Second Prize

Laser Direct Writing Digital Servo Controller Based on SOPC Technology

Institution: **Ultra-Precision Photoelectric Instrument Engineering Research Institute, Harbin Institute of Technology**

Participants: **Lei Yan, Tao Cheng, Ya Gao**

Instructor: **Wang Lei**

Design Introduction

The application of precision/ultra-precision processing technology has expanded from a few areas (such as national defense and aerospace) to different aspects of the national economy. As the semiconductor industry develops, it imposes increasing requirements on laser direct writing lithography, which is the core of very-large-scale integration (VLSI) manufacturing.

Laser direct writing lithography applies a flexible dose exposure on a substrate surface with a variable intensity laser beam. A key step of this process is high-precision scanning with a computer-controlled laser beam. During lithography, the substrate on the stage moves with the platform and is exposed with a flexible dose by controlling the laser beam intensity with an acousto-optic modulator. The stage's positioning accuracy and motion stability directly affect the performance of the lithography machine and the quality of the lithographic components. Therefore, a rapid, high-precision straight-line feed system is needed.

The voice coil motor (VCM) has high displacement resolution, zero-length feed drive chain (i.e., direct drive or zero drive), strong dynamic sensitivity, and good responsiveness. Therefore, the VCM has replaced the traditional positioning structure of the rotating servo motor plus ball screw shaft, becoming the motion servo core for micron or even submicron location of the laser direct writing stage.

The digital servo-based motion controller is the key to an ultra-precision positioning system. Digital servo means that digital technology is used for closed-loop control and system regulation. Additionally, the control and regulation are based on software, so that the pulse width modulator (PWM) control signal is exported directly. Alternatively, a digital-to-analog converter (DAC) chip generates the DC drive current and the VCM is driven after power amplification of the motor driver. Software-based regulators allow a variety of controls, such as vector control, parametric adaptive control, sliding mode variable structure control, fuzzy control, neuron network control, etc. Software eases parameter self-optimization and fault self-diagnosis functions, and improves system control performance. It also overcomes the shortcomings of simulated closed-loop servo systems, including difficulty with weak signals, noise separation for weak signals, difficulty in improving control precision to above 0.1%, vulnerability to temperature, zero drift error of the location control, etc.

To curb noise interference (including gas film disturbances in movement of the laser direct writing stage) and to implement fast submicron location, this paper describes a VCM digital servo motion controller based on Altera's Nios® II embedded soft-core processor. The design uses the low-cost, high-performance Cyclone® II FPGA family and an integrated dual Nios II soft-core application system for various functions, including fast sampling and high-resolution decoding of signals from a displacement sensor (laser interferometer), motion state monitoring of a controlled object (static pressure air-bearing slider and VCM), motion characteristics spectrum analysis and digital filtering, PWM control signal or DAC DC voltage output based on the integration split packet identifier (PID) control algorithm, etc. With highly integrated system-on-a-programmable-chip (SOPC) technology, the design features flexible functions, reduced electromagnetic interference, improved processing speed and control reliability, reduced development costs, and easier system upgrades and maintenance.

Function Description

In our laser direct writing motion control system, the dual Nios II-based digital servo motion control reduces noise caused by load disturbances of the motor rotation, gas film disturbances of the air-bearing slider, and mechanical resonance of the drive system. Our closed-loop servo system has four parts:

- *Controlled object*—The controlled object is the stage operating on a static-pressure, air-bearing slider, and is driven by the VCM.

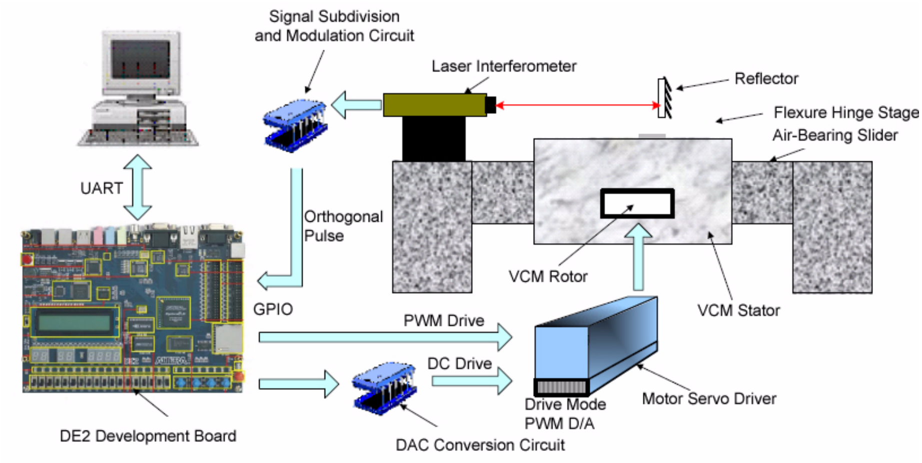
- *Displacement sensor (feedback)*—The sensor is Renishaw's laser interferometer, which, based on principles of interferometric measurement, transforms the stage's displacement information into a corresponding orthogonal pulse signal and implements a logic-level conversion of the pulse signal through a signal modulation circuit.

- *Control unit*—The control unit is the Altera® Development and Education (DE2) development board that integrates dual Nios II embedded soft-core processors. It performs several tasks, including signal acquisition, digital filtering, a PID control algorithm, a DAC chip interface, a PWM control quantity output, and user interface based on the µC/OS-II real-time operating system (RTOS).

- *Actuator*—The actuator is a motor servo driver with two operating modes: DC voltage driving and PWM driving. The system can export the control algorithm result through a DAC interface module or custom PWM peripheral module, which controls the motor servo driver in different ways.

Figure 1 shows the structure of the digital servo system.

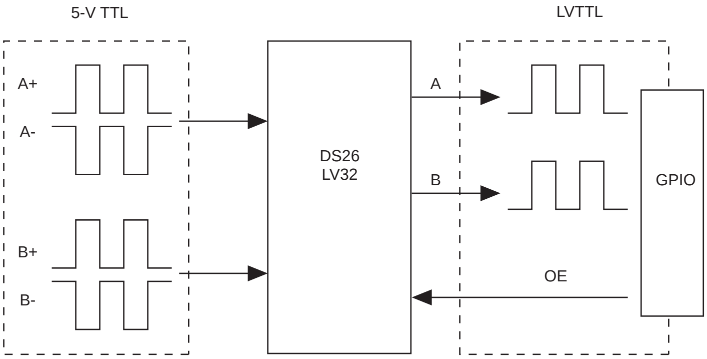
Figure 1. Digital Servo System Structure



Signal Modulation Circuit

The laser interferometer exports four types of differential signals, including A+, A-, B+, and B-, with an orthogonal phase relationship and a 5-V TTL logic level. We used the 10-MHz DS26LV32 differential receiver to convert the differential signals into single-ended signals and provide level conversion, acquiring LVTTL orthogonal pulses A and B for general-purpose I/O (GPIO) sampling of the control unit. It can also control the chip enable to prevent sampling errors. Figure 2 shows a schematic diagram of the signal modulation circuit.

Figure 2. Signal Modulation Circuit



Pulse Signal Subdivision Decoding Unit

The pulse signal subdivision decoding unit consists of the subdivision sense module and the M/T decoding module. The subdivision sense module generates a standard-width spike pulse at the rising and falling edge of the orthogonal pulses A and B, performs logical OR operations, obtains a quadruple subdivision pulse (`pulse`), obtains a system resolution of 80 nm/pulse by calculating the laser interferometer resolution indexes, and calculates the motor's motion direction (`direc`) based on the phase relationship (`advance` or `lag`). The standard frequency signal `CLK_100` provides a latency base value to prevent post-processing counting errors due to narrow spike pulses. Figure 3 shows a schematic diagram of the top-level module.

