DESIGN AND IMPLEMENTATION OF MOTION AND I/O CONTROL ASIC - EPCIO

Yuan-Wei Tseng, Ph.D.[1]
Wen-Chuan Chen, Associate Researcher [2]
Armin Chen, Manager [3]
E100 MIRL/ITRI, Bldg. 21, 195-3 Chung Hsing Rd., Section 4,
Chutung, Hsinchu, Taiwan 310, R.O.C.
[1]E-mail: 860663@itri.org.tw
[2]E-mail: 850287@itri.org.tw
[3] E-mail: 820525@itri.org.tw

Abstract

Design and implementation of the motion and I/O control ASIC-EPCIO developed by Mechanical Industry Research Laboratories (MIRL) are presented in this paper. When EPCIO is in master mode, it can control six servomotors and two spindle motors using eight sets of 16-bit digital to analog control circuits. Its nine 32-bit pulse counters can read counter values from six external encoders and from three hand wheels. In addition, eight external signals can be sensed through its 12-bit analog to digital interface. Furthermore, two remote I/O control modules can handle up to 384 remote inputs and 384 remote outputs. When EPCIO acts as slave mode, it can deal with 64 inputs and 64 outputs. Therefore, applying EPCIO in both motion and I/O control card, designs as well as maintenance become very easy.

Key Words: EPCIO, ASIC (Application Specification Integrated Circuit), Motion and I/O Control, PC-based

1. Introduction to EPCIO

Mechanical Industry Research Laboratories at Industrial Technology Research Institute (MIRL, ITRI) recently announced the advent of a new ASIC, EPCIO, which is the abbreviation of “exquisite positioning control, inputs and outputs”. The intention of developing EPCIO is to integrate the above indispensable functions of motion and I/O controllers into an ASIC with one hundred twenty thousand gates. In order to not only reduce cost but also improve the maintenance. Furthermore, the circuit design of PC-based controller becomes very feasible and flexible when EPCIO is applied. The purposes of developing EPCIO are as follows.

1. Integrating mature technologies to reduce costs at manufacture and maintenance.
2. Simplifying controller architecture to improve reliability.
3. Providing flexibility in designs of PC-based motion and I/O cards to enhance the competitive ability in motion and I/O control market.

4. According to the needs of both Taiwan and worldwide industries, the target markets of EPCIO include milling and drilling machines, which could have 3 servo motors, 128 digital inputs, 128 digital outs and 1 analog output. EPCIO is also suitable for IC wire bonding machine, which could have 3 servomotors, 8 stepping motors, 80 digital inputs, 40 digital outputs.

2 Introduction to Functional Modules of EPCIO

Interior modules of EPCIO are shown in figure 1. The major features of EPCIO are divided into the following categories.

1. Positioning control:

For servo motor control, EPCIO includes six sets of positioning control loops, digital to analog interfaces, together with 6 encoder counters to achieve up to six axial positioning closed loop control. Six digital differential analyzers can generate pulse commands. Furthermore, additional three encoder counters are available to 3 hand wheel inputs. Another pair of digital to analog interfaces are available for control of 2 spindle motors.

2. Analog interface:

EPCIO has a set of 8 analog to digital interfaces, which connect to MAX 186 (analog to digital IC), allowing EPCIO to serially deal with up to 8 sensed external analog signals.

3. 28 local input/output ports:

There are seven identical circuit modules in EPCIO. Each module contains four programmable input or output ports. Therefore, EPCIO can control twenty-eight local inputs or outputs.

4. 768 Remote control input/output ports: When "Mode Pin" of EPCIO is connected to Vcc (+5V), EPCIO is instantiated as "Master mode". When "Mode Pin" of EPCIO is connected to GND (0V), EPCIO is instantiated as "Slave mode". One EPCIO in master mode along with six EPCIO in slave mode can handle up to 384 serial inputs and 384 outputs.

5. Auxiliary circuits:

To improve the flexibility of implementation, EPCIO also has many auxiliary functions such as ISA bus interface, a variety of interruption controllers, watch dog and interval timer.

