บรรณานุกรม

กฤษดา วิศวธีรานนท์. ฟังก์ชั่นชาร์ต ภาษาใหม่สำหรับการโปรแกรม PLC. วารสารเทคนิค ฉบับที่ 85, 2535.

--------. PC ตัวควบคุมซีเคว้นซ์. หลักการทำงานและการประยุกต์. กรุงเทพมหานคร : บุญชัยวิศวกรรม, 2534.

ธานินทร์ ถาวรศาสนวงศ์ และ ทินกร ดุ๊ก. การอินเตอร์เฟส IBM PC. กรุงเทพมหานคร : ฟิสิกส์เซ็นเตอร์การพิมพ์.

Charles Easteal and Gordon Davies. Software engineering. Analysis and design. Berkshire, England : McGRAW-HILL Book Company (UK) Ltd., 1989.

Epson. LQ-850 and LQ-1050 24 pins dot matrix printers. Reference guide. Nagono, Japan : 1989.

Gilles Michel. Programmable Logic Controller. Architecture and Application. West Sussex, England : John Willey & Sons Ltd., 1990.

Hewlett Packard. LaserJet III printer technical reference manual. U.S.A. : 1990.

----. LaserJet III printer user's manual. West Germany : 1990.

IBM. PC/AT Technical Reference. Florida : 1984.

International Electrotechnical Commission. Preparation of Function Charts for Control System. International Standard IEC848. 1st ed. Suisse, France : Bureau Central de la Commission Electrotechnique Internationale, 1988.

Michale Lloyd. GRAFCET : Graphical Function Chart Programming. TELEMACHANIQUE.

OMRON. OMRON user's manual. Host link unit (for SYSMAC C Series). Japan : OMRON Tateisi electronics co.

----. OMRON user's manual. Programmable controller model SYSMAC C-500. Japan : OMRON Tateisi electronics co.

SIEMENS. SIEMATIC S5 GRAPH 5 Introduction.

TELEMACHANIQUE. TSXT607 : Programming Technical User's Manual. Book 3 : Grafcet language, 1987.

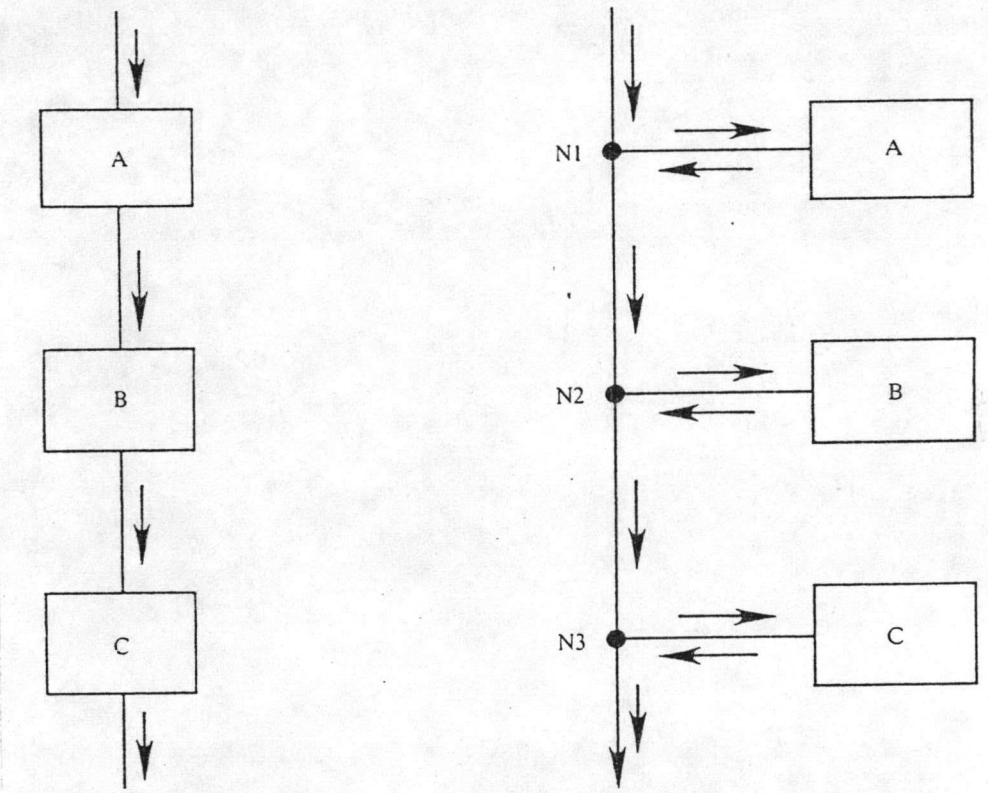
ภาคผนวก

ภาคผนวก ก

Design Structure Diagram

เกริ่น

DSD[1] เป็นรูปแบบกราฟฟิก (Graphical notation) ซึ่งใช้เขียนอธิบายขั้นตอนการทำงานของโปรแกรม โดยดัดตอนมาจากส่วนหนึ่งของ British Standard Specification (BSI, 1987) DSD มีโครงสร้างในลักษณะที่เรียกว่า "Fall-back diagram" ในรูปที่ ข-1 โครงสร้างทั้ง 2 แบบให้ผลการทำงานเหมือนกัน แบบ Fall-back จะนำสเต็ปการทำงานแยก



(a) Normal rules (b) Fall-back rules

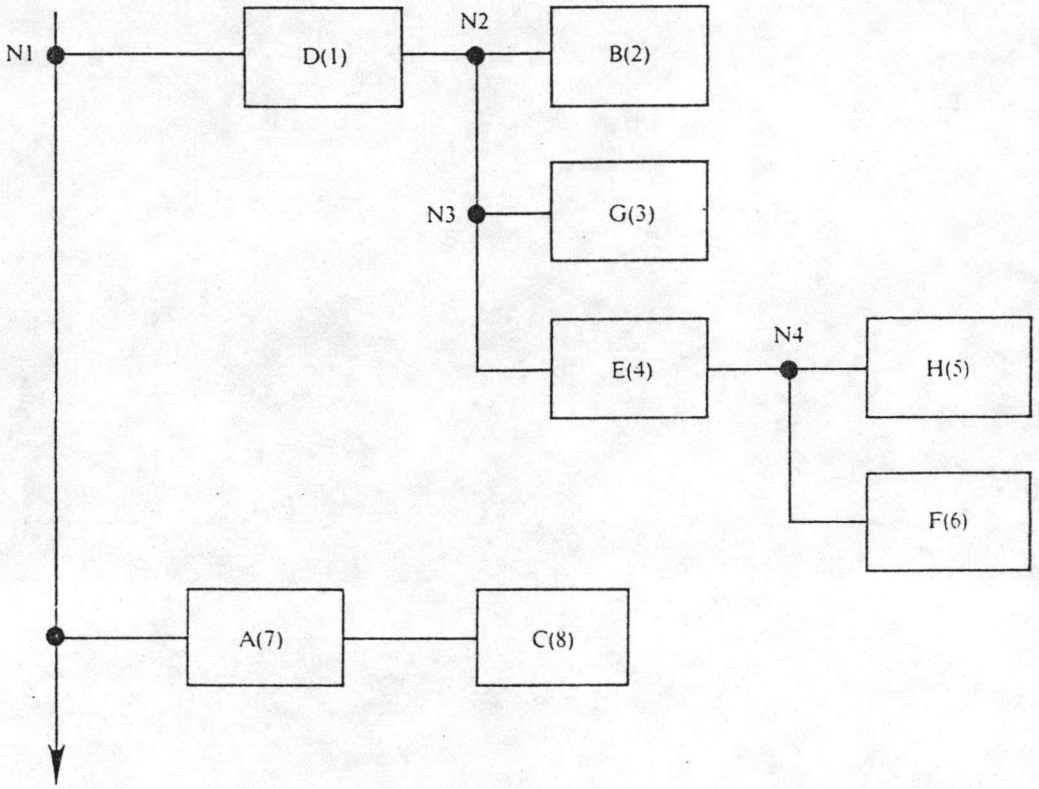
รูปที่ ข-1 : เปรียบเทียบโครงสร้างแบบปกติกับแบบ Fall-back

ออกจากเส้นในแนวดิ่ง และลากเส้นเชื่อมต่อกัน จุดเชื่อมต่อเรียกว่า "โหนด (Node)" สามารถแสดงเส้นทางการทำงานได้ดัง

---

1. Charles Eastel and Gordon Davies. Software Engineering. Analysis and design. Berkshire, England : McGRAW-HILL Book Company (UK) Ltd., 1989.