Figure 3. Pulse Subdivision Sense Top-Level Module

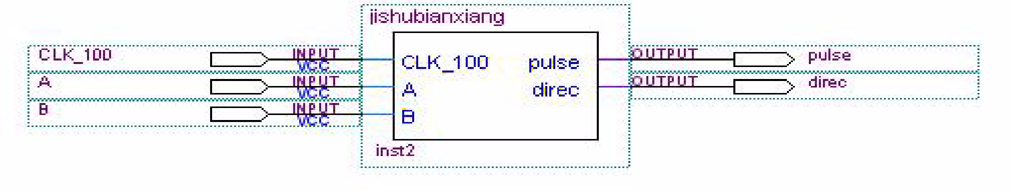
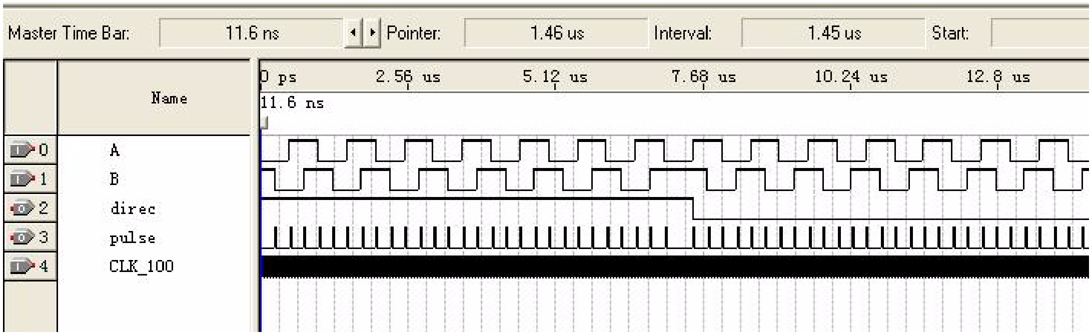


Figure 4 shows the simulation result of the subdivision sense module functions. `CLK_100` is a 100-MHz standard frequency signal.

Figure 4. Pulse Subdivision Sense Module Simulation Result


Master Time Bar: 11.6 ns
Pointer: 1.46 us
Interval: 1.45 us
Start:

The M/T decoding module processes quadruple-subdivision signals while acquiring the motor movement's relative position and speed, placing the motor under the control of a position loop or speed loop plus dual-position loop. The system can acquire the relative motor position (`position`) by adding and subtracting `pulse` by `direc`. The system synchronously presets gate signal `T1` by the tested signal to acquire the actual gate signal `T2`. It counts the standard frequency (`CLK_50`) and `pulse` simultaneously in `T2` to acquire count values N_C and N_P. The quadruple subdivision pulse frequency $f_P = N_P f_C/N_C$, and the motor speed can be acquired according to the motor displacement corresponding to each pulse of f_P. We used a synchronous gate to ensure integral counting multiples and implement an equal precision measurement within the measuring scope of the signals. Simultaneously counting the tested frequency and standard frequency allows the system to switch the cycle and frequency measurements automatically according to the motor's speed change and ensures a wide band for speed measurement. Figure 5 shows the schematic diagram of the top-level module.

Figure 5. M/T Decoding Top-Level Module

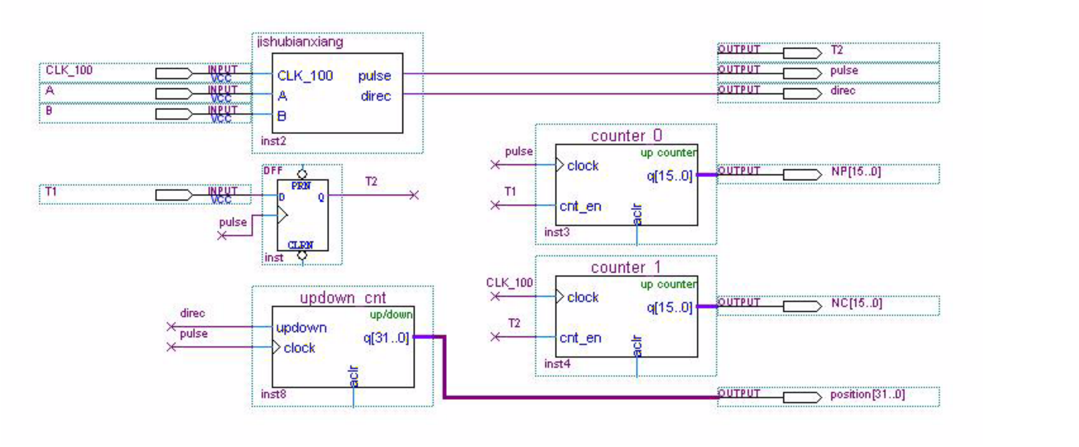
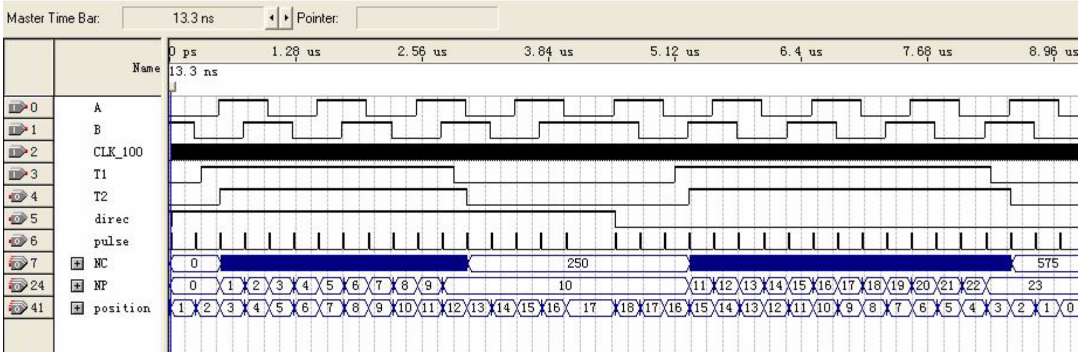


Figure 6 shows the M/T decoding module simulation result. CLK_100 is the 100-MHz standard frequency signal, T1 is the preset gate, T2 is the synchronous gate, and NC and NP are the standard frequency and count result of quadruple subdivision frequency, respectively.

Figure 6. M/T Decoding Module Simulation Result



Data Processing Unit

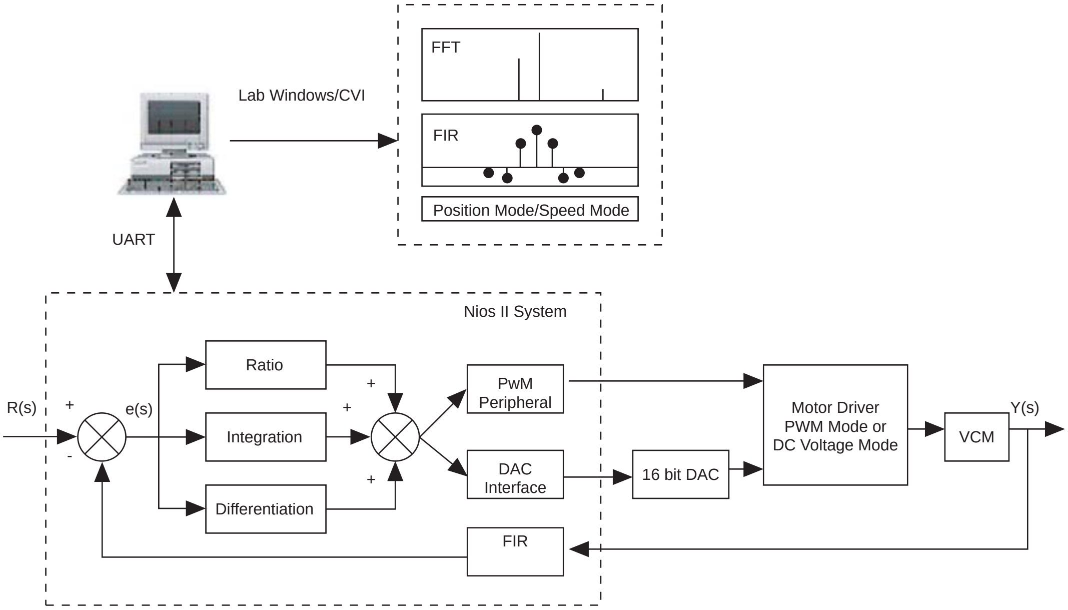
For the data processing unit, the core of the closed-loop control system consists of a Nios II hardware platform (including the Nios II embedded soft-core processor, memory, on-chip peripherals, custom peripherals, etc), as well as the supporting LabWinsows/CVI software on the PC side:

- *Nios II hardware platform*—We used a dual-core Nios II processor and updated parameters such as control quantity, PID, and finite impulse response (FIR) coefficients in real time via communications between the serial port and the high-level computer software. The system performs an efficient real-time FIR filter operation using a custom multiply-add instruction, collects sensor signals to integrate a split PID operation, and drives a custom PWM or DAC module to control the VCM servo driver operation. Figure 7 shows the data processing unit structure.