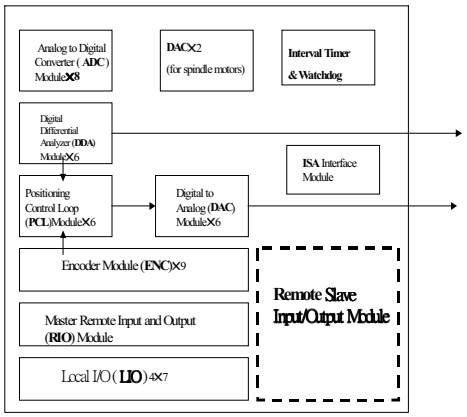

Figure 1. Interior modules of EPCIO

2.1 ISA Interface Module

EPCIO provides ISA interface, which is compatible to many available PC-based industrial computers. Figure 2 shows the block diagram of ISA interface. Address decoder along with corresponding control units generate signals can reset all modules in EPCIO or can generate interruption and can set wait state. Furthermore, based on page and address, address decoder can also work with a multiplex (MUX) to transfer

data or Page number to host PC for process. Both signal LAdd (local address) and signal Page, and decoders in the other modules to trigger the rest of all functional modules in EPCIO.

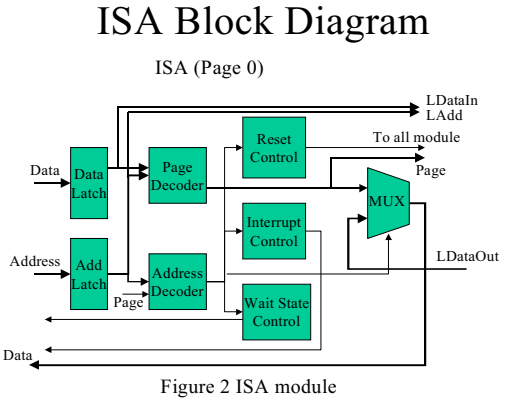

Figure 2 ISA module

2.2 Digital Differential Analyzer (DDA) Module

The purpose of utilizing DDA module is to allow EPCIO to smoothly send pulse commands out using DDA algorithm [1]. Figure 3 shows the interior functional blocks of DDA module. DDA module consists of 63 sets of 16 bit FIFO (First in first out stack) which can store up to 63 DDA commands, allowing DDA module to smoothly send DDA commands through Pulse Generator and Pulse Out Control between two interrupts. Therefore, utilizing FIFO can obtain better interpolation as well as reduce the number of interruption. Based on FIFO information, interrupt can be generated. For example, an interrupt to request new DDA commands can occur when the remaining number of the DDA commands in FIFO reaches to a specific number preset by EPCIO. The function of DDA CLK & Length Control unit is to change both DDA length and DDA clock frequency while Pulse Generator Control unit manages the Pulse signal page, which is generated through page decoder, are used together with address generator and send interrupt to Interrupt Control & Latch unit for further process for interruption. Address decoder can control MUX to send information of either interrupt or current DDA command in FIFO back to host PC.

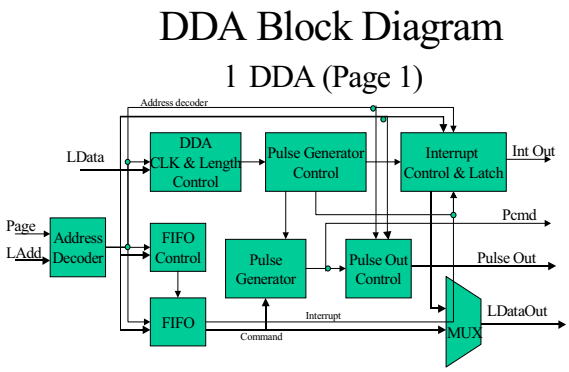

Figure 3 DDA module

2.3 Encoder Module

Figure 4 shows the interior functional blocks of Encoder module. There are two digital filters to reject the noises of A/B phase signals and index signal C. Clock Divider is the frequency-divided clock source of digital filters. Based on A/B phase input signals, Feedback Decoder can generate × 1, × 2, × 4 pulses while Input Control can change the sign of these pulses, and then send those pulses to 32 bit Counter and to Positioning Control Loop, which is explained in next section. To meet 16 bit data format of ISA bus, 32 bit data in Counter are divided into two sets of 16 bit data, which are consecutively sent to host PC through 16 bit Latch and MUX. EPCIO can set a Pre-load Value to compare with Counter value to generate proper interruption. Latch control can enable Index Latch due to control word setting or external trigger. Interrupt Control & Latch

manages the interruption of encoder module. According to address decoding result, proper data are transferred to host PC via MUX and ISA bus.