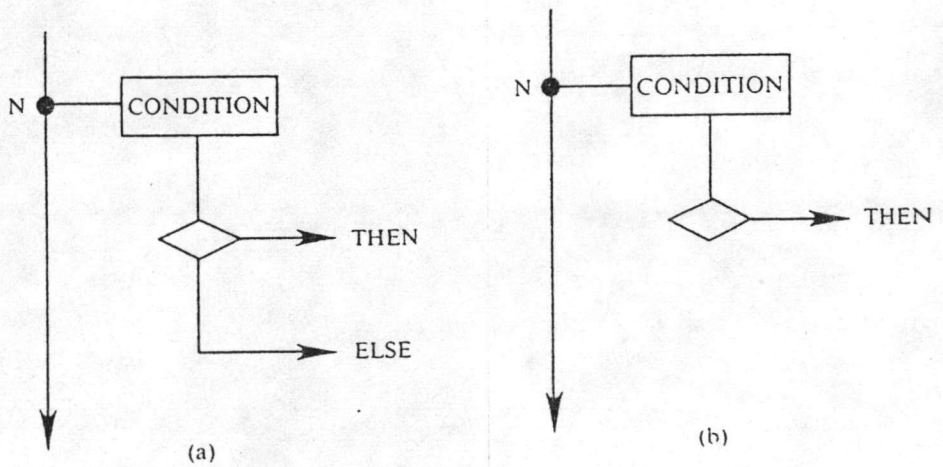
ลูกศร กล่าวคือ การทำงานจะทำจากบนลงล่าง เมื่อถึงโหนดจะแยกไปทางขวาและทำงานจนครบทุกสเต็ปภายใน จากนั้น การควบคุมจะย้อนกลับตามเส้นทางเดิม แต่ในคราวนี้สเต็ปต่างๆ ไม่ต้องทำงาน (Fall-back) ด้วยหลักการเช่นนี้ จึงทำให้การเขียนอธิบายในแบบ DSD สามารถอธิบายได้ชัดเจนกว่าแบบปกติ โดยเฉพาะการอธิบายงานย่อยในงานใหญ่ และสามารถสอดแทรกเพิ่มเติมปรับปรุงแก้ไขได้โดยง่าย ในรูปที่ ข-2 เป็นการยกตัวอย่าง DSD ตัวเลขในวงเล็บแสดงลำดับที่ในการทำงาน