- *LabWindows/CVI platform*—This platform sets the PID control parameter to control the location and size. It uploads the follow-up signal position collected by the Nios II hardware platform through the serial port, performs a fast Fourier transform (FFT) operation, and draws a delay/frequency response curve. It sets the digital FIR filter parameters (such as filter type, order, quantification coefficient, and window function) according to the FFT analysis result.

Additionally, it sets excitation signals (e.g., sine, square wave, and step) and observes the follow-up curve position after filtering.

Figure 7. Data Processing Unit Structure



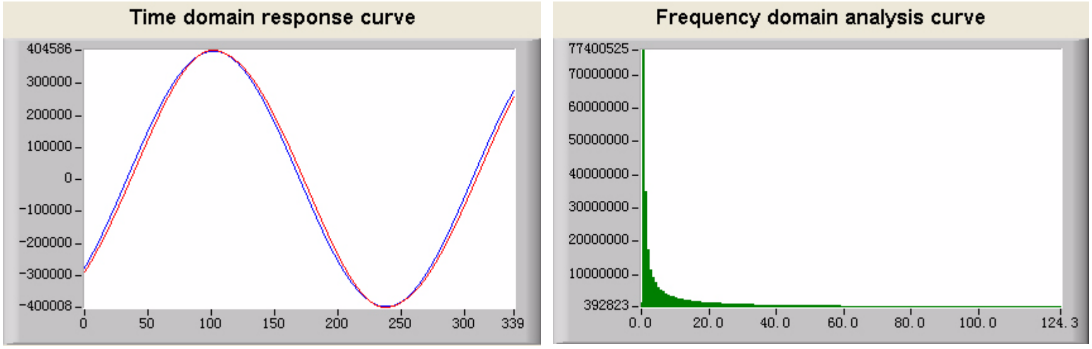
FFT

The discrete Fourier transform (DFT) of N* sample points is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad W_N = e^{-j\frac{2\pi}{N}}, \quad k = 0, 1, \cdots, N-1$$

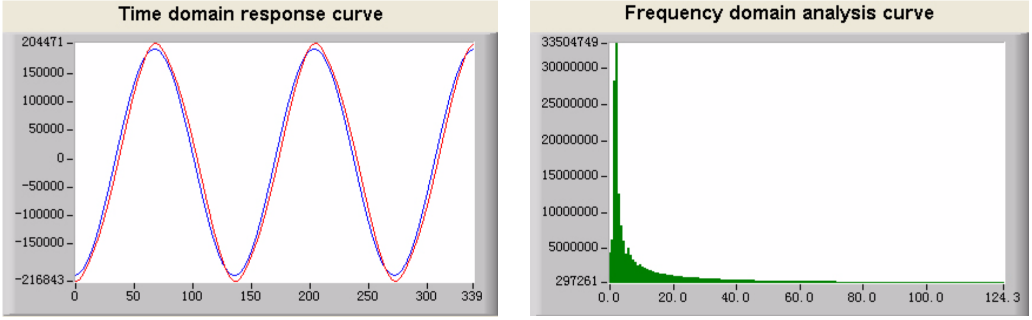
FFT, as an efficient implementation of DFT, delivers high operation efficiency and is suitable for real-time digital signal processing (DSP). Because the noise signal frequencies caused by gas film disturbances, mechanical resonance, etc. are relatively stable, we used post-processing and a CVI program for FFT operation on the Nios II-collected feedback signals. While ensuring the spectrum analysis result reliability, this method prevents real-time FFT operation with a Nios II soft-core processor and greatly reduces the system's operation load. We tested the system frequency response using sine wave excitation with a pulse amplitude of 400,000, a frequency of 1 Hz, a sampling frequency of fs = 250 Hz, and N = 340 sampling points. Figure 8 shows the position-time domain response curve and the acquired frequency analysis curve.

Figure 8. Time and Frequency Domain Response Curves with 1-Hz Sine Excitation



We tested the system's frequency response using sine excitation with a 200,000 pulse amplitude and 2-Hz frequency. The sampling frequency and the number of sampling points remain unchanged. See Figure 9.

Figure 9. Time and Frequency Domain Response Curves with 2-Hz Sine Excitation



According to the frequency response curve, noise interference exists at frequencies of 5 to 60 Hz and is especially significant around 5 Hz, which is quite similar to the gas film disturbance frequency in theoretical analysis. Other noise may come from mechanical resonance, electromagnetic interference, etc.

Digital FIR Filter

FIR filters are widely used in digital system design and can acquire a strict linear phase while ensuring amplitude characteristics and satisfying technical requirements. The system function H(z) of an N-order FIR filter is:

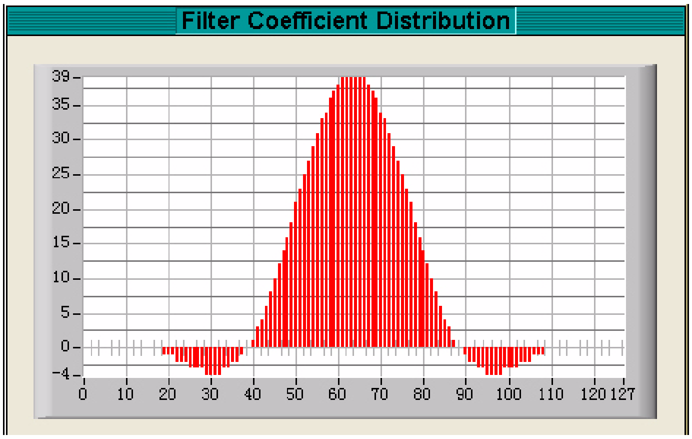
$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

In a digital system, the output and input functions $y[n]$ and $x[n]$ are:

$$y[n] = \sum_{k=0}^{N} a(k) x[n-k]$$

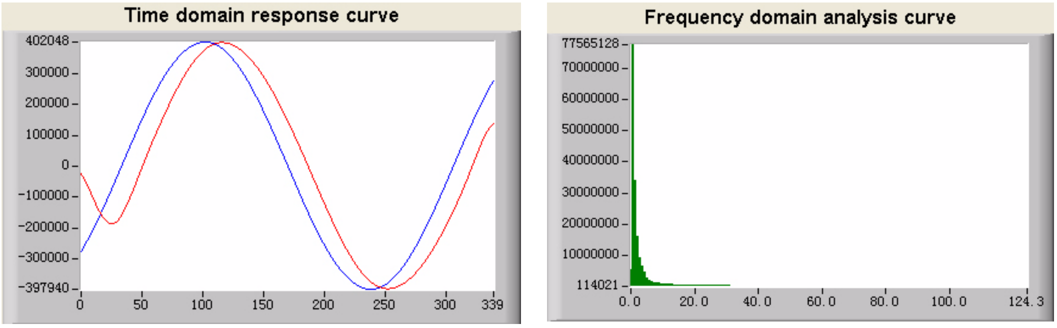
In LabWindows/CVI, we set a 128-order, low-pass FIR filter with a 2-Hz cutoff frequency, a 100-Hz sampling frequency, a Hanning window, and a 1,000 quantification coefficient. Figure 10 shows the coefficient distribution curve.

Figure 10. 2-Hz Low-Pass FIR Filter Coefficient Distribution Curve



We downloaded the filter coefficient torque to the Nios II hardware platform through the serial port. Input $x[n]$ and coefficient $a(k)$ are multiplied and added by a custom multiply-add instruction, implementing real-time digital filtering of the input signals. Figure 11 shows the time and frequency response curves for sine excitation with a 400,000 pulse amplitude and 1-Hz frequency.

Figure 11. Time and Frequency Domain Response Curves after Filtering for 1-Hz Sine Excitation



In LabWindows/CVI, we set a 100-order, low-pass FIR filter with a 5-Hz cutoff frequency, a 100-Hz sampling frequency, a Blackman window, and a quantification coefficient of 1,000. Figure 12 shows the coefficient distribution curve.

