

Microprocessor Controlled Milling Machine: A Student Project

Mohammad Fotouhi, Ali Eydgahi, Joshua Wagner
University of Maryland Eastern Shore

Abstract

This paper describes the details of an undergraduate design project in our Design Technology course and the experience gain by the student involved. The intent of the course is to expose students to real world design projects. Students are expected to be creative and innovative in their design projects and utilize a multitude of engineering disciplines that Engineering Technology Program offers at the University of Maryland Eastern Shore. The objective of this project was to use a 68HC12 Motorola micro-controller to control a three-axis motion of a milling machine. This micro-controller was chosen because it has multiple inputs/outputs, built in 8-bit analog to digital converter and timer, pulse width modulator, 16 bit I/O bus, and on board flash memory. In this project, student designed an interfaced circuit to micro-controller and wrote a program in assembly language to control the motion of each stepper motor that is responsible for each axial.

I. Introduction

The undergraduate major of Electronic Engineering Technology at the University of Maryland Eastern Shore requires each student to complete a design course. The interface of a milling machine to a Motorola micro-controller was one of the design projects offered in this course¹⁻³.

The milling machine was a small tabletop model that has a two-direction movable vise, and an adjustable height cutter head. A small 120-volt AC motor turns the cutter head. The cutter head speed can be controlled electronically from 200 to 2000 RPM's. With the two-axis (X, Y) movable vise and the single axis (Z) movable cutter head, three axis (X, Y, Z) control of the cutting tool is achievable. Each axis is driven by a stepper motor turning a lead screw attached to the movable structure at one end. The other end is threaded through a fixed block and is coupled to the stepper motors.

The stepper motors utilized in the milling machine are bipolar and variable reluctance types with 200 discrete steps per revolution. The lead screws have 16 threads per inch of travel. It is possible to control the position of the cutter head to 1/3200 of an inch with a 1.8 degree step angle. The minimum power required by the stepper motors is 4.75^V DC at 1 Ampere. This type of motor requires a push-pull bipolar drive circuit. The drive circuit

requires 8 high power transistors connected as shown in Figure 1 to drive only one of the stepper motors.

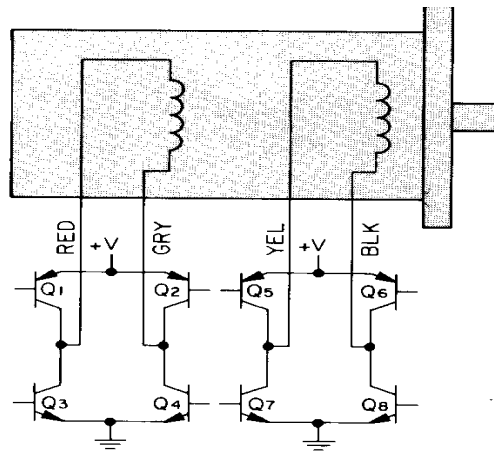


Figure 1: Connection of transistors driver to a bipolar stepper motor

This type of circuit is sometimes referred to as an H-drive, since the connection of the transistors looks like a H. By switching the transistors on or off in the right combination, the current flow through the motor windings can be reversed. Caution must be used not to short the power supply by turning on the wrong combination of transistors. Once the H-drive was constructed, tests were performed by manually applying power to the stepper motor windings as shown in Table 1.

Table 1: Stepping sequences

STEP	Q1-Q4	Q2-Q3	Q5-Q8	Q6-Q7
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

This was accomplished by applying a bias voltage to the base connection of each of the appropriate transistors. In order to rotate the stepper motor in a clockwise direction the windings had to be energized in a stepping order of 1, 2, 3, 4, 1, etc. To rotate the motor in the opposite direction the stepping order needs to be reversed as 4, 3, 2, 1, 4, etc. The stepper motor performed best when the input voltage ($V+$) was approximately 15^V DC.