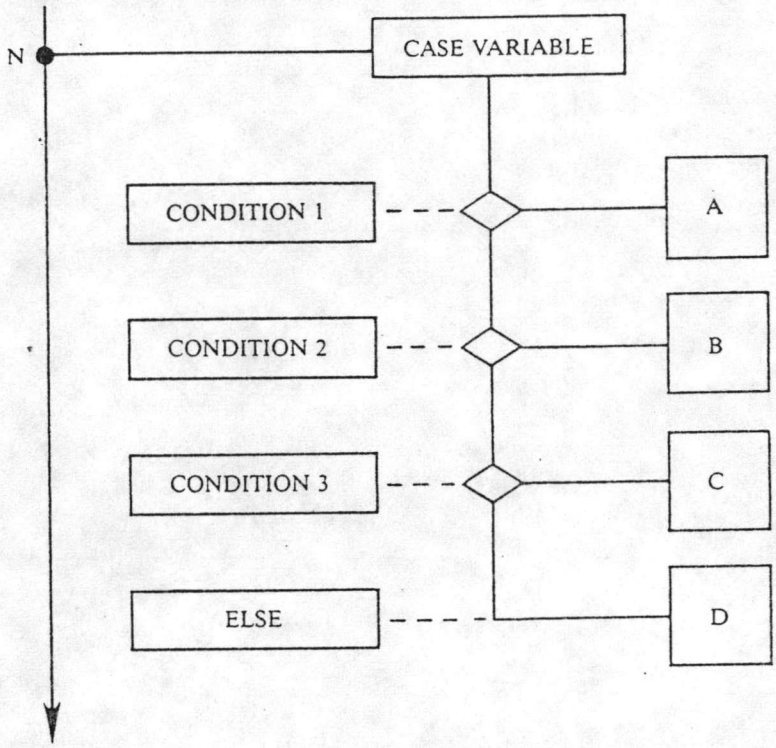
รูปที่ ข-2 : ตัวอย่าง DSD

องค์ประกอบต่างๆ

1. IF-THEN-ELSE

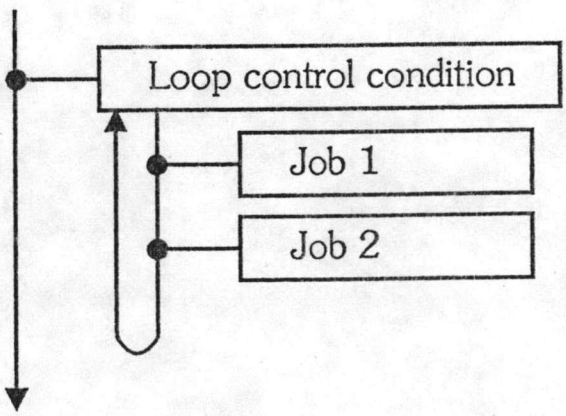


2. CASE

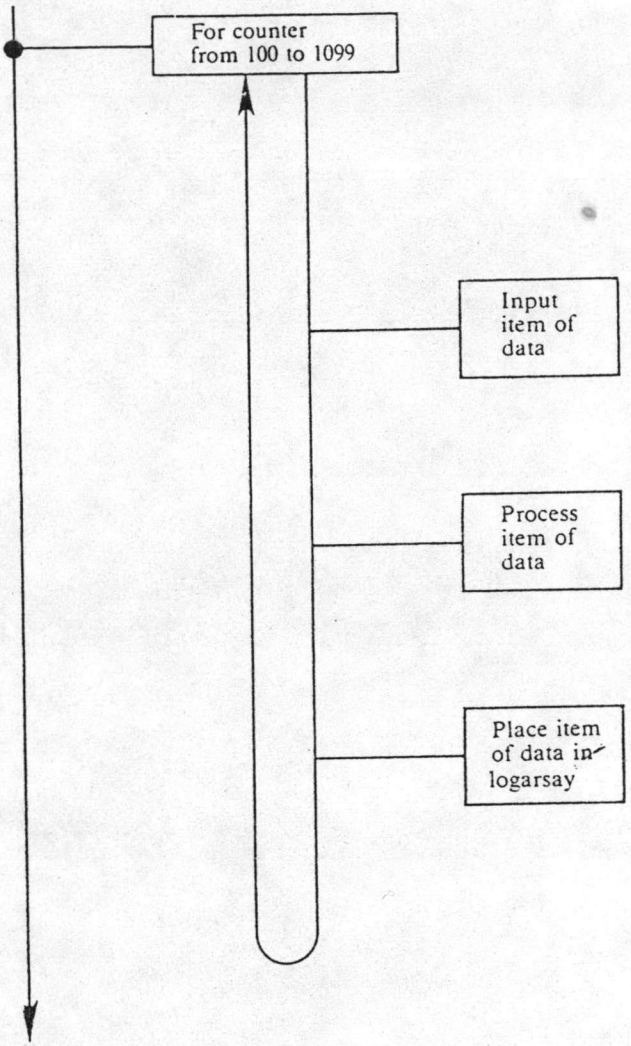


3. LOOP

3.1 แบบปกติ

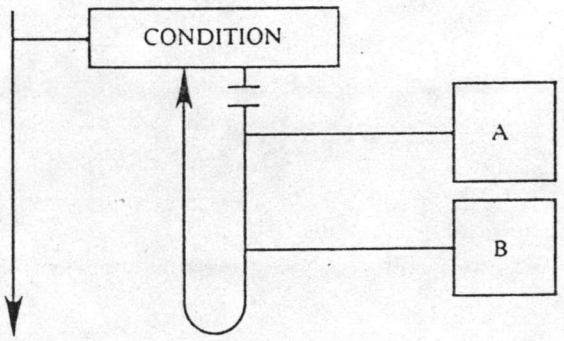


ตัวอย่าง

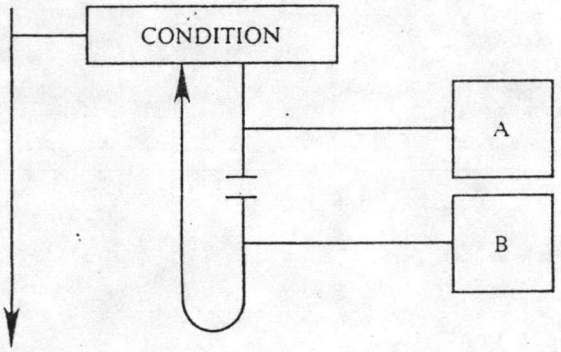


3.2 แบบตรวจสอบเงื่อนไข

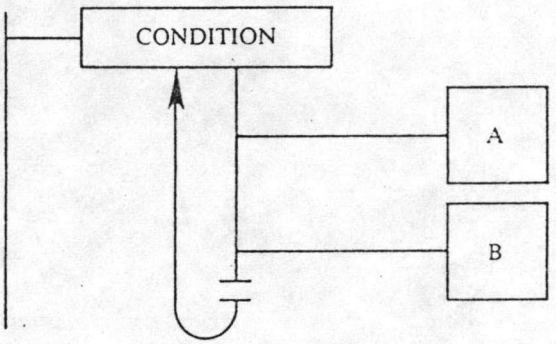
3.2.1 ตรวจสอบเงื่อนไขก่อน



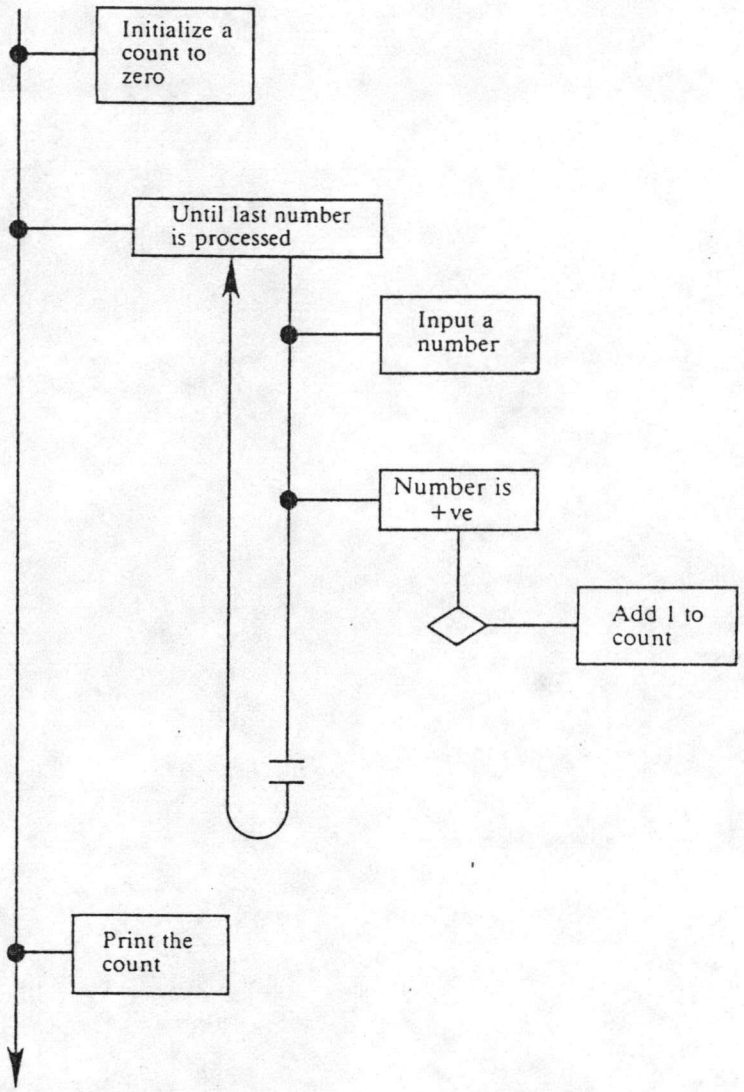
3.2.2 ตรวจสอบเงื่อนไขภายใน



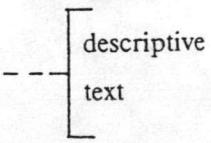
3.2.3 ตรวจสอบเงื่อนไขทีหลัง



ตัวอย่าง



4. คำอธิบาย : ในกรณีต้องการเขียนคำอธิบายเพิ่มเติม



ภาคผนวก ข

ตารางรหัสภายในของออมร่อน

INSTRUCTIONS IN MACHINE CODE FOR C-SERIES (C120, C250, C500)

No. of byte / Instruction	1	2	3	4	5	6	7	8	9	10	Note
LD I,II	LSRA	ORA	EXT								
	44	BA	OPERAND *1)		= 00(bit)00(CH-No)						
LD NOT I,II	LSRA	ORA	EXT								II:except TR
	55	BA	OPERAND *2)		= 00(bit)01(CH-No)						
AND I,II	ANDA	EXT									II:except TR
	B4	OPERAND *1)									
AND NOT I,II	ANDA	EXT									II:except TR
	B4	OPERAND *2)									
OR I,II	ORA	EXT									II:except TR
	BA	OPERAND *1)									
OR NOT I,II	ORA	EXT									II:except TR
	BA	OPERAND *2)									
AND LD	LSLA	BCS	02	ANDA	%$7F						
	48	25	02	84	7F						
OR LD	LSLA	BCC	02	ORA	%$80						
	48	24	02	8A	80						
OUT I,II	LDB	EXT		STA	EXT						II:only for TR
	F6	OPERAND *1)		B7	OPERAND *1)						
OUT NOT I	COMA	LDB	EXT	STA	EXT	COMA					
	43	F6	OPERAND *1)	B7	OPERAND *1)	43					

No. of byte / Instruction	1	2	3	4	5	6	7	8	9	10	Note
KEEP I	LDB	EXT		STA	EXT						
	F6	OPERAND *2)		B7	OPERAND *2)						
TIM 0...127	LDB	% 0-127 JSR		ADDR		DC	SET VALUE				III:exceptT/C,DM
III,IV	C6	BINARY→	BD	C0	15	*000000	CH/CONSTANT				CONSTANT:0-9999
CNT 0...127	LDB	% 0-127 JSR		ADDR		DC	SET VALUE				ditto
III,IV	C6	BINARY→	BD	C0	18	*000000	CH/CONSTANT				
IL	INCB	JSR	ADDR								
	5C	BD	C0	03							
ILC	INCB	JSR	ADDR								
	5C	BD	C0	06							
MOV	JSR	8BIT OFFSET		DC	SET A		SET B				SETB:IIIexcept T/C
SETA:III,IV,V SETB:III,V	AD	D8	04	* *0000	CH/CONSTANT		CH				CONSTANT:0-FFFF
MVN	JSR	8BIT OFFSET		DC	SET A		SET B				ditto
SETA:III,IV,V	AD	D8	06	* *0000	CH/CONSTANT		CH				
FAL 0-99	LDX	%$ 00-99		JSR	ADDR						
	8E	00	BCD	BD	C0	27					
FALS 1-99	LDX	%$ 01-99		JSR	ADDR						
	8E	00	BCD	BD	C0	2A					
CMP	JSR	8BIT OFFSET		DC	SET A		SET B				
SETA:III,IV,V SETB III,IV,V	AD	D8	02	* *0000	CH/CONSTANT		CH/CONSTANT				
SFT	JSR	8BIT OFFSET		DC	SET A		SET B				III:except T/C,DM
SETA: III SETB: III	AD	D8	00	00000000	CH/CONSTANT		CH/CONSTANT				

No. of byte / Instruction	1	2	3	4	5	6	7	8	9	10	Note
DIFU I	LDX	%EXT		JSR	ADDR						
	8E	OPERAND *1)		BD	C0	21					
DIFD I	LDX	%EXT		JSR	ADDR						
	8E	OPERAND *1)		BD	C0	24					
END	JMP	ADDR									
	7E	C0	00								
CNTR 0-127	LDB	0-127	JSR	ADDR		DC	SET VALUE				III:except T/C,DM
SET: III,IV	C6	BINARY	BD	C0	1B	*000000	CH/CONSTANT				CONSTANT:0-9999
JMP	INCB	JSR	ADDR								
	5C	BD	C0	09							
JME	INCB	JSR	ADDR								
	5C	BD	C0	0C							
BIN	JSR	8BIT OFFSET		DC	SET A		SET B				SET B:
SETA: III,V SETB: III,V	AD	D8	08	* *0000	CH		CH				III:except T/C
BCD	JSR	8BIT OFFSET		DC	SET A		SET B				III:except T/C
SETA/B:III,V	AD	D8	0A	* *0000	CH		CH				
ADD	JSR	ADDR		DC	SET A		SET B		SET C		SET C
SETA:III,IV,V SETB:III,IV,V SETC:III,V	BD	C0	2D	* * *00	CH/CONSTANT		CH/CONSTANT		CH		III:except T/C CONSTANT:0-9999
SUB	JSR	ADDR		DC	SET A		SET B		SET C		ditto
SETA:III,IV,V SETB:III,IV,V SETC:III,V	BD	C0	30	* * *00	CH/CONSTANT		CH/CONSTANT		CH		
MUL	JSR	ADDR		DC	SET A		SET B		SET C		ditto
SETA:III,IV,V SETB:III,IV,V SETC:III,V	BD	C0	33	* * *00	CH/CONSTANT		CH/CONSTANT		CH		

No. of byte / Instruction	1	2	3	4	5	6	7	8	9	10	Note
DIV	JSR	ADDR		DC	SET A		SET B		SET C		SETC
SETA:III,IV,V SETB:III,IV,V SETC:III,V	BD	C0	36	* * *00	CH/CONSTANT		CH/CONSTANT		CH		III:except T/C
ANDW	JSR	ADDR		DC	SET A		SET B		SET C		ditto
SETA:III,IV,V SETB:III,IV,V SETC:III,V	BD	C0	39	* * *00	CH/CONSTANT		CH/CONSTANT		CH		
ORW	JSR	ADDR		DC	SET A		SET B		SET C		ditto
SETA:III,IV,V SETB:III,IV,V SETC:III,V	BD	C0	3C	* * *00	CH/CONSTANT		CH/CONSTANT		CH		
STC	INCB	JSR	ADDR								
	5C	BD	C0	0F							
CLC	INCB	JSR	ADDR								
	5C	BD	C0	12							
NOP	CMPX	%$8C8C									
	8C	8C	8C								
TIMH 0-127	LDB	%0-127	JSR	ADDR		DC	SET VALUE				III:except T/C,DM
SET: III,IV	C6	BINARY	BD	C0	1E	*000000	CH/CONSTANT				CONSTANT:0-9999
MNEMONIC	DECB	JSR	8BIT OFFSET		DC	SET A					SETA
SETA:III,V	5A	AD	D8	(*3)	*000000	CH					III:except T/C
	*3) INC=10,DEC=12,ASL=14,ASR=16,ROL=18,ROR=1A,COM=0E										
WSFT	JSR	8BIT OFFSET		DC	SET A		SET B				SETA/B
SETA/B,III,V	AD	D8	0C	* *0000	CH		CH				III:except T/C