Figure 12. 5-Hz Low-Pass FIR Filter Coefficient Distribution Curve

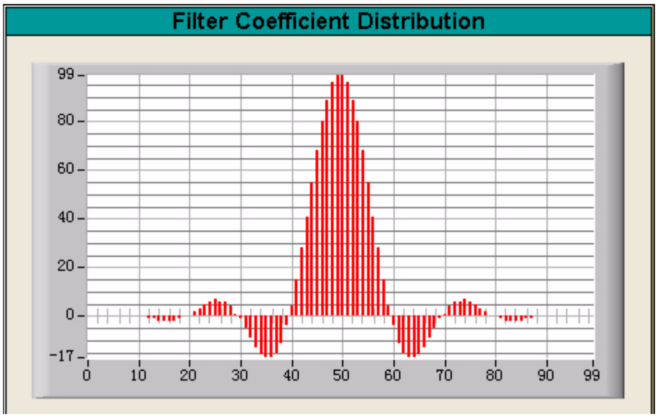
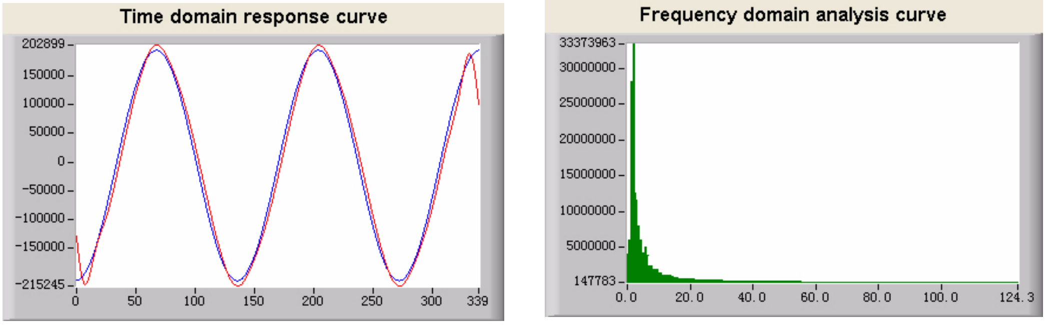


Figure 13 shows the time and frequency response curves for sine excitation with a 200,000 pulse amplitude and 2-Hz frequency.

Figure 13. Time and Frequency Domain Response Curves after Filtering for 2-Hz Sine Excitation



According to our analysis, noise in the feedback signals is curbed after we added the digital FIR filter. The position response curve becomes smooth but a linear phase shift (directly related to the filter order) occurs. In this project, we use a custom instruction for the filtering algorithm. The operation time for a single order is controlled to within 10 clock cycles (or 100 ns) and the total latency for a 128-order operation is about 13 µs, which is negligible compared to the 200-µs servo cycle. Therefore, the steady state error influence is within the specified scope. We can reduce the latency by increasing the system clock frequency or reducing filter orders. Additionally, we can reduce the phase delay error significantly by introducing phase compensation, further improving precision control.

Integrated Split PID Control Algorithm

In ordinary PID control, the proportional element reflects the control system’s deviation signals proportionally and the controller performs control functions immediately when the deviation occurs to reduce the deviation. The integrating element mainly eliminates static error and improves the system’s astatism. The derivative element reflects the variation rate of the deviation signals and introduces a valid early amendment signal to accelerate system operation and shorten regulation time. The main purpose of the integrating element is to eliminate static error and improve control precision. However, at the beginning, end, or substantial change of a setting, there is a large system output deviation for a short time, leading to an integrated PID operation, control quantity exceeding the control quantity limits corresponding to the maximum actuating range allowed by the actuator, significant system overshoot, or even oscillation.

To prevent this situation, we used an integrated split PID algorithm in this design. The integration function is cancelled when there is a large deviation between the controlled quantity and the set value to prevent system instability and increased overshoot caused by integration functions. When the controlled quantity is about the same as the set value, the system introduces an integration function to eliminate static error and improve precision control. See the following steps:

1. Set a threshold $\varepsilon > 0$ according to the actual situation.

2. When $|error(k)| > \varepsilon$, PD control prevents excessive overshoot and ensures quick system response.

3. When $|error(k)| \leq \varepsilon$, PID control ensures the system's control precision.

The integration split control algorithm can be expressed as:

$$u(k) = K_P E(k) + \beta K_I \sum_{j=0}^{k} E(j)T + K_D[E(k) - E(k-1)]/T$$

Of which, K_P is the proportionality coefficient, K_I is the integration coefficient, K_D is the differential coefficient, T is the sampling time, and β is the integration switching coefficient. See the following equation:

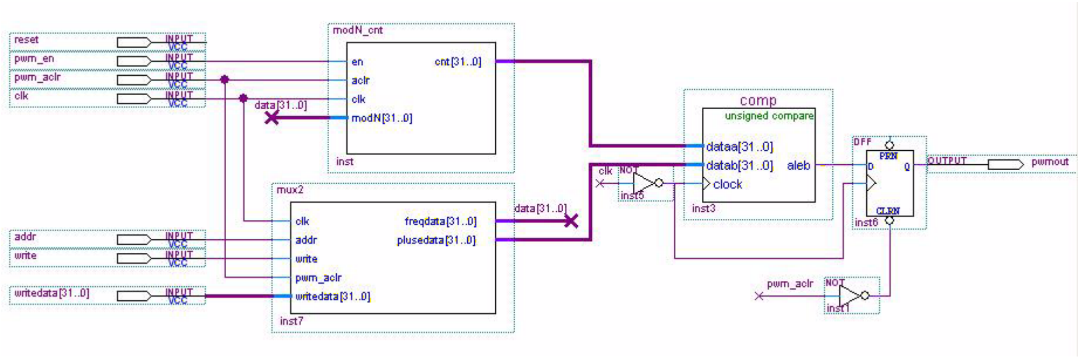
$$\beta = \begin{cases} 1 & |error(k)| \leq \varepsilon \\ 0 & |error(k)| > \varepsilon \end{cases}$$

Custom Subordinate PWM Peripheral

To adapt to the motor servo driver's PWM operating mode and to convert PID operation results into control quantity signals, this design uses a customized Avalon® switch fabric-based PWM subordinate peripheral with an adjustable output frequency and duty ratio set according to the PWM waveform generation principle. The subordinate peripheral provides linear conversion from the operation result to the drive voltage, and the theoretical quantizer ratio is $1/2^{32}$.

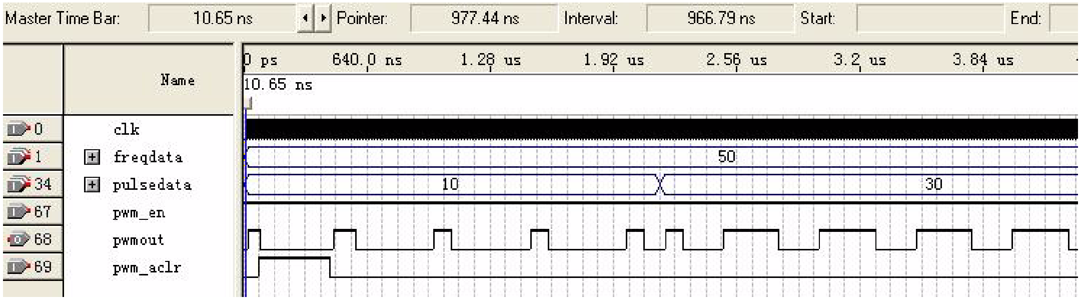
The PWM module consists of a bus interface controller, a 2-bit frequency register, a 32-bit duty ratio register, and a comparator. Figure 14 shows the top-level module.

Figure 14. PWM Top-Level Module



Ignoring the Avalon bus control signals, the PWM module simulation result is shown in Figure 15. `clk` is the 100-MHz reference clock, the square wave output frequency (`freqdata`) is 50, the set duty ratio (`pulsedata`) goes from 10 to 30, `pwm_en` is the module enable signal, `pwm_aclr` is the asynchronous clear signal, and `pwm_out` is the output. The simulation result shows that the module adjusts the duty ratio from 10/(50 + 1) to 30/(50 + 1).