II. The Design Project

To gain experience, a Heathkit 6800 microprocessor trainer board was used to interface a stepper motor to the microprocessor. The goal was to control one stepper motor with the

6800 trainer board. Experience was gained by varying both software and hardware, and taking note of the effects that these changes made in the stepper motor performance. Figure 2 shows a block diagram of a typical driving system for a stepper motor.

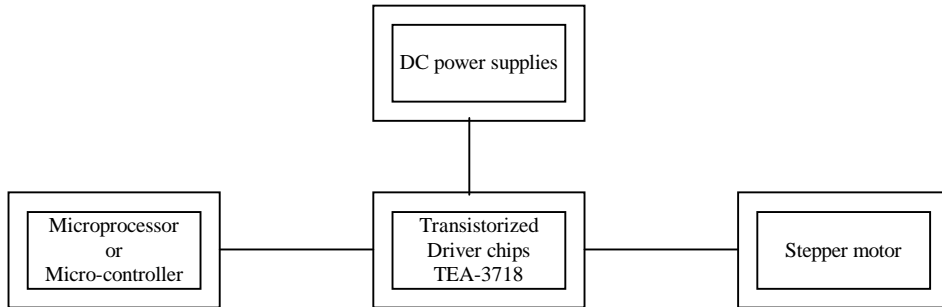


Figure 2: Typical driving system for a stepper motor

To reduce the amount of breadboard connections and circuit complexity, driver chips were used. The chips used were model TEA-3718 made by SGS-THOMSON. The chips are essentially half of the transistors circuit shown in Figure 1. Therefore, it takes two of these chips to run one stepper motor, one for each motor winding. By following the recommendations of the driver chip manufacturer, the required circuit to interface the power supply, driver chips, microprocessor, and the stepper motor was constructed as shown in Figure 3. The schematic shows connections needed between the power supplies, driver chips, and stepper motor.

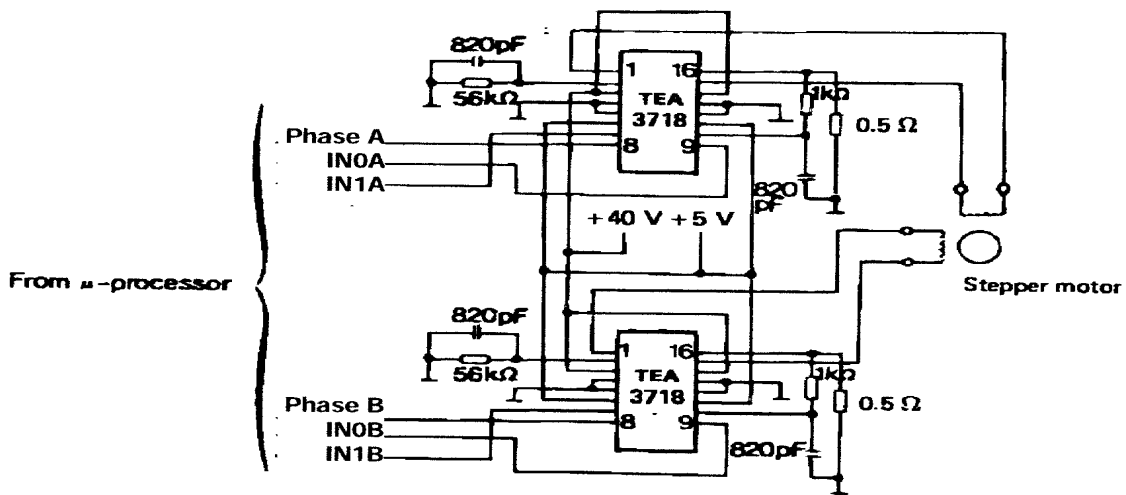


Figure 3: Schematic of driver chip connections

The chips require two external power supplies. The first one is a 5^V DC to power the logic circuit in the chip and the second one is a 15^V DC that powers the stepper motor windings by getting switched on or off by the chip. Each chip has three inputs that need to be connected to the microprocessor data bus for control signals. To drive one stepper motor 6 data lines are required since two chips are needed per motor. In this design, several of the chip inputs are connected together to reduce the number of data lines to three needed per motor. The input pins connected together are IN0A, IN0B, IN1A, and IN1B. This reduces the complexity of the programming required to send the correct data to the driver chips. Table 2 shows the connections made between the microprocessor data bus and the driver chips.