No. fo byte / Instruction	1	2	3	4	5	6	7	8	9	10	Note
MNEMONIC	JSR	8BIT OFFSET		DC	SET A		SET B		SET C		SET C
SETA:III,IV,V SETB:III,IV,V SETC:III,V	BD	C0	*4)	* * *00	CH/CONSTANT		CH/CONSTANT		CH		III:except T/C CONSTANT:0-FFFF
	*4) XORW=3F, XNRW=42										
FUN(70-99)	JSR	ADDR		DC	SET A		SET B		SET C		
SETA:III,IV,V	BD	C0	*5)	* * *00	CH/CONSTANT		CH/CONSTANT		CH/CONSTANT		

*5) FUN-No.=ADDR as follows:

FUN	ADDR	FUN	ADDR	FUN	ADDR
70	45	80	63	90	81
71	48	81	66	91	84
72	4B	82	69	92	87
73	4E	83	6C	93	8A
74	51	84	6F	94	8D
75	54	85	72	95	90
76	57	86	75	96	93
77	5A	87	78	97	96
78	5D	88	7B	98	99
79	60	89	7E	99	9C

TYPE OF OPERAND

Type	0	0	BIT-No. x4	x2	x1	*7) AND *6)		CH-No:							BIT x8	CONTROL-PORT	Note Start ADDR	x	8421
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
TYPE I (CONTACT)																				
C-I/O,Aux.relay	0	0	4	2	1	)(	)	0	0	x	x	x	x	x	x	8	BIT	000	0-63	0-15
LR	0	0	4	2	1	)(	)	0	1	0	x	x	x	x	x	8	BIT	080	0-31	0-15
HR	0	0	4	2	1	)(	)	0	1	1	x	x	x	x	x	8	BIT	0C0	0-31	0-15
TYPE II (CONTACT)																				
TIM	0	0	0	0	0	)(	)	1	0	x	x	x	x	x	x	x	BIT	100	0-127	
CNT	0	0	0	0	1	)(	)	1	0	x	x	x	x	x	x	x	BIT	100	0-127	
TR	0	0	4	2	1	0	0	0	0	1	1	1	1	1	1	0	BIT	07E		0-7
TYPE III (CHANNEL)																				
C-I/O,Aux.relay	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	0	BIT	000	0-63	
LR	0	0	0	0	0	0	0	0	1	0	x	x	x	x	x	0	BIT	080	0-31	
HR	0	0	0	0	0	0	0	0	1	1	x	x	x	x	x	0	BIT	0C0	0-31	
TIM/CNT	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	0	BIT	180	0-127	
DM	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	0	BIT	280	0-511	
TYPE IV (CONSTANT)																				
DECIMAL(4 digits)	10^3				10^2				10^1				10^0				BIT	0-9999	0-9999	
HEXA-DECIMAL (4 digits)	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	BIT	0-FFFF	0-FFFF	
TYPE V (INDIRECT)																				
DM	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	0	BIT	280	0-511	

Note: *6) NOT=1, KEEP=1 by coil o' Keep relay *7) AND=1

ภาคผนวก ค

การสื่อสารทางพอร์ทอนุกรมของ IBM PC

Controller-Accessible Registers

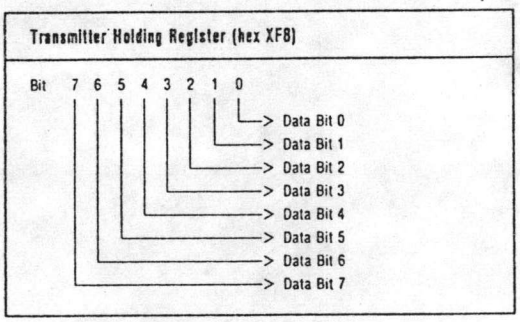
The controller has a number of accessible registers. The system programmer may gain access to or control any of the controller registers through the microprocessor. These registers are used to control the controller's operations and to transmit and receive data. The X in the register address determines the the port selected; 3 is for port 1 and 2 is for port 2.

Specific registers are selected according to the following figure:

I/O Address	Register Selected	DLAB State
XF8	TX buffer	0 (write)
XF8	RX buffer	0 (read)
XF8	Divisor Latch LSB	1
XF9	Divisor Latch MSB	1
XF9	Interrupt Enable Register	0
XFA	Interrupt Identification Register	
XFB	Line Control Register	
XFC	Modem Control Register	
XFD	Line Status Register	
XFE	Modem Status Register	
XFF	Reserved	

Controller-Accessible Registers

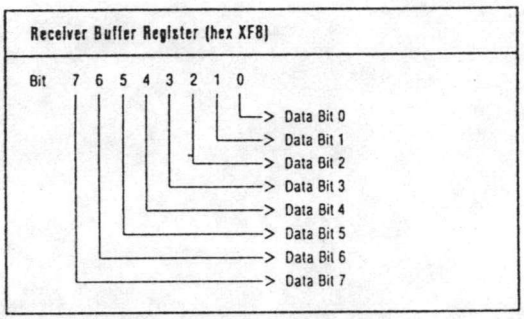
Transmitter Holding Register (Hex XF8): The transmitter holding register (THR) contains the character to be sent.



Transmitter Holding Register

Bit 0 is the least-significant bit and the first bit sent serially.

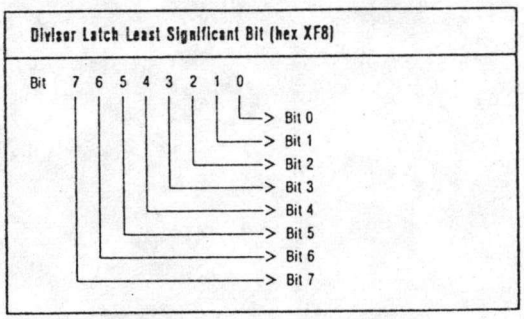
Receiver Buffer Register (Hex XF8): The receiver buffer register (RBR) contains the received character.



Receiver Buffer Register

Bit 0 is the least-significant bit and the first bit received serially.

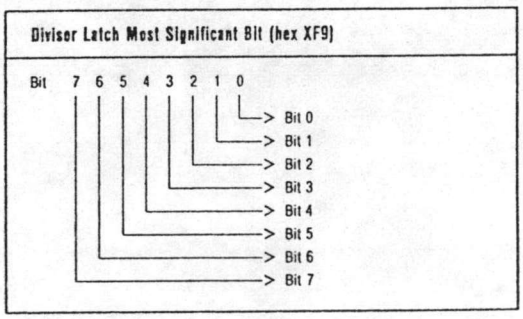
Divisor Latch LSB (Hex XF8)



Divisor Latch Least Significant Bit

Information about this register may be found under "Programmable Baud Rate Generator" later in this section.

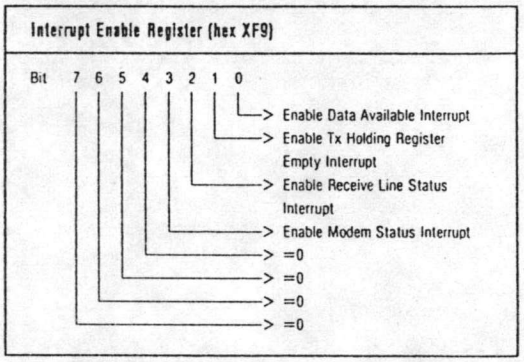
Divisor Latch MSB (Hex XF9)



Divisor Latch Most Significant Bit

Information about this register may be found under "Programmable Baud Rate Generator" later in this section.

Interrupt Enable Register (Hex XF9): This 8-bit register allows the four types of controller interrupts to separately activate the 'chip-interrupt' (INTRPT) output signal. The interrupt system can be totally disabled by resetting bits 0 through 3 of the interrupt enable register (IER). Similarly, by setting the appropriate bits of this register to logical 1, selected interrupts can be enabled. Disabling the interrupt system inhibits the 'IER' and the active 'INTRPT' output from the chip. All other system functions operate normally, including the setting of the line-status and modem-status registers.



Interrupt Enable Register

Bit 0 When set to logical 1, enables the received-data-available interrupt.

Bit 1 When set to logical 1, enables the transmitter-holding-register-empty interrupt.

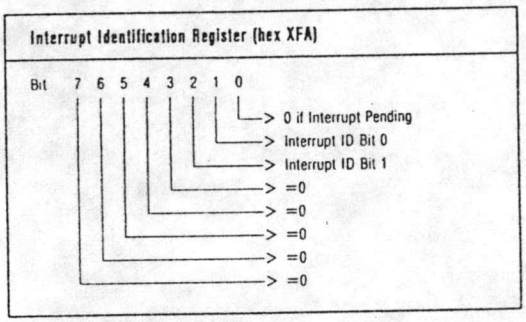
Bit 2 When set to logical 1, enables the receiver-line-status interrupt.

Bit 3 When set to logical 1, enables the modem-status interrupt.

Bits 4–7 These four bits are always logical 0.

Interrupt Identification Register (Hex XFA): The controller has an on-chip interrupt capability that makes communications possible with all of the currently popular microprocessors. In order to minimize programming overhead during data character transfers, the controller prioritizes interrupts into four levels: receiver line status (priority 1), received data ready (priority 2), transmitter holding register empty (priority 3), and modem status (priority 4).