ENC Block Diagram

1 ENCCNT (Page 2-4)

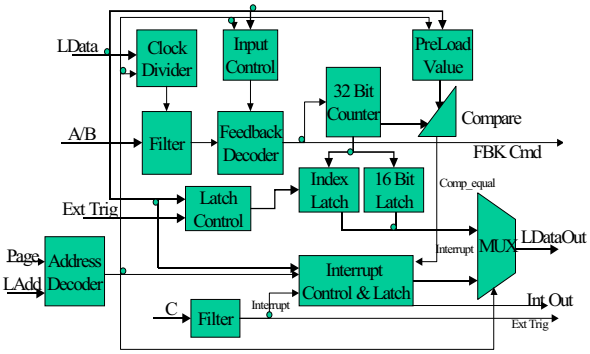

Figure 4 Encoder Module

2.4 Positioning Control Loop Module

This module with closed loop control function, can generate compensate commands to digital to analog converter module (DAC module) according to positioning error. Positioning error is computed in Summer. Then the result is sent to 16 bit Error counters. 24 bit compensate command is obtained by 8 bit Gain multiples 16 bit Error and is sent to Shift Register. The function of Wait Control is to set enough wait states to ensure that the multiplication is complete. Compensate command in Shift Register can be further shifted by Shift Gain. The highest 4 bits of Shift Register are reserved for overflow control. If the final result in Shift Register exceeds the DAC range, Overflow Control will limit the output at the extreme values of DAC register. If no overflow occurs, the last 4 bits of compensate command in Shift Register will be trimmed so that a 16 bit compensate command is sent to DAC register. Interrupt Control & Latch manages the interruption of positioning control loop module.

PCL Block Diagram

1 PCL (Page 9)

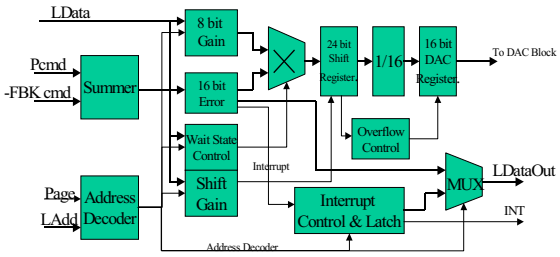

Figure 5 PCL Module

2.5 Digital to Analog (DAC) Module

The function of DAC module is to generate serial commands to analog chips (PMC-56P) on controller board. Therefore, DAC module is basically a digital to analog interface. There are two types of sources of DAC commands. One is generated from positioning control loop (PCL) module, the other is obtained form software command. Software command can be obtained either from DAC register or Pre-load value triggered by external inputs. Trigger Source Control unit manages the selection of software command. DAC Source Control unit selects the sources of DAC command. DAC command is converted into serial digital commands in Serial Interface unit, which is triggered by frequency-divided clock generated from Clock Divider unit.

DAC Block Diagram

1 DAC (Page 10-12)

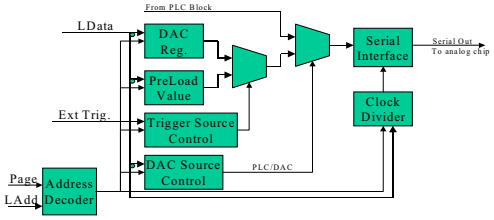

Figure 6 DAC Module

2.6 Analog to Digital Converter (ADC) Module

EPCIO provides MAX186 format ADC interface for 8 analog signals. Clock Divider provides a frequency-divided clock source (SCLK). ADC Control unit manages the selection of analog signals, which has been converted into 12 bit digital signal by MAX186. Serial Interface unit then converts this 12 bit signal into serial signal and then send it out. Comparison Register restores 16 bit data for comparison with external signal to generate interrupt. Analog signal is first converted into 12 bit digital signal by MAX186, then is sent to Shift Register unit to generate corresponding 16 bit data. Mask unit can trim the last 0, 1,2 or 3 bit of these 16 bit data for the purpose of noise rejection. Then, the obtained data is compared with data in Comparison Register to generate a signal sent to Interrupt Control & Latch unit, which manages the interruption of ADC module. MUX can send back proper data to host PC through ISA bus.

ADC Block Diagram

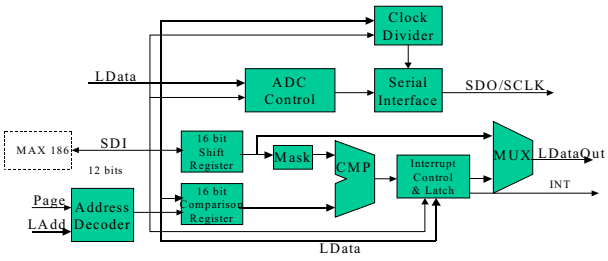

Figure 7 ADC Module

2.7 Local I/O (LIO), Timer & Watch Dog Modules

Parallel data of 28 local in/outputs are restored in DI register or DO register. DO Control unit controls the direction of local data flow. Interval Timer unit regularly send signals to both Watch Dog and Interrupt Control & Latch unit. Watch dog with a counter inside can reset the system if no system action takes place after specific number of signals received from Interval Timer.