Figure 15. PWM Module Simulation Result

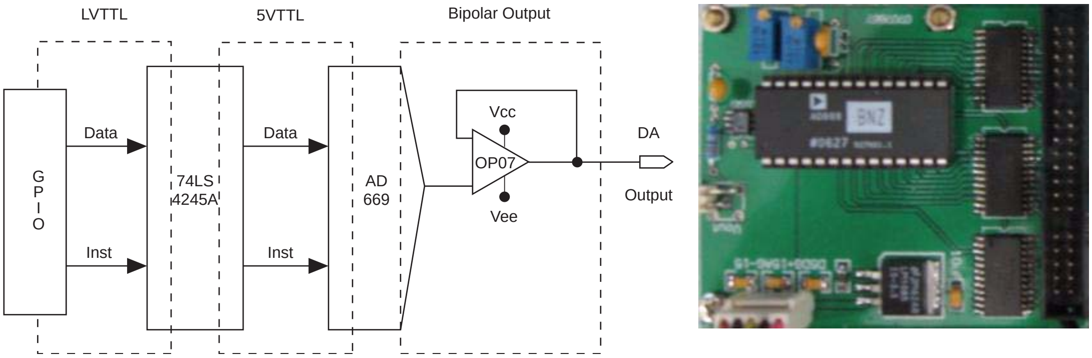


Master Time Bar: 10.65 ns Pointer: 977.44 ns Interval: 966.79 ns Start: End:

DAC Drive Unit

To adapt to the motor servo driver’s DC voltage operating mode, we designed a AD669-based DAC drive circuit for functions such as level matching of the Nios II processing unit control signals, DAC control output, signal modulation, etc. The AD669 device is a high-performance, parallel DAC chip produced by Analog Devices, Inc. It features a bipolar voltage output (voltage between -10 and +10 V), 16-bit conversion precision, nonlinear error ≤0.003%, and an output setting time of ≤13 μs. The Nios II system writes the operation result into the DAC through the general-purpose I/O (GPIO) and controls the pin startup and conversion through the LDAC, DACS, etc. In the circuit, the 74LS4245A device performs level matching and the OP07 emitter provides impedance transformation with the circuit. Figure 16 shows the DAC drive unit structure and a photo of the circuit board.

Figure 16. DAC Drive Unit Structure and Photo of Circuit Board



Performance Parameters

The high-precision VCM control system should have quick response, small overshoot, and a stable rate. Therefore, besides performance benchmarks, we must consider the relationship between parameters when designing the control system. The system’s main technical indicators are:

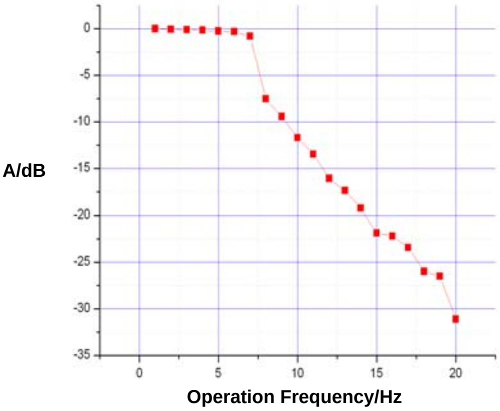
- Bandwidth: 10 Hz
- Maximum travel of VCM: 300 mm
- Minimum resolution: 80 nm

- Maximum tracking speed: 400 mm/second
- Steady state error: ≤3%

System Bandwidth

The system bandwidth reflects the control system's reproduction of useful signals and suppression of useless signals. The control system must have a quick response to curb the dynamic error after disturbance and fast attenuation to reduce interference, so we proposed a bandwidth requirement. Because the control system model is unknown, signal excitation is reproduced to acquire the actual system bandwidth. In this paper, we used a sine signal with a 300-mm amplitude (placed at the maximum travel of the system) on the controlled object and gradually increased the excitation signal's frequency to obtain the amplitude frequency characteristics curve. See Figure 17.

Figure 17. System Amplitude Frequency Characteristics Curve



According to the test result, the system responds quickly when the excitation signal frequency is around 8 Hz, but the position falls significantly as the frequency increases. Therefore, the system bandwidth for our project meets the 10-Hz design requirement.

Minimum Resolution and Maximum Tracking Speed

VCM features quick response and high tracking speed. The maximum tracking speed is mainly restricted by the output frequency of the laser interferometer; the system resolution and maximum tracking speed parameters are contradictory. According to performance parameters of the laser interferometer, the laser interferometer resolution is set as 80 nm and the tracking speed upper limit is 400 mm/second.

Steady State Error

We used step signals with different amplitudes for excitation to acquire the system position response and analyzed the test result to obtain the design's PID parameter. The system controls the steady state error to within 3% of the total amplitude.

Design Architecture

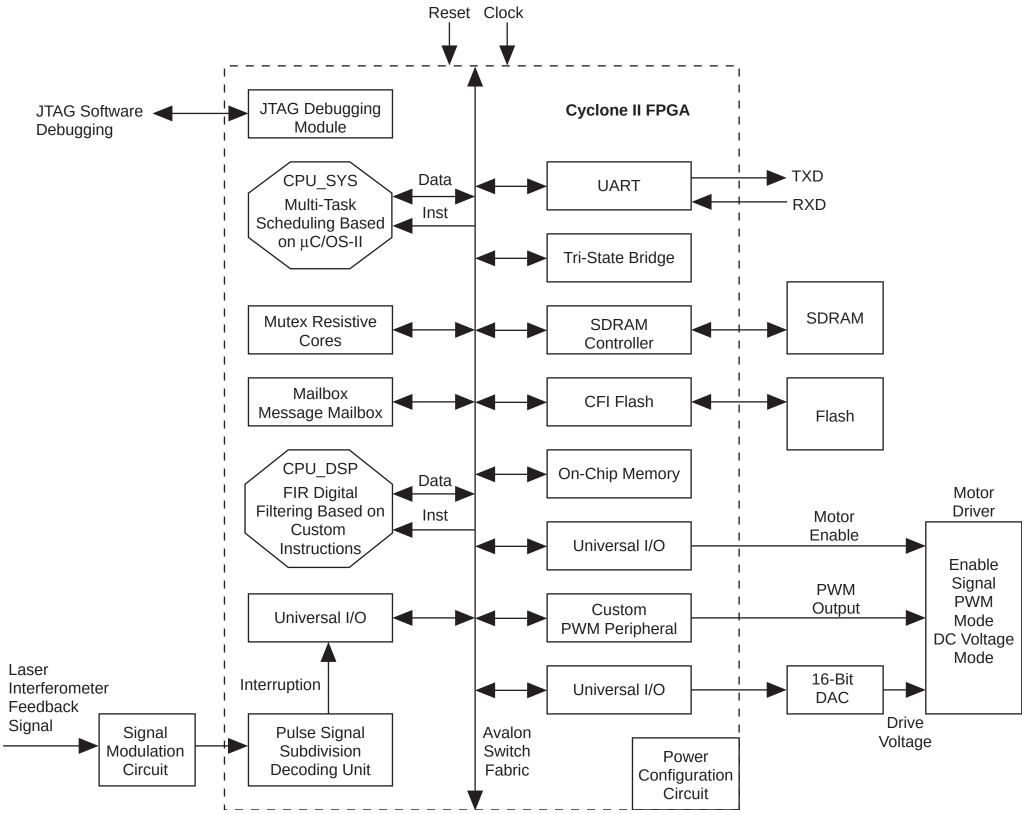
This section describes the design architecture.

System Hardware Design Diagram

The hardware structure of the SOPC-based laser direct writing digital servo controller comprises the DE2 development board, sensor signal modulation circuit, DAC drive circuit, etc. The core control unit (the DE2 development board) consists of a Cyclone EP2C35 FPGA, SDRAM, flash memory, UART interface, USB-Blaster download interface, clock, configuration circuit, power, etc. The EP2C35 FPGA integrates the dual Nios II processors, CPU_SYS and CPU_DSP, based on the Avalon switch fabric bus. To ensure correct access and modification of shared resources, we used Mutex resistive cores to restrict access rights. Data exchange between the two cores is implemented using MailBox (inbox and outbox).

To enable serial port communication, SDRAM, flash chip access, DAC control, etc, we configure several additional components, including an Avalon bus-based UART core, tri-state bus, SDRAM controller, CFI flash controller, M4K on-chip RAM, and universal I/O. Additionally, we customized the PWM subordinate peripheral to acquire programmable PWM pulse signals and allow the motor servo driver to operate in PWM drive mode. Figure 18 shows the system hardware block diagram.