Table 2: Interface between microprocessor and driver chips

Microprocessor data line	Driver chip input
D ₁	Phase A
D ₂	Phase B
D ₃	IN0A, IN0B, IN1A, IN1B

The phase A and phase B inputs determine the direction of current flow in the winding. The IN0 and IN1 inputs turn the current to the windings on or off. When these inputs go to zero volts, current is allowed to flow through the motor windings. Figure 4 shows inputs signals needed as they were coming from the data bus of the 6800 trainer.

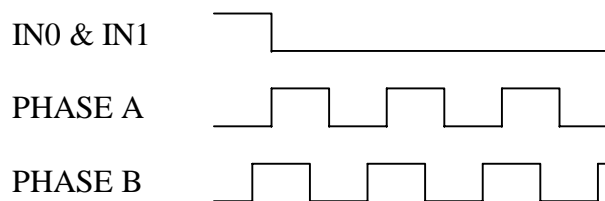


Figure 4: Signals needed from microprocessor data bus to provide motor control

The above signals were created by writing a program that provides binary data representing these signals on the microprocessor data bus. Table 3 shows the assembly language program that provides the signals shown in Figure 4.

Table 3: Signals needed from microprocessor data bus to provide motor control

PROGRAM FOR THE 6800 MICROPROCESSOR TRAINER, D0, D1, & D2 ARE USED AS INPUTS TO THE STEPPER MOTOR DRIVER CHIPS.						
0000		LDAA	#\$01	86	01	
0002		STAA	\$\$9000	B7	90	00
0005		JSR	DELAY	BD	00	23
0008		LDAA	#\$03	86	03	
000A		STAA	\$\$9000	B7	90	00
000D		JSR	DELAY	BD	00	23
0010		LDAA	#\$02	86	02	
0012		STAA	\$\$9000	B7	90	00
0015		JSR	DELAY	BD	00	23
0018		LDAA	#\$00	86	00	
001A		STAA	\$\$9000	B7	90	00
001D		JSR	DELAY	BD	00	23
0020		JMP	\$\$0000	7E	00	00
0023	DELAY	LDAX	#\$00FF	CE	00	FF
0026	LOOP	NOP		01		
0027		DEX		09		
0028		CPX	#\$0000	8C	00	00
002B		BNE	LOOP	26	FA	
002D		RTS		39		

Because the data is only present on the data bus for a short period of time, a buffer chip was used to capture the data from the data bus and send the desired data to the driver chips. A decoder/demultiplexer chip is also used to turn on the buffer chip only when valid data is being sent. This demultiplexer chip receives inputs from the address bus of the microprocessor and only sent out a chip enable signal to the data buffer chip when the address location used in the program is being accessed as shown in Figure 5. Once experience of controlling stepper motor using the 6800 trainer was gained, the next task was to control all three stepper motors with the micro-controller.

The micro-controller was a Motorola model MC68HC912B32⁴. There was an immediate advantage of the micro-controller in comparison to the 6800 trainer. The micro-controller can be programmed by several methods. In this case, the host-computer method was chosen since the program code could be generated off-board. The programs used with the micro-controller were written in assembly language. This was done by

writing the programs in a DOS based editor, then using an assembler to convert the program into machine language. Software that accompanied the micro-controller was used to download the program to flash EEPROM memory on the micro-controller chip.