Information about a pending prioritized interrupt is stored in the interrupt identification register (IIR). (See the figure "Interrupt Control Functions," later.) The IIR, when addressed during chip-select time, stops the pending interrupt with the highest priority, and no other interrupts are acknowledged until the processor services that particular interrupt.



Interrupt Identification Register

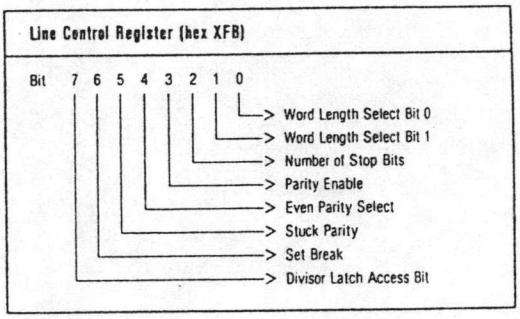
Bit 0 This bit can be used in either hard-wired, prioritized, or polled conditions to indicate if an interrupt is pending. When bit 0 is logical 0, an interrupt is pending, and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is logical 1, no interrupt is pending, and polling (if used) continues.

Bits 1–2 These two bits identify the pending interrupt that has the highest priority interrupt pending, as shown in the following figure.

Bits 3–7 These five bits are always logical 0.

Interrupt ID Register			Interrupt Set And Reset Functions			
Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	–	None	None	–
1	1	0	Highest	Receiver Line Status	Overrun Error or Parity Error or Framing Error or Break Interrupt	Reading the Line Status Register
1	0	0	Second	Received Data Available	Receiver Data Available	Reading the Receiver Buffer Register
0	1	0	Third	Trans-mitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR (if source of in-terrupt) or writing into the THR
0	0	0	Fourth	Modem Status	Clear to Send or Data Set Ready or Ring Indicator or Received Line Signal Detect	Reading the Modem Status Register

Line-Control Register (Hex XFB): The system programmer specifies the format of the asynchronous data communications exchange through the line control register. In addition to controlling the format, the programmer may retrieve the contents of the line control register for inspection. This feature simplifies system programming and eliminates the need to store line characteristics separately in system memory.



Line Control Register

Bits 0, 1 These two bits specify the number of bits in each serial character that is sent or received. The encoding of bits 0 and 1 is as follows:

Bit 1	Bit 2	Word Length (Bits)
0	0	5
0	1	6
1	0	7
1	1	8

Word Length

Bit 2 This bit specifies the number of stop bits in each serial character that is sent or received. If bit 2 is a logical 0, one stop bit is generated or checked in the data sent or received. If bit 2 is logical 1 when a 5-bit word length is selected through bits 0 and 1, 1-1/2 stop bits are generated or checked. If bit 2 is logical 1 when either a 6-, 7-, or 8-bit word length is selected, two stop bits are generated or checked.

Bit 3 This bit is the parity-enable bit. When bit 3 is logical 1, a parity bit is generated (transmit data) or checked (receive data) between the last data word and stop bit of the serial data. (The parity bit is used to produce an even or odd number of 1's when the data-word bits and parity bit are summed.)

Bit 4 This bit is the even-parity-select bit. When bit 3 is a logical 1 and bit 4 is a logical 0, an odd number of

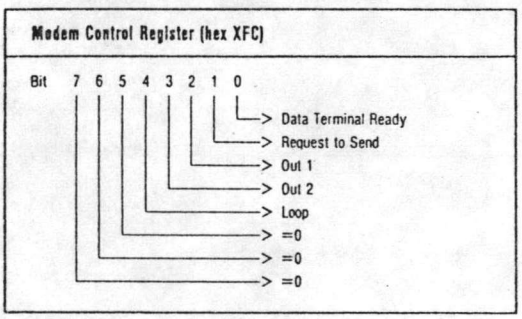
logical 1's is sent or checked in the data word bits and parity bit. When both bit 3 and bit 4 are a logical 1, an even number of bits is sent or checked.

Bit 5 This bit is the stuck-parity bit. When bit 3 is a logical 1 and bit 5 is a logical 1, the parity bit is sent and then detected by the receiver as a logical 0, if bit 4 is a logical 1, or as a logical 1 if bit 4 is a logical 0.

Bit 6 This bit is the set-break control bit. When bit 6 is set to a logical 1, the serial output (SOUT) is forced to the spacing (logical 0) state and remains there regardless of other transmitter activity. The set-break is disabled by setting bit 6 to logical 0. This feature enables the microprocessor to select a specific terminal in a computer communications system.

Bit 7 This bit is the divisor-latch access bit (DLAB). It must be set high (logical 1) to gain access to the divisor latches of the baud-rate generator during a read or write operation. It must be set low (logical 0) to gain access to the receiver buffer, the transmitter holding register, or the interrupt enable register.

Modem Control Register (Hex XFC): This 8-bit register controls the data exchange with the modem or data set (an external device acting as a modem).



Modem Control Register

Bit 0 This bit controls the '-data terminal ready' (-DTR) output. When bit 0 is set to logical 1, the -DTR output is forced active. When bit 0 is reset to logical 0, the '-DTR' output is forced inactive.

Bit 1 This bit controls the '-request-to-send' (-RTS) output. Bit 1 affects the '-RTS' output in the same way bit 0 affects the '-DTR' output.

Bit 2 This bit controls the '-Output 1' (-OUT 1) signal, which is a spare the programmer can use. Bit 2 affects the '-OUT 1' output in the same way bit 0 affects the '-DTR' output.

Bit 3 This bit controls the '-Output 2' (-OUT 2) signal, which is a spare the programmer can use. Bit 3 affects the '-OUT 2' output in the same way bit 0 affects the '-DTR' output.

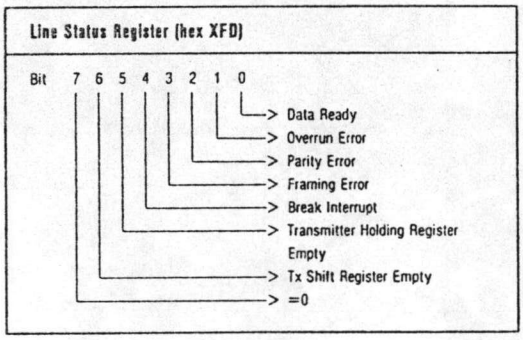
Bit 4 This bit provides a loopback feature for diagnostic testing of the controller. When bit 4 is set to logical 1, the following occur: the 'transmitter serial output' (SOUT) is set to the active state; the 'receiver serial input' (SIN) is disconnected; the output of the transmitter shift register is "looped back" to the receiver shift register input; the four modem-control inputs ('-CTS', '-DSR', '-RLSD', and '-RI') are disconnected; and the four modem-control outputs ('-DTR', '-RTS', '-OUT 1' and '-OUT 2') are internally connected to the four modem control inputs. In the diagnostic mode, data sent is immediately received. This feature allows the processor to verify the transmit- and receive-data paths of the controller.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational, as are the modem-control interrupts. But the interrupts' sources are now the lower four bits of the modem control register (MCR) instead of the four modem-control inputs. The interrupts are still controlled by the interrupt enable register.

The controller's interrupt system can be tested by writing to the lower six bits of the line status register and the lower four bits of the modem status register. Setting any of these bits to logical 1 generates the appropriate interrupt (if enabled). Resetting these interrupts is the same as for normal controller operation. To return to normal operation, the registers must be reprogrammed for normal operation, and then bit 4 of the MCR must be reset to logical 0.

Bits 5–7 These bits are permanently set to logical 0.

Line Status Register (Hex XFD): This 8-bit register provides the processor with status information about the data transfer.



Line Status Register

Bit 0 This bit is the receiver data ready (DR) indicator. It is set to logical 1 whenever a complete incoming character has been received and transferred into the receiver buffer register. Bit 0 may be reset to logical 0 by the processor either reading the data in the receiver's buffer register or writing logical 0 in it.

Bit 1 This bit is the overrun error (OE) indicator. It indicates that data in the receiver's buffer register was not read by the processor before the next character was transferred into the register, thereby destroying the previous character. The OE indicator is reset whenever the processor reads the contents of the line status register.

Bit 2 This bit is the parity error (PE) indicator and indicates the received data character does not have the correct even or odd parity, as selected by the even-parity-select bit. The PE bit is set to logical 1 upon detection of a parity error, and is reset to logical 0 whenever the processor reads the contents of the line status register.

Bit 3 This bit is the framing error (FE) indicator. It indicates the received character did not have a valid stop bit. Bit 3 is set to logical 1 whenever the stop bit following the last data bit or parity bit is detected as a zero bit (spacing level).

Bit 4 This bit is the break interrupt (BI) indicator. It is set to logical 1 whenever the received data input is held in the spacing state (logical 0) for longer than a fullword transmission time (that is, the total time of start bit + data bits + parity stop bits).

> **Note:** Bits 1 through 4 are error conditions that produce a receiver line-status interrupt whenever any of the corresponding conditions are detected.