Figure 18. System Hardware Block Diagram



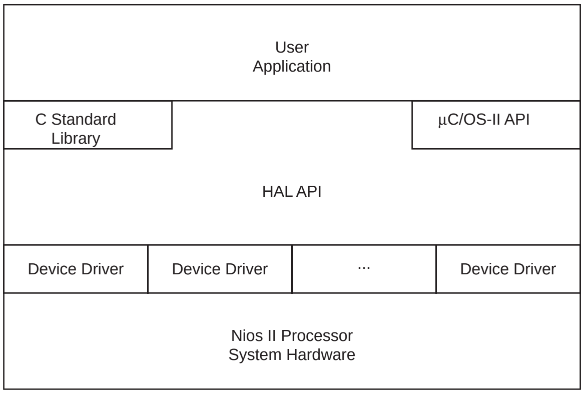
System Software

This system integrates the CPU_SYS and CPU_DSP dual Nios II soft-core processors based on the Avalon switch fabric bus. CPU_SYS is responsible for position and speed signal conversion, integration of the split PID operation, UART communication, PWM and DAC peripheral output control, etc., and is dispatched and managed by the µC/OS-II embedded RTOS. CPU_DSP is responsible for acquiring the sensor decoding signals, providing digital FIR filtering, etc.

CPU_SYS Implementation

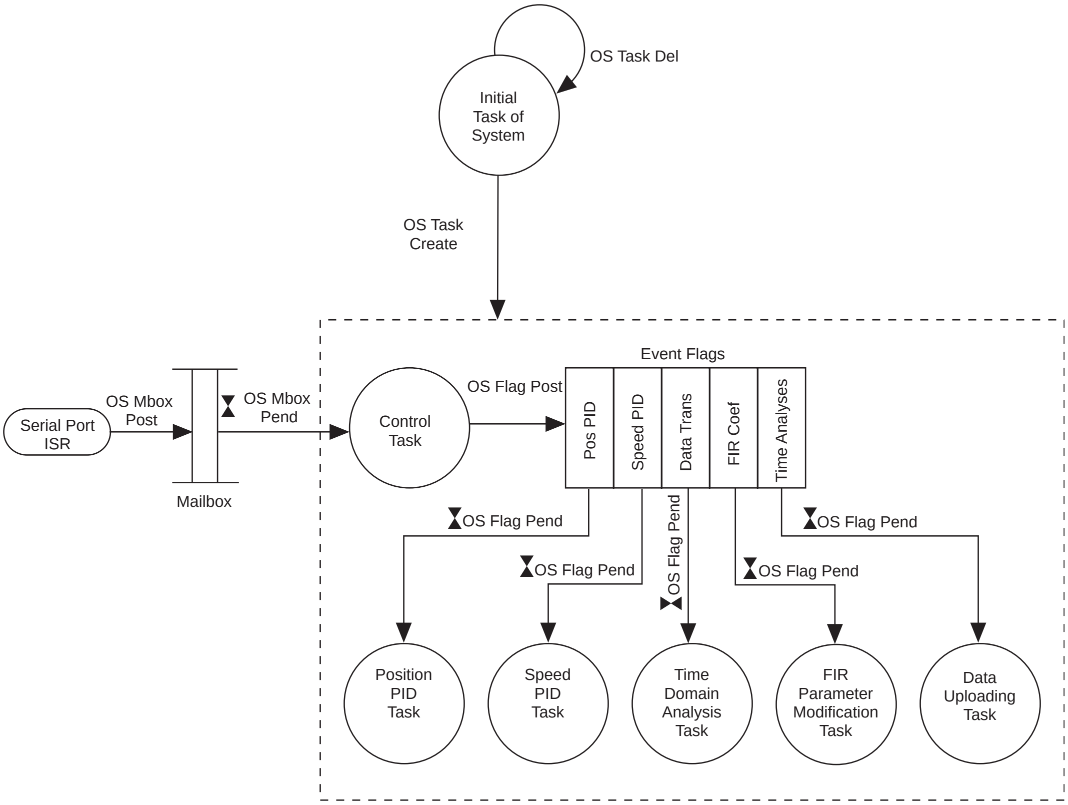
µC/OS-II is a portable, scalable, preemptive, real-time multi-tasking kernel compliant with the Radio Technical Commission for Aeronautics (RTCA) DO-178B level requirements adopted by the United States Federal Aviation Administration (FAA). It has high stability and security. Migration of µC/OS-II to the Nios II processor is based on the hardware access layer (HAL): the programs are not sensitive to low-level hardware changes and we can invoke the HAL API function. Figure 19 shows the µC/OS-II program structure.

Figure 19. µC/OS-II Program Structure



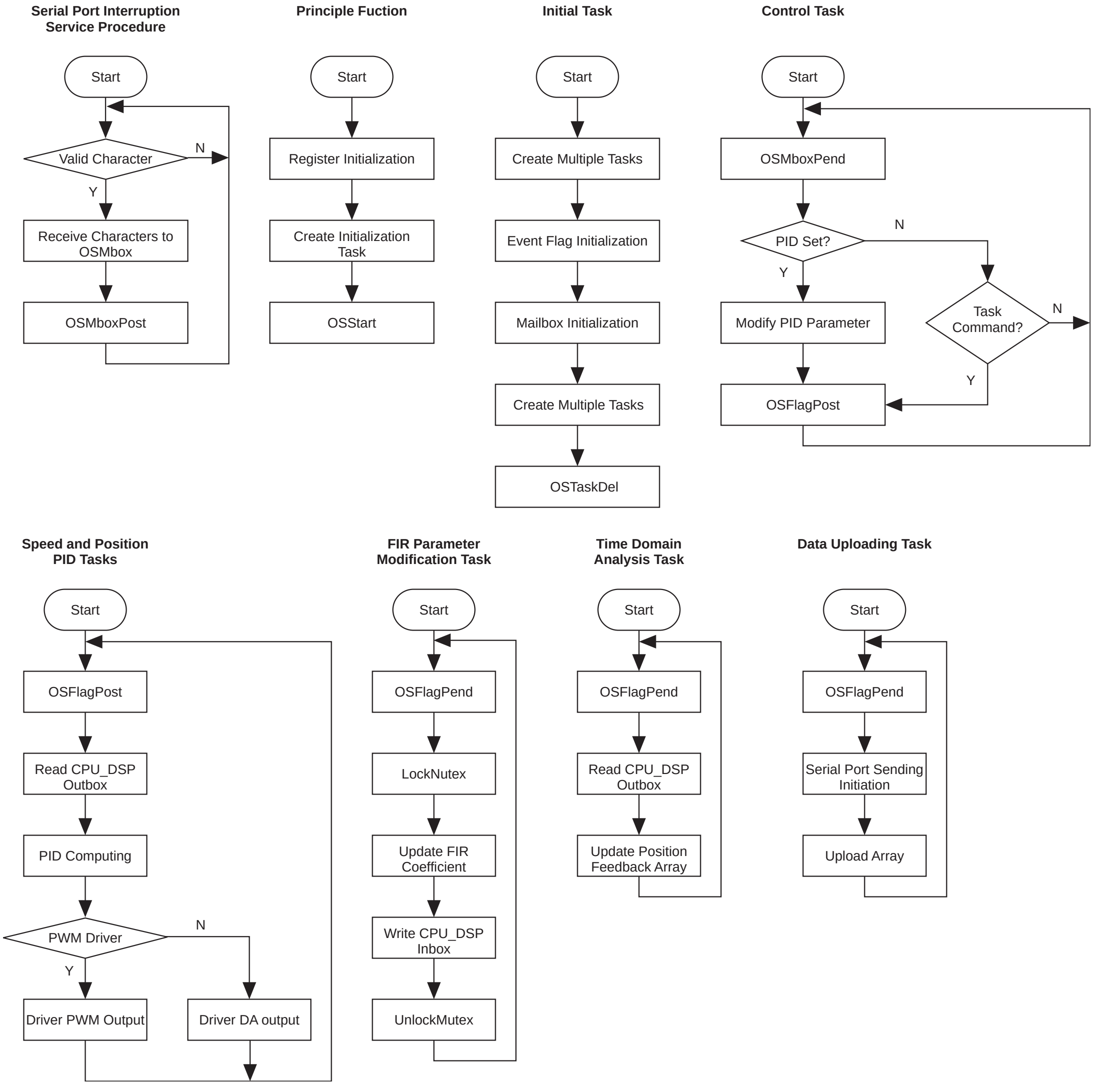
CPU_SYS, the central processing soft core, is based on the µC/OS-II multi-tasking environment. Figure 20 shows the state transition structure.