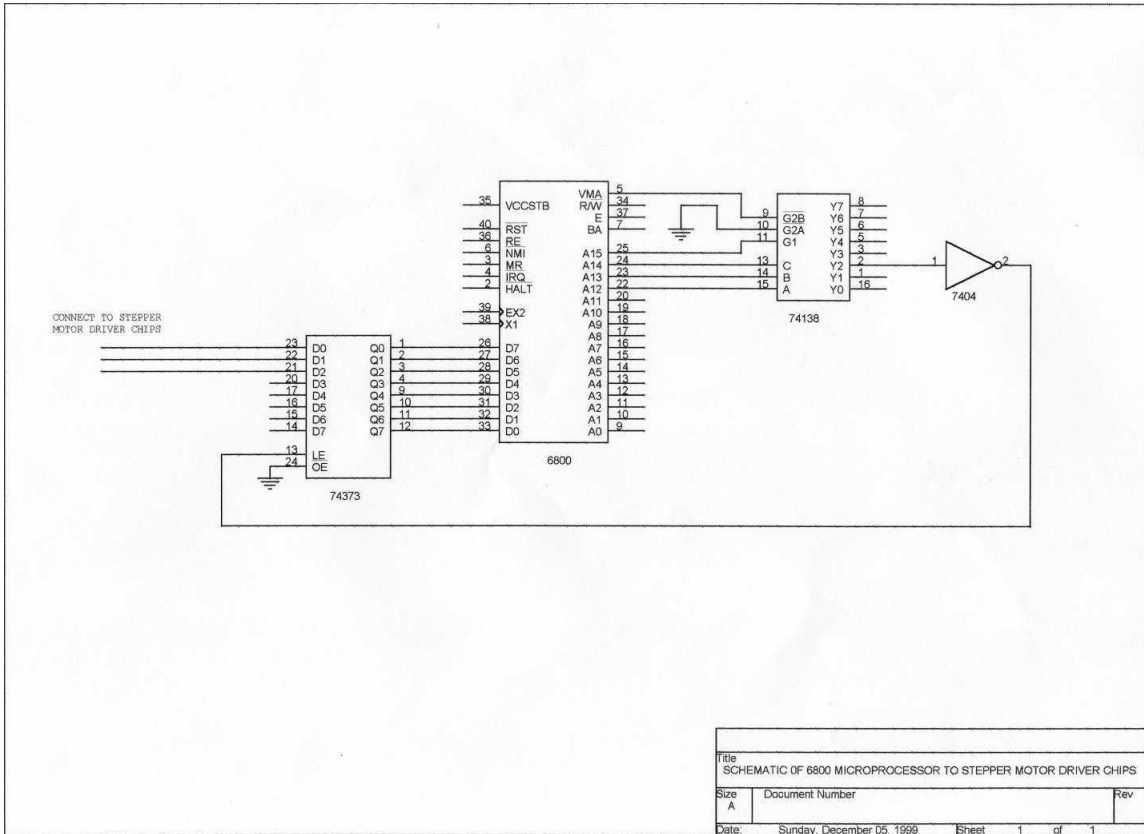


Figure 5: Schematic of 6800 microprocessor to stepper motor driver chip

This saves time when one is debugging programs. The programs could be saved on floppy disk, modified as needed, and then downloaded to the micro-controller memory. The program had to be manually entered as is the case with 6800 trainer. The data bus of the MC68HC912B32 is 16 bits wide, twice the width of the 6800 chip. This provides the extra data bits required for controlling all three stepper motors. The micro-controller pins are divided into two I/O ports A and B. The data bus is accessible through port A and port B, as shown in Table 4.

Initially to demonstrate the controlling performance of MC68HC912B32, a program as shown in Table 5 was developed. Equal performance was obtained. It was noticed speed of micro-controller was much faster than 6800, so the delay subroutine was used.