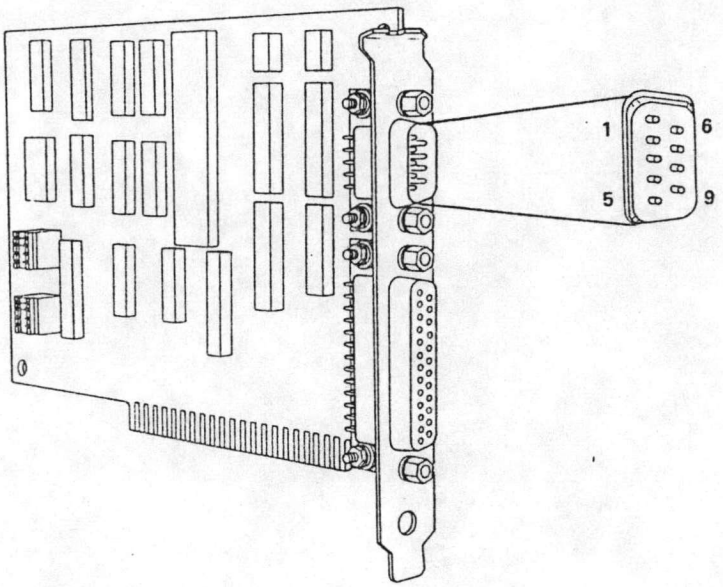
Bit 5 This bit is the transmitter holding register empty (THRE) indicator. It indicates the controller is ready to accept a new character for transmission. In addition, this bit causes the controller to issue an interrupt to the processor when the TRHE interrupt enable is set active. The THRE bit is set to logical 1 when a character is transferred from the transmitter holding register into the transmitter shift register. It is reset to logical 0 when the processor loads the transmitter holding register.

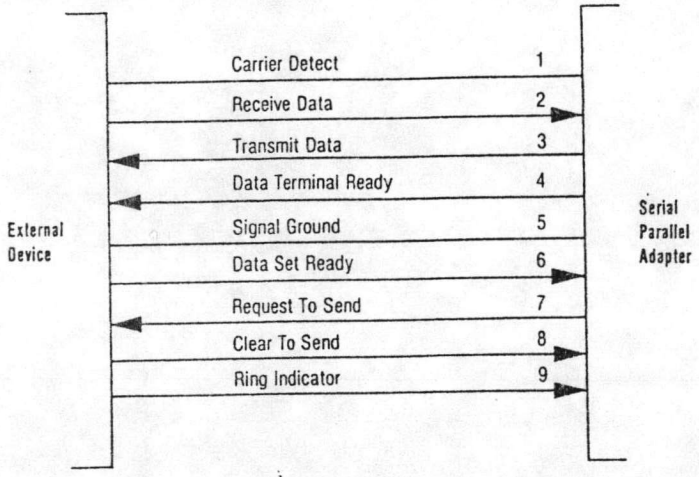
Bit 6 This bit is the transmitter empty (TEMT) indicator. It is set to logical 1 whenever the transmitter holding request (THR) and the transmitter shift request (TSR) are both empty. It is reset to logical 0 whenever THR or TSR contains a data character.

Bit 7 This bit is permanently set to logical 0.

Pin Assignment for Serial Port

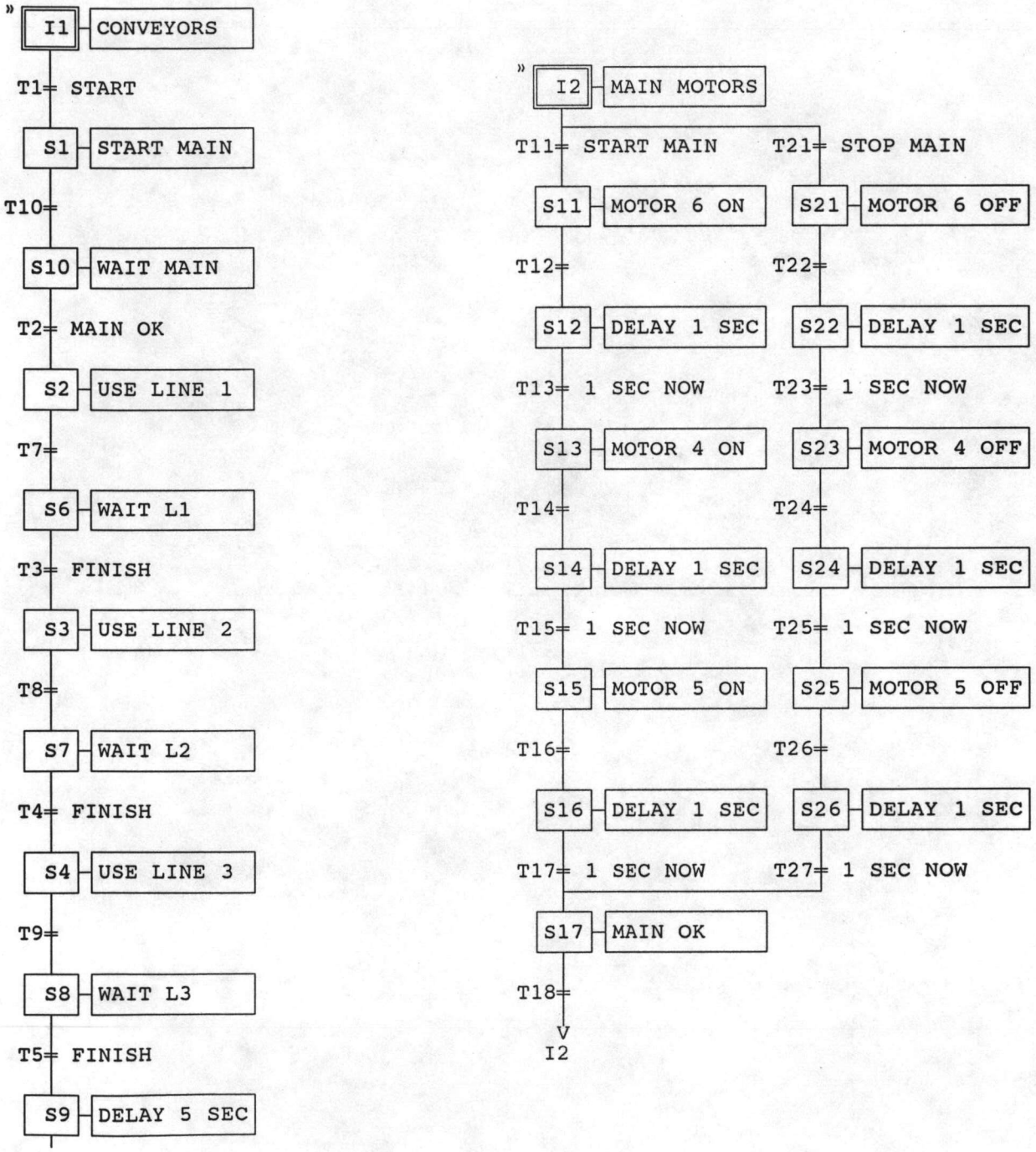
The following figure shows the pin assignments for the serial port in a communications environment.