LIO, Timer & Watch Dog Block Diagram

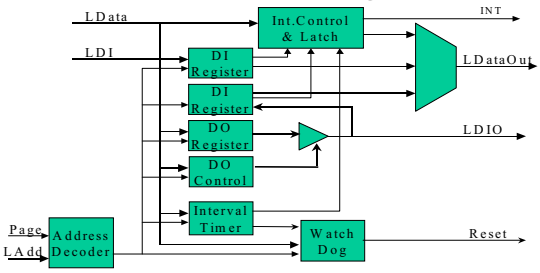

Figure 8 LIO and auxiliary modules

2.8 Remote Input and Output (RIO) Master Module

There are two master RIO modules in EPCIO. Each master RIO module controls up to 3 slave modules. Every slave module has 64 inputs and 64 outputs. Serial data are transferred between master and slave modules as follows. Output register can restore up to 3 sets of 64 bit data, which will go to slave modules after being converted into serial data in Shift Register unit. The corresponding cyclic redundancy checker codes (CRC) [2] of every set of serial data is generated in CRC generator and is sent to remote slave module along with serial data via MUX. The result of data transmission will be sent back to Echo Code Checker unit. If error occurred in the transmission, Echo Code Checker will notify Shift Register to send the data again and the value of Transmission Error Counter will be increased by one. When Transmission Error Counter accumulates specific number of error, an interrupt request will be sent to Interrupt Control & Latch unit. Shift Register unit can also convert the input serial data (SDI) from slave modules into up to 3

sets of 64 bit data, which will be restored in Input Register later. Every set of CRC code of input serial data, which are sent from remote slave module is examined in CRC Checker. If it is correct, Echo Code Generator will send a confirm signal to remote slave module via MUX.

RIO Block Diagram(Master Mode)

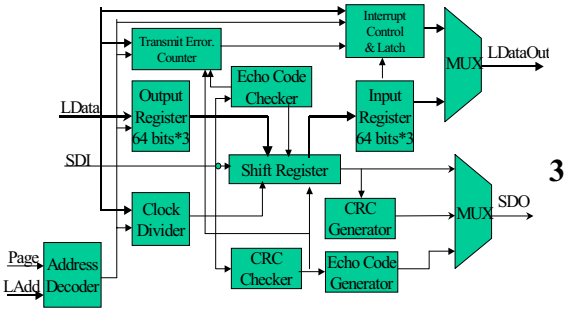

Figure 9 RIO Module (Master Mode)

2.9 Remote Input / Output (RIO) Slave Module

When EPCIO is preset as slave mode, only this module in EPCIO is active. Slave module can control 64 digital input pads and 64 digital output pads. 64 digital inputs stored in Input Register are converted into serial data in Shift Register. Then serial data are transferred to remote master module along with CRC code through MUX. After remote master confirms the receiving of correct data, an echo code will send back to Echo Code Checker in slave module. On the other hand, incoming Serial Data input (SDI) from remote master are converted into 64 bit data in Shift Register while CRC checker examines the accompanied CRC code. If it is correct, Echo Code Generator will send back a confirm signal to remote master module via MUX and the 64 bit data will be sent to 64 digital output pads in slave modules through Output Register.

RIO Block Diagram(Slave Mode)

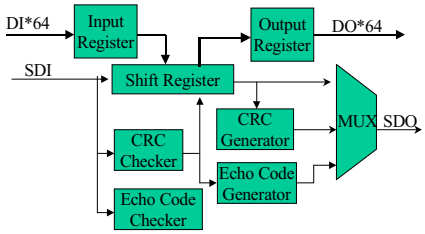

Figure 10 Slave I/O Module

3. Implementation of EPCIO

Since EPCIO has many modules inside, it provides a lot of flexibility for motion and I/O add-on card designs. Based on this low cost but versatile ASIC, a full functional control board can be designed and is illustrated in figure 11. The connections between EPCIO master board and EPCIO slave board can also be visualized in figure 11. Therefore, ample features in EPCIO can be tailored dependent of the needs for custom designs of motion and I/O cards.