Table 4: Micro-controller data bus pin numbers

Port	Pin Number	Data Bus Bit
A	0	D ₀
A	1	D ₁
A	2	D ₂
A	3	D ₃
A	4	D ₄
A	5	D ₅
A	6	D ₆
A	7	D ₇
B	0	D ₈
B	1	D ₉
B	2	D ₁₀
B	3	D ₁₁
B	4	D ₁₂
B	5	D ₁₃
B	6	D ₁₄
B	7	D ₁₅

To emulate the full power of the MC68HC912B32 micro-controller, a group of switches were used as shown in Figure 6 to manually send the control signals from the micro-controller to each stepper motor driver circuit. This enabled manual control of each axis of the milling machine. Only two data lines from the micro-controller were required when using the emulation switches. There was a series of switches that controls the direction of rotation for each stepper motor by reversing the phase A and phase B inputs of the driver chips. Also, another series of switches turned the individual motors on by connecting the driver chips inputs IN0 and IN1 to zero volts. There was also a switch to select whether the input to the motor drivers was coming from the hand controller or directly from the micro-controller.

Table 5: Program for MC68HC912B32

0000	[01]	86	FF		01		LDAA	#\$FF
0002	[02]	5A	03		02		STAA	\$0003
0004	[01]	86	01		03	START:	LDAA	#\$01
0006	[02]	5A	01		04		STAA	\$0001
0008	[02]	07	14		05		BSR	DELAY
000A	[01]	86	03		06		LDAA	#\$03
000C	[02]	5A	01		07		STAA	\$0001
000E	[02]	07	0E		08		BSR	DELAY
0010	[01]	86	02		09		LDAA	#\$02
0012	[02]	5A	01		10		STAA	\$0001
0014	[02]	07	08		11		BSR	DELAY
0016	[01]	86	00		12		LDAA	#\$00
0018	[02]	5A	01		13		STAA	\$0001
001A	[02]	07	02		14		BSR	DELAY
001C	[01]	20	E6		15		BRA	START
001E	[02]	CE	0F	FF	16	DELAY:	LDX	#\$0FFF
0021	[01]	A7			17	LOOP:	NOP	
0022	[01]	09			18		DEX	
0023	[02]	8E	00	00	19		CPX	#\$0000
0026	[01]	26	F9		20		BNE	LOOP
0028	[05]	3D			21		RTS	
					22			
					23			
SYMBOL TABLE								
DELAY	001E							
LOOP	0021							
START	0004							

To cut path with a micro-controller program, there were nine data lines used as inputs to the driver chips. Table 6 shows those connections.