Figure 20. Task State Transition Structure



The task sequence, according to priority, is: system initial task, control task, position PID task, speed PID task, time domain analysis task, FIR parameter modification task, and data upload task. Figure 21 shows the serial port interruption and specific task flow charts.

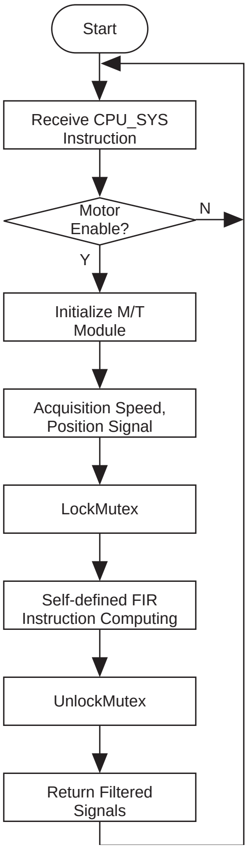
Figure 21. Serial Port Interruption Service and Task Flow Charts



CPU_DSP Implementation

CPU_SYS, the data processing soft core, initializes the M/T module, receives CPU_SYS commands and FIR filter parameters through the message mailbox, protects the shared memory using hardware resistive cores, and uses a custom multiply-add instruction for efficient digital FIR filter computing. Figure 22 shows the CPU_DSP flow chart.

Figure 22. CPU_DSP Flow Chart


Start
Receive CPU_SYS Instruction
Motor Enable?
N
Y
Initialize M/T Module
Acquisition Speed, Position Signal
LockMutex
Self-defined FIR Instruction Computing
UnlockMutex
Return Filtered Signals

Design Methodology

We used the following design methodology.

Nios II System Overall Resource Configuration

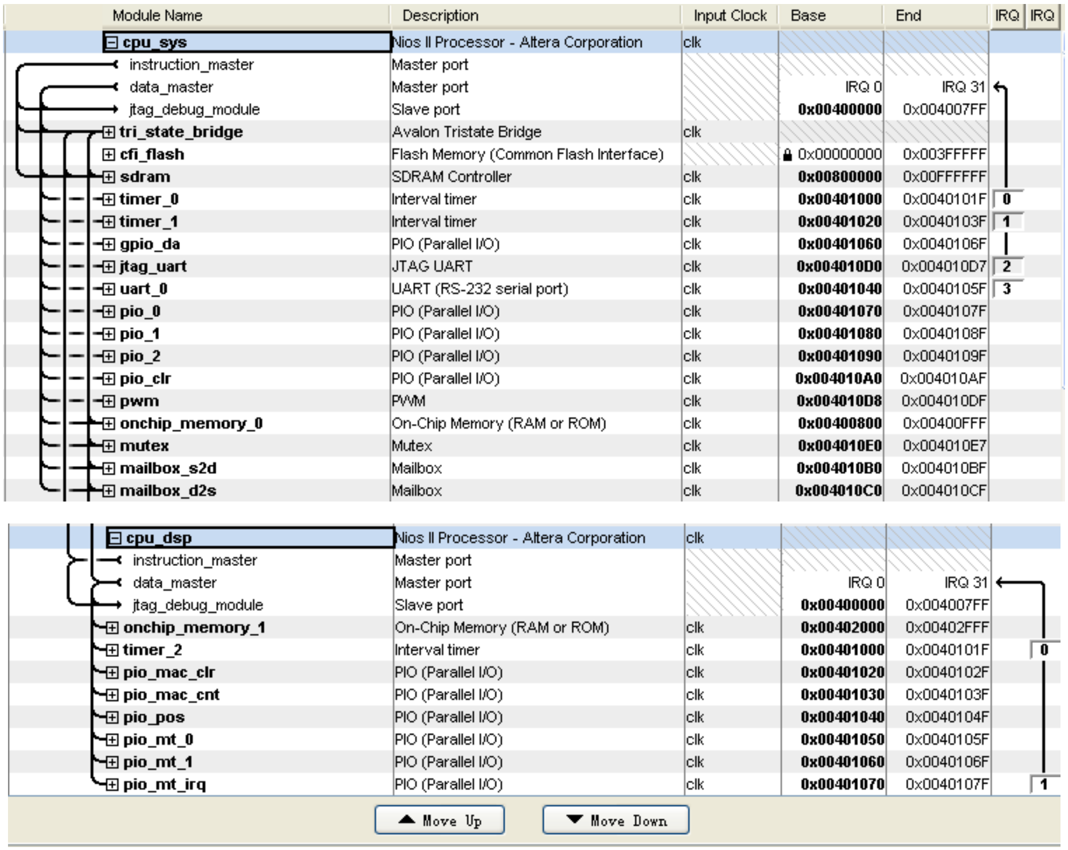
We implemented a multi-processor structure based on the Avalon bus: the dual cores share SDRAM and flash memory and start from different flash memory addresses by setting different reset addresses. We set multiple startup codes and programs in a single flash device based on the Nios II Integrated Development Environment (IDE) programming environment. By setting a break address, we can divide the SDRAM into two equal parts that operate in their respective areas without interfering with each other. The programmable I/O (PIO) ports are configured for the cores to set the signal subdivision decoding unit and read the position counting and M/T speed measurement results. The UART serial port communication module is configured for CPU_SYS to communicate with the high-level LabWinsows/ CVI computer and control the motor state in real time. Considering memory's read/write speed, the system allocates a 4-Kbyte on-chip memory for CPU_DSP to implement fast storage position sampling and speed signals. The dual cores share another 4-Kbyte on-chip memory for FIR filter coefficients and to protect resources using Mutex resistive cores. The dual cores send messages through two mailboxes for task synchronization. Figure 23 shows the resource configuration structure.

Figure 23. Nios II System Resource Configuration Structure

Admin / Resource	CPU_DSP	CPU_SYS
PIO	5 Groups	6 Groups
SDRAM	4M	4M
FLASH	2M	2M
UART	UART_0	
PWM	1 Channel	
On-Chip Memory	4KB (Shared by Mutex)	
		4KB (Fast Signal Sampling)
Mailbox	Mailbox_d2s →	
	Mailbox_d2s ←	

We implemented this configuration in SOPC Builder and generated the Nios II system. Figure 24 shows the system structure.

Figure 24. Nios II System in SOPC Builder



Module Name	Description	Input Clock	Base	End	IRQ	IRQ
⊟ cpu_sys	Nios II Processor - Altera Corporation	clk				
instruction_master	Master port					
data_master	Master port		IRQ 0	IRQ 31		
jtag_debug_module	Slave port		0x00400000	0x004007FF		
⊞ tri_state_bridge	Avalon Tristate Bridge	clk				
⊞ cfi_flash	Flash Memory (Common Flash Interface)		0x00000000	0x003FFFFF		
⊞ sdram	SDRAM Controller	clk	0x00800000	0x00FFFFFF		
⊞ timer_0	Interval timer	clk	0x00401000	0x0040101F	0	
⊞ timer_1	Interval timer	clk	0x00401020	0x0040103F	1	
⊞ gpio_da	PIO (Parallel I/O)	clk	0x00401060	0x0040106F		
⊞ jtag_uart	JTAG UART	clk	0x004010D0	0x004010D7	2	
⊞ uart_0	UART (RS-232 serial port)	clk	0x00401040	0x0040105F	3	
⊞ pio_0	PIO (Parallel I/O)	clk	0x00401070	0x0040107F		
⊞ pio_1	PIO (Parallel I/O)	clk	0x00401080	0x0040108F		
⊞ pio_2	PIO (Parallel I/O)	clk	0x00401090	0x0040109F		
⊞ pio_clr	PIO (Parallel I/O)	clk	0x004010A0	0x004010AF		
⊞ pwm	PWM	clk	0x004010D8	0x004010DF		
⊞ onchip_memory_0	On-Chip Memory (RAM or ROM)	clk	0x00400800	0x00400FFF		
⊞ mutex	Mutex	clk	0x004010E0	0x004010E7		
⊞ mailbox_s2d	Mailbox	clk	0x004010B0	0x004010BF		
⊞ mailbox_d2s	Mailbox	clk	0x004010C0	0x004010CF		
⊟ cpu_dsp	Nios II Processor - Altera Corporation	clk				
instruction_master	Master port					
data_master	Master port		IRQ 0	IRQ 31		
jtag_debug_module	Slave port		0x00400000	0x004007FF		
⊞ onchip_memory_1	On-Chip Memory (RAM or ROM)	clk	0x00402000	0x00402FFF		
⊞ timer_2	Interval timer	clk	0x00401000	0x0040101F		0
⊞ pio_mac_clr	PIO (Parallel I/O)	clk	0x00401020	0x0040102F		
⊞ pio_mac_cnt	PIO (Parallel I/O)	clk	0x00401030	0x0040103F		
⊞ pio_pos	PIO (Parallel I/O)	clk	0x00401040	0x0040104F		
⊞ pio_mt_0	PIO (Parallel I/O)	clk	0x00401050	0x0040105F		
⊞ pio_mt_1	PIO (Parallel I/O)	clk	0x00401060	0x0040106F		
⊞ pio_mt_irq	PIO (Parallel I/O)	clk	0x00401070	0x0040107F		1

Nios II Soft-Core Interaction

We added Mutex components to the design to prevent operation errors when the CPU_DSP and CPU_SYS cores modify the shared FIR parameters. Before accessing shared resources, the processor first tests whether Mutex is available; if it is, the resource access right is obtained in the operation. After using the Mutex-related shared resources, the processor releases the Mutex. Resource sharing between the dual cores is implemented with the hardware configuration shown previously.