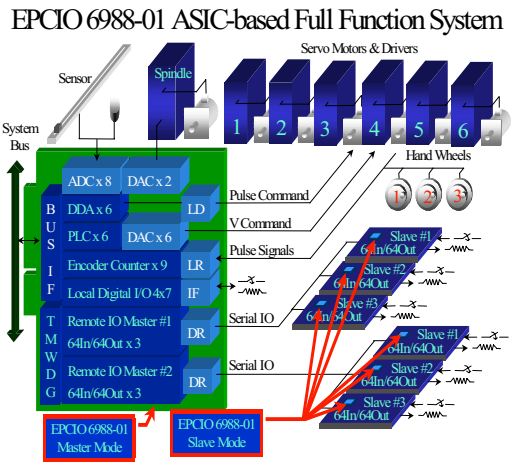

Figure 11 EPCIO-based control system

MIRL is currently developing various motion and I/O cards based on EPCIO. Currently, EPCIO-601 in figure12 and EPCIO 6000 in figure 13 are

under tests.

Figure 12 EPCIO-601 (with ISA bus)

Features of both EPCIO-601 and EPCIO-6000 are very similar to each other. The major difference between them is bus interface. EPCIO-601 has ISA bus while EPCIO-6000 employs PCI bus.

Figure 13 EPCIO-6000 (with PCI bus)

The features of EPCIO-6000 are introduced as follows.

❒ Bus Interface: PCI

❒ Motion:

- Positioning Axes: 6
- Max. DDA Command: 2^{10-15} pulse
- DDA Cycle: 25us ~3350ms Programmable
- Velocity Command Range: +/- 10V
- Pulse Output Format: Pulse/Direction, CW/CCW, A/B Phase
- Error Counter (For Output Pulse): 16 bits

❒ Remote IO:128 IN/128 OUT Maximum

❒ D/A Converter: 6 channels with 16 Bit Resolution

❒ A/D Converter Interface: 8 channels 12bit Resolution

❒ Encoder Input:

- 6 Axes
- Interface: Line Driver
- Input Format: A/B/Z Phase, pulse/direction , CW/CCW
- Decoder: x0, x1,x2 ,x4 software programmable in A/B/Z phase input
- Position Counter: 32 bits
- Latch: 15 trigger signals for each axis

❒ Local Inputs/Outputs:

- Home Sensor Signal Input: 6
- Positive Over Travel Signal Input: 6
- Negative Over Travel Signal Input: 6
- Inhibit Signal Output: 6
- Emergency Stop Input: 1
- Position Ready Output: 1

4. Future development of controllers using EPCIO

With the increasing requirements in real-time performance and user friendly but very CPU time- consuming windows based human machine interface (HMI) in industrial PC-based controllers, a competitive design must find a breakaway from the above dilemma. In the point of view in hardware design, using digital signal processor (DSP) to handle tasks with real time performance requirement is a solution. Therefore, MIRL is devoting itself in developing PC-based controller with both EPCIO and DSP. The preliminary plan

is to design two such controllers, namely PMC32-600 and PMC32-6000, with ISA bus and PCI bus respectively. For example, PMC32-600 uses a 32 bit floating point DSP as arithmetic kernel along with EPCIO to construct a powerful motion and I/O controller with neat architecture. PMC32-600 will have 6 positioning pulse outputs as well as 6 positioning closed loops, 6 encoder inputs, 1 hand wheel input, 8 analog to digital converter interface, 8 digital to analog converter interface, and 1 set of remote serial I/O interface, which can handle 64 inputs and 64 outputs. Since PMC32-600 has a DSP, it can stand alone as an embedded system controller. An alternative application for PMC32-600 is to work together with an industrial personal computer (IPC) to become a Dual-CPU system. Therefore, IPC can take care of tasks, which do not need real time responses, such as HMI, collecting and analyzing data and selecting motion command types (point to point, trajectory following, etc.). On the other hand, real time tasks such as command interpretation, trajectory planning and servo control, are controlled by PMC32-600. This architecture is not only powerful but also flexible. Furthermore, it even can achieve hard real time requirement when it operates in Microsoft Windows operating systems.

5. Conclusion

EPCIO is indeed a versatile motion and I/O control ASIC, which can meet the requirements in many applications. The long-term plan for MIRL is to design a more advanced ASIC with both EPCIO and DSP modules. In a word, with the help of considerable progresses in IC industries in Taiwan, MIRL will endeavor to develop system–on-chip motion controller based on the success of EPCIO.

Reference

[1] T.R. Sizer, “The Digital Differential Analyzer”, Campman & Hall, London, 1968.

[2] William Stallings, “Data Computer Communications”, page 105-110, ISBN 0-02-415440-7, Stallings, 1985