 Data Storage Processing Unit

```
      Void Save_result( float result )
        {
           alt_flash_fd* fd;
           int ret_code;
           Float TDC_SAVE [100];
TDC_SAVE[k] =result;                              // K is the storage location counter
fd = alt_flash_open_dev("/dev/ext_flash");       // Open FLASH memory

        If (fd)
            { printf("open flash device successfully\n");
// LCD Display as standard output

             ret_code = alt_write_flash(fd, k*32,TDC_SAVE, 32);
            // storage data to the FLASH device
             If (!ret_code)
{ printf("save result on %d\n",k\n");}                // Show storage location
               alt_flash_close_dev (fd);
               printf("save result successfully\n");
               // close the FLASH
               }
        Else
          {printf("Cant open flash device\n");
               }
```

```
                    Return 0;
                            }
```

4. Conclusion

The work uses NiosII soft-core processor to design TDC-GPl time-to-digital converter chip interface circuits and interfaces driver. As NiosII soft-core processor design can use a lot of nuclear molding IP, and Nios II processors can link the external memory, LCD display, ethernet controller and USB, peripherals for the coordination and sharing of data. So we therefore based on the Nios II TDC module interface flexibility in the configuration of any treatment based on the Nios II Components of the products design, with the advantages of flexible and easy, low power consumption, simply debugging, testing of high precision and accuracy, real-time data processing, human-machine interface humanized, remote operation and embedded operating systems, etc.

References

Liu, Guofu and Liu, Bo. (2004). TDC-GP1 time interval measurement precision chip and its Application. *Microcontroller and Embedded Systems.* 11. 38-40.

Nios II Processor Reference Handbook. (2005). Altera Corp.

TDC-GP1 fuctional description Handbook. (2001). Acam Corp.

Nios II Software Developer’s Handbook. (2005). Altera Corp.