The cores interact using MailBox; we configured `MailBox_d2s` and `MailBox_s2d` in this design. MailBox consists of two resistive cores and a shared memory. The resistive cores ensure single read-write operation for the shared memory, and the size of the shared memory is configurable. In the HAL, Altera provides the `altera_avalon_mailbox_pend()` and `altera_avalon_mailbox_post()` functions for using the mailbox. We implement message interaction and task synchronization between the cores using these functions.

Custom PWM Peripherals

To acquire the PWM drive signal, we designed custom PWM peripherals. See Figure 14 on page 182 for the hardware structure. The Nios II soft-core processor accesses the peripherals through the Avalon bus. The access interfaces include `reset` (bus reset signal), `clk` (bus clock), `avalon_chip_select` (bus chip select signal), `addrest` (1-bit address signal), `write` (write enable signal), and `writedata` (32-bit write data signal). The peripherals export drive signals through the `pwm_out` pin.

The write drive functions are `user_avalon_pwm_init()` (for initialization of the custom peripherals), `user_avalon_pwm_enable()` (for enabling the output), `user_avalon_pwm_disable()` (for disabling the output), and `user_avalon_pwm_data()` (for setting the duty ratio and frequency).

Custom Multiply-Add Instruction

For a Nios II embedded system, time-critical software algorithms can be executed faster by adding custom instructions. These instructions simplify complex standard instruction sequences into one instruction that is implemented in hardware, thereby optimizing the algorithm structure. In this design, we customize a multi-cycle instruction for the multiply-add operation, which is a key step in the digital FIR filter algorithm. At a 100-MHz clock frequency, it takes only 40 ns to complete one double 16-bit multiply-add operation, which is faster than the software algorithm. The latency for 128-order filtering is only 13 µs, improving the real-time system performance. Figure 25 shows the multiply-add instruction top-level structure.

Figure 25. Multiply-Add Instruction Top-Level Structure

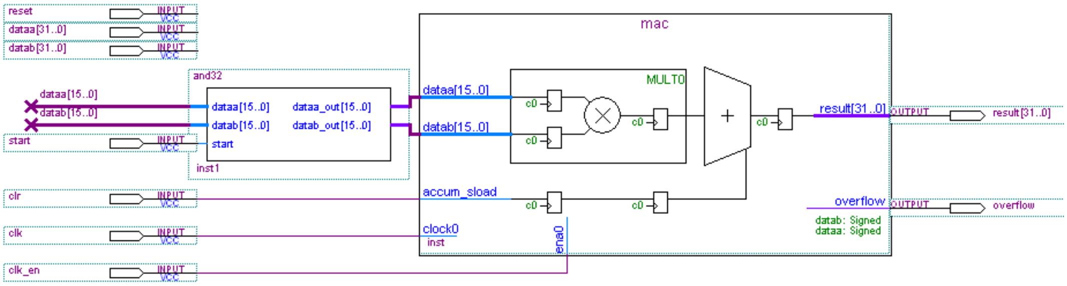
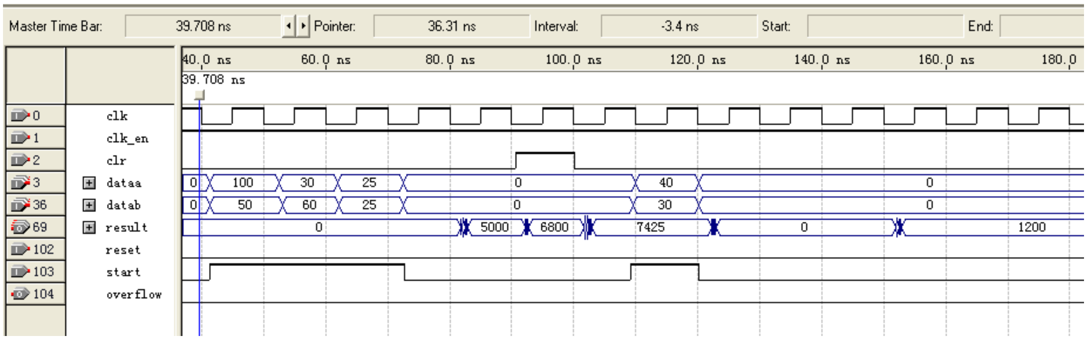


Figure 26 shows the simulation result of this module. In this figure, `clk` is 100 MHz, `dataa` and `datab` perform synchronous computation with `clk` under the control of the `start` signal, and the accumulator signal `overflow` indicates whether there is overflow in the result.

Figure 26. Multiply-Add Instruction Simulation Result



Design Features

Our design has the following features:

- The design implements a digital FIR filter with adjustable parameters using the Nios II processor to curb noise caused by gas film disturbances, mechanical resonance, etc. in the laser direct writing motion control. We customized a special hardware multiply-add instruction for the filtering algorithm to improve the algorithm's execution efficiency and shorten the software implementation time.

- The PWM subordinate peripheral waveform, based on the Avalon bus, is customized and the DAC drive unit outputs the DC voltage. With this implementation, the system can match two drive modes (PWM and DC voltage) of the motor servo driver, reflecting the flexibility of SOPC design.

- For the current VCM digital servo motion control, the MCU (or DSP) is always used for the control algorithm and the FPGA performs signal decoding. The signals are require a lot of resources, and are unstable and vulnerable to noise due to their unmatched speed with on-board signals. Using an FPGA with embedded dual Nios II processors as the main control kernel, this design implements the MCP and DSP functions at the same time, integrates the system to the greatest extent, greatly reduces electromagnetic interference, and improves system stability. Additionally, it simplifies software and hardware design and shortens the product development cycle.

- Migrating the reliable, real-time µC/OS-II into the Nios II processor has a significant effect in critical areas such as motor control.

- We used the LabWindow/CVI software to write the PC driver and operating interface, implement a friendly user interface, and complete off-line analysis of the VCM motion noise spectrum characteristics.

- With a larger FPGA, we could integrate several VCM digital servo controllers into a single chip to implement fast, high-precision monitoring of a multiple-degree-of-freedom motor system.

Conclusion

Altera's RISC-based Nios II soft-core processor had strong performance during system design and debugging. The VCM servo controller with an embedded Nios II soft-core processor based on SOPC technology features fast, flexible, scalable hardware design. SOPC Builder and the Nios II IDE provide a complete solution for software development. The design gives full play to the high-speed, high-integration features of the FPGA, allowing us to implement a complete motor servo control solution on a single chip.

In this design, the PIO reads the M/T decoding module result. We could greatly improve the CPU utilization if this module were re-created as a custom peripheral and connected to the Avalon bus. For the digital FIR filter design, considering the system servo cycle, our project implements a filtering operation of not more than 128 orders. If we used a faster clock, linear phase compensation, and a further optimized custom hardware structure, we could shorten the phase delay and further improve the system control precision. Using the serial port for the data transmission channel is not fast and affects the real-time system performance; using a high-speed bus such as USB can improve this area.

With development of the manufacturing industry, the technical requirements for motor servo control has increased. Our control system based on integrated SOPC technology has significant advantages in system volume, speed, signal integrity, etc., and will become the future trend of motor servo control systems.