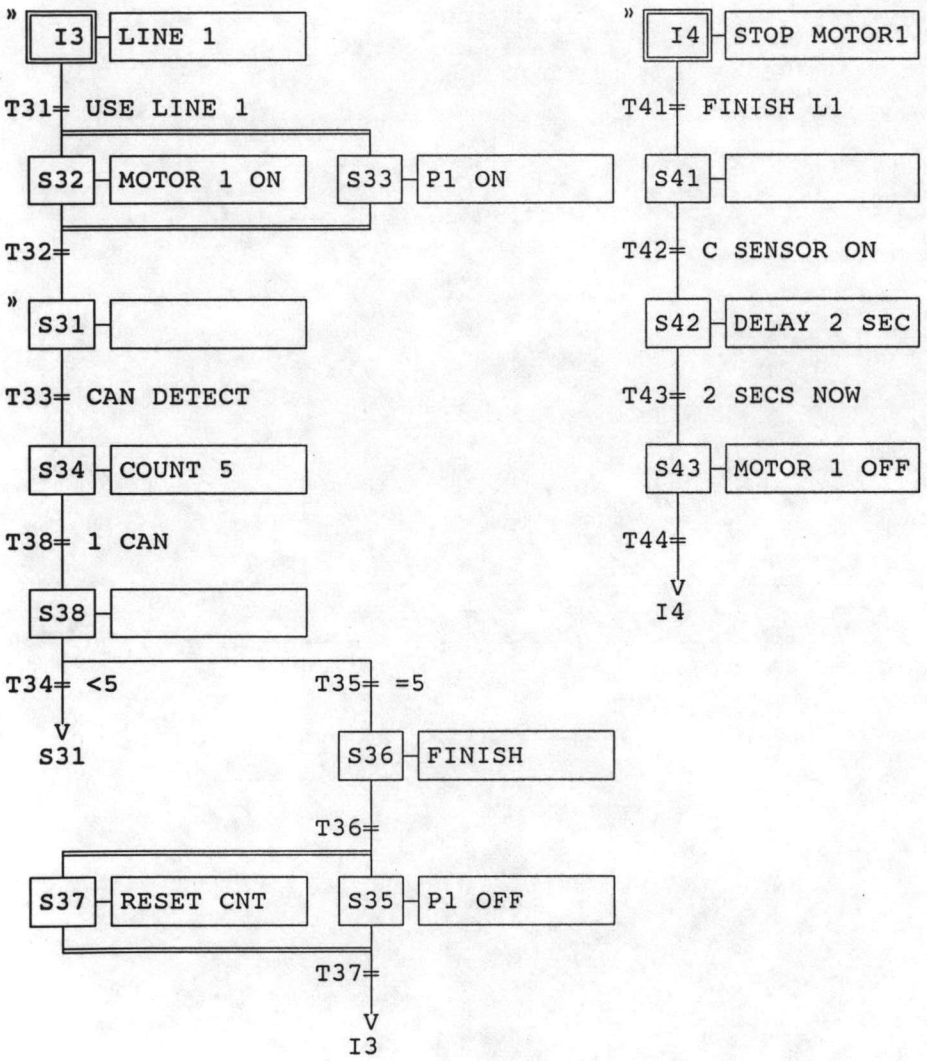


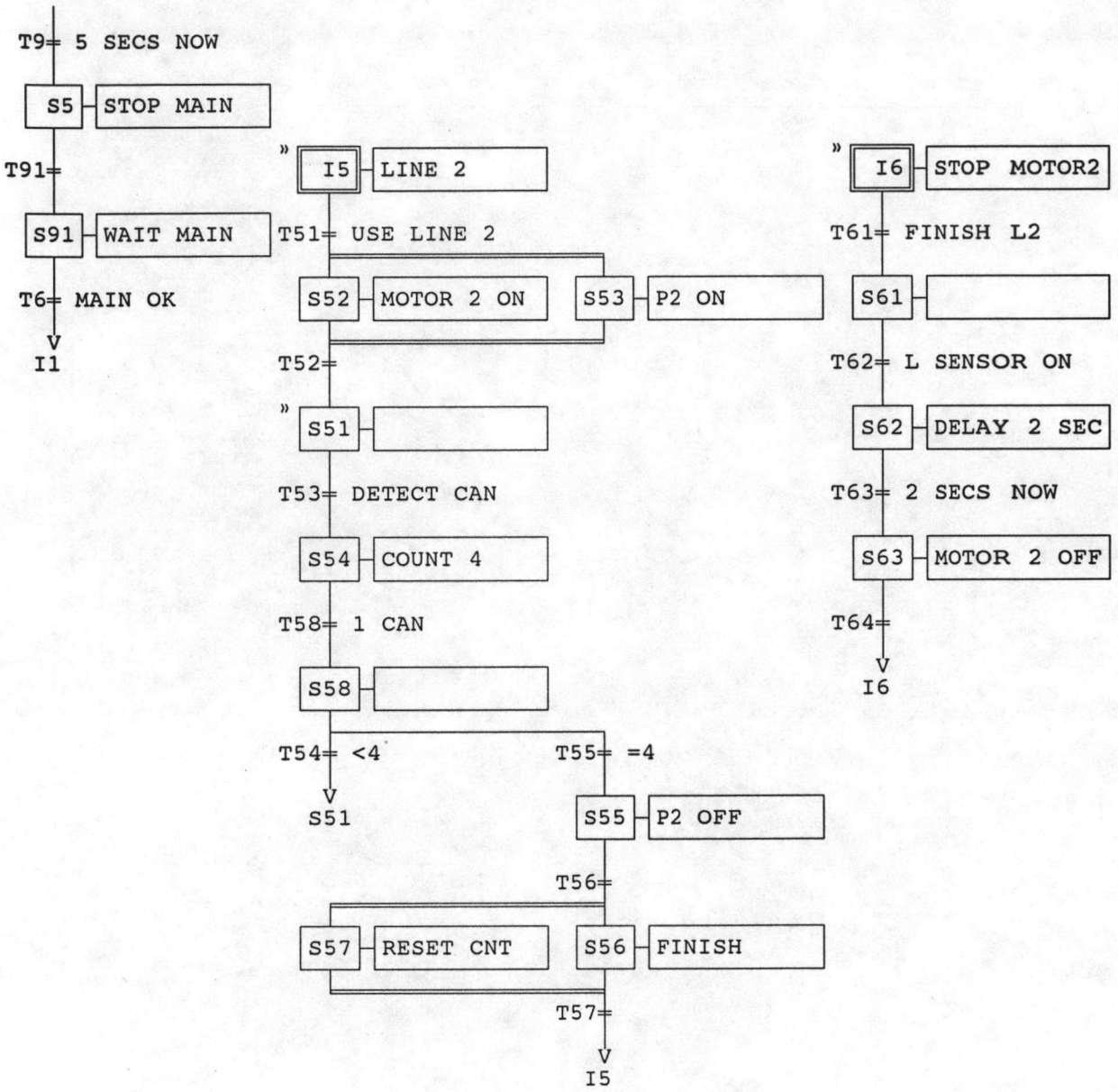


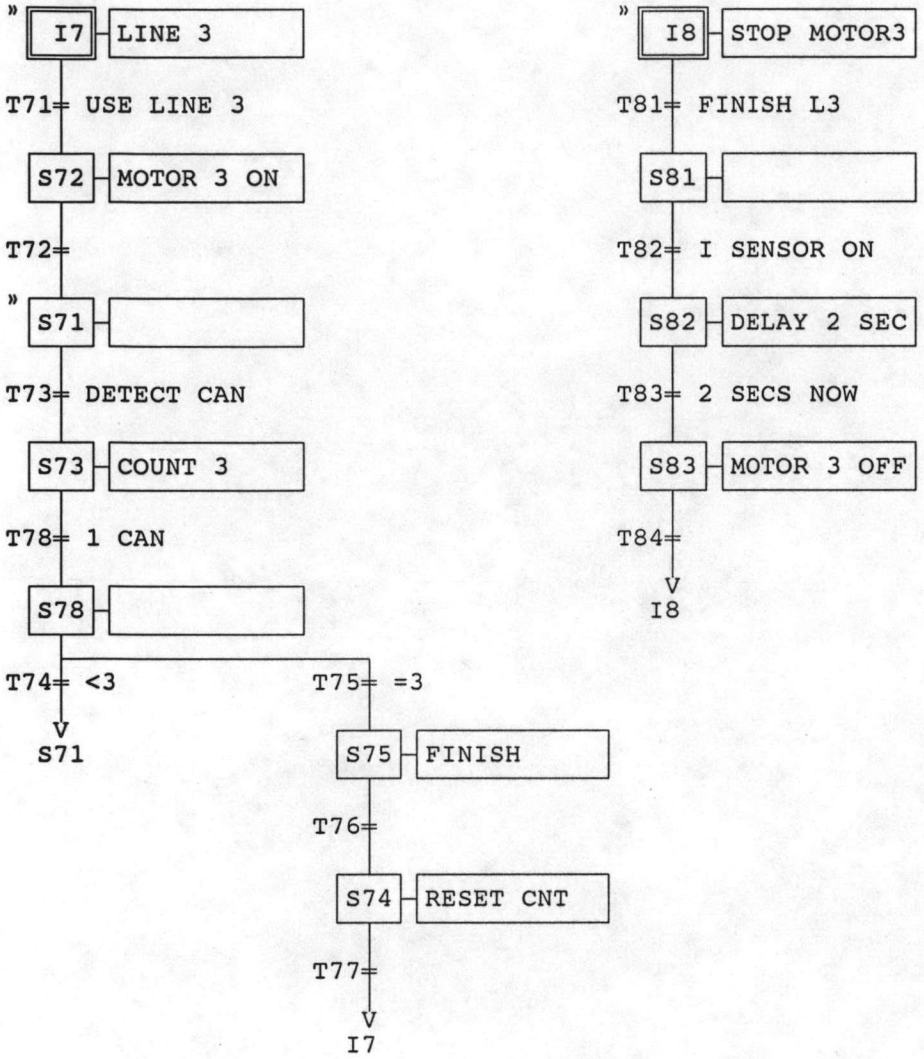
ภาคผนวก ง

ตัวอย่างเอกสารที่พิมพ์


I1
CONVEYORS
T1= START
S1
START MAIN
T10=
S10
WAIT MAIN
T2= MAIN OK
S2
USE LINE 1
T7=
S6
WAIT L1
T3= FINISH
S3
USE LINE 2
T8=
S7
WAIT L2
T4= FINISH
S4
USE LINE 3
T9=
S8
WAIT L3
T5= FINISH
S9
DELAY 5 SEC
I2
MAIN MOTORS
T11= START MAIN
T21= STOP MAIN
S11
MOTOR 6 ON
S21
MOTOR 6 OFF
T12=
T22=
S12
DELAY 1 SEC
S22
DELAY 1 SEC
T13= 1 SEC NOW
T23= 1 SEC NOW
S13
MOTOR 4 ON
S23
MOTOR 4 OFF
T14=
T24=
S14
DELAY 1 SEC
S24
DELAY 1 SEC
T15= 1 SEC NOW
T25= 1 SEC NOW
S15
MOTOR 5 ON
S25
MOTOR 5 OFF
T16=
T26=
S16
DELAY 1 SEC
S26
DELAY 1 SEC
T17= 1 SEC NOW
T27= 1 SEC NOW
S17
MAIN OK
T18=
I2


I3
LINE 1
T31= USE LINE 1
S32
MOTOR 1 ON
S33
P1 ON
T32=
S31
T33= CAN DETECT
S34
COUNT 5
T38= 1 CAN
S38
T34= <5
S31
T35= =5
S36
FINISH
T36=
S37
RESET CNT
S35
P1 OFF
T37=
I3
I4
STOP MOTOR1
T41= FINISH L1
S41
T42= C SENSOR ON
S42
DELAY 2 SEC
T43= 2 SECS NOW
S43
MOTOR 1 OFF
T44=
I4