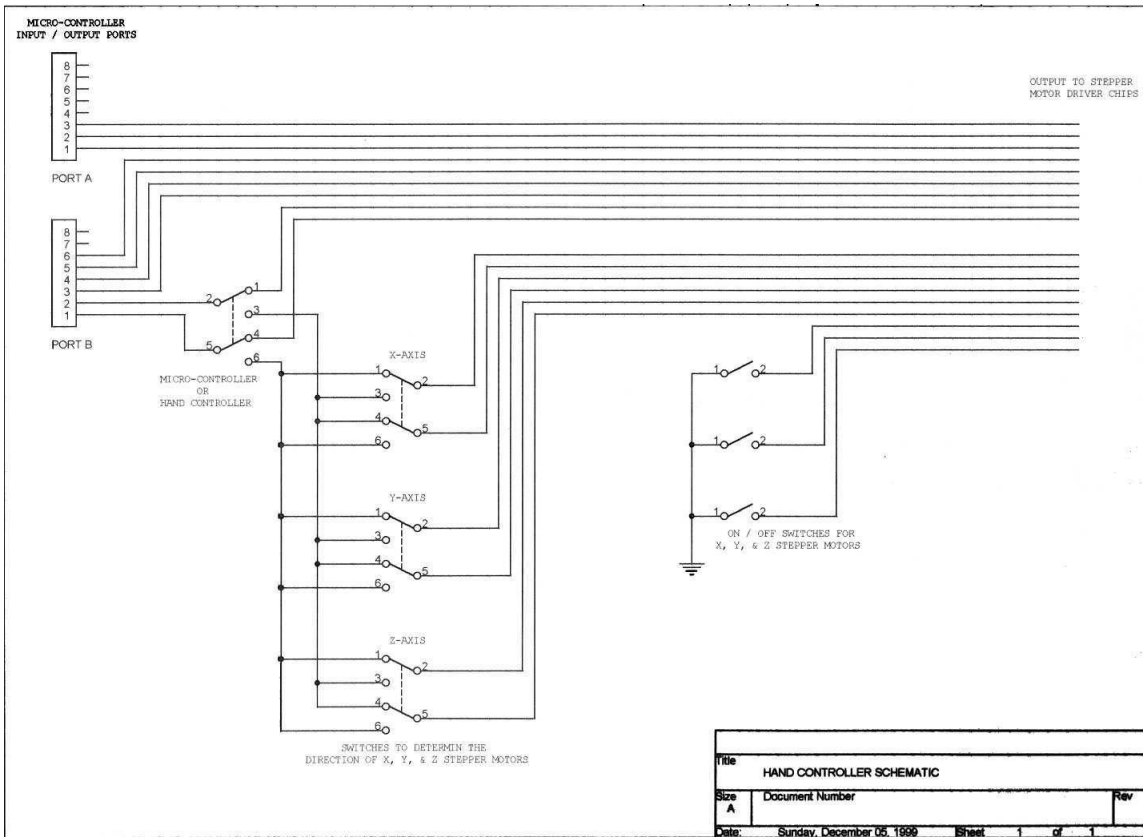


Figure 6: Circuit for manual control of milling machine

Table 6: Connection between micro-controller and driver chips

Micro-controller data line	Stepper motor controlled	Driver chip inputs
D ₀	X-axis	Phase A
D ₁		Phase B
D ₂		IN0A, IN1A, IN0B, IN1B
D ₃	Y-axis	Phase A
D ₄		Phase B
D ₅		IN0A, IN1A, IN0B, IN1B
D ₈	Z-axis	Phase A
D ₉		Phase B
D ₁₀		IN0A, IN1A, IN0B, IN1B

III. Example

The program shown in Table 7 represents a section of a program for the 68HC912 micro-

controller used to drive milling machine to cut a desired path of a TIC-TAC-TOE Board. This section shows the code required to drive the stepper motors first in the positive Y-axis and then in the negative X-axis.

Table 7: Program to Cut TIC-TAC-TOE Board

0000	[01]	86	FF		001		LDA	#\$FF
0002	[02]	5A	03		002		STAA	\$0003
0004	[02]	CD	03	20	003		LDY	#\$0320
0007	[02]	07	5F		004	ONE	BSR	Y(+1)
0009	[02]	8D	00	00	005		CPY	#\$0000
000C	[01]	03			006		DEY	
000D	[01]	26	F8		007		BNE	ONE
000F	[02]	CD	09	F6	008		LDY	#\$09F6
0012	[02]	07	6D		009	TWO	BSR	X(-1)
0014	[02]	8D	00	00	010		CPY	#\$0000
0017	[01]	03			011		DEY	
0018	[01]	26	F8		012		BNE	TWO
001A	[02]	CD	03	20	013		LDY	#\$0320
0068	[01]	86	0C		049	Y(+1)	LDA	#\$0C
006A	[02]	5A	01		050		STAA	\$0001
006C	[02]	07	5E		051		BSR	SPEED
006E	[01]	86	1C		052		LDA	#\$1C
0070	[02]	5A	01		053		STAA	\$0001
0072	[02]	07	58		054		BSR	SPEED
0074	[01]	86	14		055		LDA	#\$14
0076	[02]	5A	01		056		STAA	\$0001
0078	[02]	07	52		057		BSR	SPEED
007A	[01]	86	04		058		LDA	#\$04
007C	[02]	5A	01		059		STAA	\$0001
007E	[02]	07	4C		060		BSR	SPEED
0080	[05]	3D			061		RTS	
0081	[01]	86	21		062	X(-1)	LDA	#\$21
0083	[02]	5A	01		063		STAA	\$0001
0085	[02]	07	45		064		BSR	SPEED
0087	[01]	86	23		065		LDA	#\$23
0089	[02]	5A	01		066		STAA	\$0001
008B	[02]	07	3F		067		BSR	SPEED
008D	[01]	86	22		068		LDA	#\$22
008F	[02]	5A	01		069		STAA	\$0001
0091	[02]	07	39		070		BSR	SPEED
0093	[01]	86	20		071		LDA	#\$20
0095	[02]	5A	01		072		STAA	\$0001
0097	[02]	07	33		073		BSR	SPEED
0099	[05]	3D			074		RTS	
00CC	[02]	CE	0F	FF	101	SPEED	LDX	#\$0FFF
00CF	[01]	A7			102	DELAY	NOP	
00D0	[01]	09			103		DEX	
00D1	[02]	8E	00	00	104		CPX	#\$0000
00D4	[01]	26	F9		105		BNE	DELAY
00D6	[05]	3D			106		RTS	

SYMBOL	
TABLE	
ONE	0007
TWO	0012
SPEED	00CC
DELAY	00CF
Y(+1)	0068
X(-1)	0081

IV. Conclusion

Controlling of the milling machine with both the microprocessor and micro-controller was achieved. For programming simplicity, four step increments were used. This gives positional control of 1/800 of an inch. There are some improvements that could be made in the future. The use of power MOSFET's to drive the stepper motors could allow them to be controlled directly from the micro-controller. Adding electronic counters on each stepper motor with a feedback loop could allow the position of the cutter head to be monitored and automatically controlled. This project provided the experiences of using, controlling, and interfacing stepper motor with microcomputer for the student.

Bibliographies

1. Floyd, T.L., Digital Fundamentals, Merrill Publication, OH, 1998.
2. Skroder, J.C., Using M68HC11 Microcontroller, Prentice Hall, N.J., 1997.
3. Kheir, M., The M68HC11 Microcontroller: Applications in Control, Instrumentation, and Communication, Prentice-Hall, N.J., 1997.
4. Motorola Inc., Motorola MC68HC912B32 Reference Manual, Phoenix, AZ, 1998.

MOHAMMAD FOTOUHI

Dr. Fotouhi is a Professor of electrical engineering technology at University of Maryland Eastern Shore. He received his Ph.D. in power System Engineering from University of Missouri-Rolla, M.S. from Oklahoma State University and B.S. from Tehran Polytechnic College. He has been conducting a practical research on the growth and characterization of the dilute magnetic semiconductor since 1985. He is a member of Eta Kappa Nu Honor Society. He was chairman of Student and Industry Relation and Host Committee member of IEEE Conference on Power Systems Computer Application in 1991. He also was chairman of Student Relation and Host Committee member of the IEEE Power Society Winter Meeting in 1996.

ALI EYDGAHI

Dr. Eydgahi is an Associate Professor of electrical engineering at University of Maryland Eastern Shore. He received his Ph.D. and M.S. in Electrical and Computer Engineering from Wayne State University. Since 1986 and prior to joining University of Maryland Eastern Shore he has been with the State University of New York, University of Tehran, Wayne County Community College, and Oakland University. Dr. Eydgahi is recipient of the Dow Outstanding Young Faculty Award from American Society for

Engineering Education in 1990, and the Silver Medal for outstanding contribution from International Conference on Automation in 1995. He is the ASEE Campus Representative at UMES and has served as a regional and chapter chairman of IEEE and SME in New York. He also has served as a session chair and a member of scientific and international committees for many international conferences. He has published more than seventy papers in refereed international and national journals and conference proceedings.

JOSHUA WAGNER

Mr. Wagner graduated from the University of Maryland Eastern Shore with a Bachelor of Science degree in Electronic Engineering Technology in 1998. While a student at the University of Maryland Eastern Shore, he was on the dean's list for 4 semesters. He joined Airpax Corporation in Cambridge, Maryland in June 1998 as a Product Engineer. He has recently been transferred to the Tool Design Engineering department where his team responsibilities include the design of punch and die sets, plastic molds, PLC and micro-controller controlled automated machines, electrical analyzer), welding nests, SPC checking fixtures, ultrasonic welding nests and horns, pneumatic and hydraulic clamps and jigs, and assorted production aid tools.