T9= 5 SECS NOW
S5 STOP MAIN
T91=
S91 WAIT MAIN
T6= MAIN OK
V
I1
I5 LINE 2
T51= USE LINE 2
S52 MOTOR 2 ON
S53 P2 ON
T52=
S51
T53= DETECT CAN
S54 COUNT 4
T58= 1 CAN
S58
T54= <4
V
S51
T55= =4
S55 P2 OFF
T56=
S57 RESET CNT
S56 FINISH
T57=
V
I5
I6 STOP MOTOR2
T61= FINISH L2
S61
T62= L SENSOR ON
S62 DELAY 2 SEC
T63= 2 SECS NOW
S63 MOTOR 2 OFF
T64=
V
I6


I7
LINE 3
T71= USE LINE 3
S72
MOTOR 3 ON
T72=
S71
T73= DETECT CAN
S73
COUNT 3
T78= 1 CAN
S78
T74= <3
S71
T75= =3
S75
FINISH
T76=
S74
RESET CNT
T77=
I7
I8
STOP MOTOR3
T81= FINISH L3
S81
T82= I SENSOR ON
S82
DELAY 2 SEC
T83= 2 SECS NOW
S83
MOTOR 3 OFF
T84=
I8

```
I1=(I1+T6*S91)*/S1
I2=(I2+T18*S17)*/S11*/S21
I3=(I3+T37*S35*S37)*/S32*/S33
I4=(I4+T44*S43)*/S41
S1=(S1+T1*I1)*/S10
S11=(S11+T11*I2)*/S12
S21=(S21+T21*I2)*/S22
S32=(S32+T31*I3)*/S31
S33=(S33+T31*I3)*/S31
S41=(S41+T41*I4)*/S42
S10=(S10+T10*S1)*/S2
S12=(S12+T12*S11)*/S13
S22=(S22+T22*S21)*/S23
S31=(S31+T32*S32*S33+T34*S38)*/S34
S42=(S42+T42*S41)*/S43
S2=(S2+T2*S10)*/S6
S13=(S13+T13*S12)*/S14
S23=(S23+T23*S22)*/S24
S34=(S34+T33*S31)*/S38
S43=(S43+T43*S42)*/I4
S6=(S6+T7*S2)*/S3
S14=(S14+T14*S13)*/S15
S24=(S24+T24*S23)*/S25
S38=(S38+T38*S34)*/S31*/S36
S3=(S3+T3*S6)*/S7
S15=(S15+T15*S14)*/S16
S25=(S25+T25*S24)*/S26
S36=(S36+T35*S38)*/S35*/S37
S7=(S7+T8*S3)*/S4
S16=(S16+T16*S15)*/S17
```

```
S26=(S26+T26*S25)*/S17
S37=(S37+T36*S36)*/I3
S35=(S35+T36*S36)*/I3
S4=(S4+T4*S7)*/S8
S17=(S17+T17*S16+T27*S26)*/I2
S8=(S8+T9*S4)*/S9
S9=(S9+T5*S8)*/S5
S5=(S5+T9*S9)*/S91
I5=(I5+T57*S56*S57)*/S52*/S53
I6=(I6+T64*S63)*/S61
I7=(I7+T77*S74)*/S72
I8=(I8+T84*S83)*/S81
S91=(S91+T91*S5)*/I1
S52=(S52+T51*I5)*/S51
S53=(S53+T51*I5)*/S51
S61=(S61+T61*I6)*/S62
S72=(S72+T71*I7)*/S71
S81=(S81+T81*I8)*/S82
S51=(S51+T52*S52*S53+T54*S58)*/S54
S62=(S62+T62*S61)*/S63
S71=(S71+T72*S72+T74*S78)*/S73
S82=(S82+T82*S81)*/S83
S54=(S54+T53*S51)*/S58
S63=(S63+T63*S62)*/I6
S73=(S73+T73*S71)*/S78
S83=(S83+T83*S82)*/I8
S58=(S58+T58*S54)*/S51*/S55
S78=(S78+T78*S73)*/S71*/S75
S55=(S55+T55*S58)*/S56*/S57
S75=(S75+T75*S78)*/S74
```

```
S57=(S57+T56*S55)*/I5
S56=(S56+T56*S55)*/I5
S74=(S74+T76*S75)*/I7
```

TYPE	NAME	MATTER	I/O	AUX RL	COMMENT
INITIAL	I1	CONVEYORS		3200	
	I2	MAIN MOTORS		3202	
	I3	LINE 1		3203	
	I4	STOP MOTOR1		3204	
	I5	LINE 2		3512	
	I6	STOP MOTOR2		3513	
	I7	LINE 3		3514	
	I8	STOP MOTOR3		3515	
STEP	S1	START MAIN	2000	3205	
	S11	MOTOR 6 ON	0206	3210	HSET
	S21	MOTOR 6 OFF	0206	3211	HRST
	S32	MOTOR 1 ON	0201	3212	HSET
	S33	P1 ON	0211	3213	HSET
	S41			3214	
	S10	WAIT MAIN		3215	
	S12	DELAY 1 SEC	T000	3302	0010
	S22	DELAY 1 SEC	T004	3303	0010
	S31			3304	
	S42	DELAY 2 SEC	T011	3305	0020
	S2	USE LINE 1	2002	3306	
	S13	MOTOR 4 ON	0204	3311	HSET
	S23	MOTOR 4 OFF	0204	3312	HRST
	S34	COUNT 5	C101	3313	0005
	S43	MOTOR 1 OFF	0201	3314	HRST
	S6	WAIT L1		3315	
	S14	DELAY 1 SEC	T001	3402	0010
	S24	DELAY 1 SEC	T005	3403	0010
	S38			3404	
	S3	USE LINE 2	2004	3405	
	S15	MOTOR 5 ON	0205	3410	HSET
	S25	MOTOR 5 OFF	0205	3411	HRST
	S36	FINISH	2003	3412	
	S7	WAIT L2		3413	
	S16	DELAY 1 SEC	T002	3415	0010
	S26	DELAY 1 SEC	T006	3500	0010
	S37	RESET CNT	C101	3501	CRST
	S35	P1 OFF	0211	3502	HRST
	S4	USE LINE 3	2006	3503	
	S17	MAIN OK	2001	3506	
	S8	WAIT L3		3507	
	S9	DELAY 5 SEC	T021	3509	0050
	S5	STOP MAIN	2008	3511	
	S91	WAIT MAIN		3600	
	S52	MOTOR 2 ON	0202	3606	HSET
	S53	P2 ON	0212	3607	HSET
	S61			3608	
	S72	MOTOR 3 ON	0203	3609	HSET
	S81			3610	
	S51			3613	
	S62	DELAY 2 SEC	T012	3614	0020
	S71			3615	

TYPE	NAME	MATTER	I/O	AUX RL	COMMENT
	S82	DELAY 2 SEC	T013	3700	0020
	S54	COUNT 4	C102	3705	0004
	S63	MOTOR 2 OFF	0202	3706	HRST
	S73	COUNT 3	C103	3707	0003
	S83	MOTOR 3 OFF	0203	3708	HRST
	S58			3711	
	S78			3712	
	S55	P2 OFF	0212	3801	HRST
	S75	FINISH	2007	3802	
	S57	RESET CNT	C102	3803	CRST
	S56	FINISH	2005	3804	
	S74	RESET CNT	C103	3805	CRST
TRANSITION	T1	START	0000	3201	
	T11	START MAIN	2000	3206	
	T21	STOP MAIN	2008	3207	
	T31	USE LINE 1	2002	3208	
	T41	FINISH L1	2003	3209	
	T10		1	6113	
	T12		1	6113	
	T22		1	6113	
	T32		1	6113	
	T42	C SENSOR ON	0005	3300	
	T2	MAIN OK	2001	3301	
	T13	1 SEC NOW	T000	3307	
	T23	1 SEC NOW	T004	3308	
	T33	CAN DETECT	0002	3309	
	T43	2 SECS NOW	T011	3310	
	T7		1	6113	
	T14		1	6113	
	T24		1	6113	
	T38	1 CAN	0002	3400	I
	T44		1	6113	
	T3	FINISH	2003	3401	
	T15	1 SEC NOW	T001	3406	
	T25	1 SEC NOW	T005	3407	
	T34	<5	C101	3408	I
	T35	=5	C101	3409	
	T8		1	6113	
	T16		1	6113	
	T26		1	6113	
	T36		1	6113	
	T4	FINISH	2005	3414	
	T17	1 SEC NOW	T002	3504	
	T27	1 SEC NOW	T006	3505	
	T37		1	6113	
	T9		1	6113	
	T18		1	6113	
	T5	FINISH	2007	3508	
	T9	5 SECS NOW	T021	3510	
	T91		1	6113	
	T51	USE LINE 2	2004	3601	
	T61	FINISH L2	2005	3602	
	T71	USE LINE 3	2006	3603	
	T81	FINISH L3	2007	3604	
	T6	MAIN OK	2001	3605	

TYPE	NAME	MATTER	I/O	AUX RL	COMMENT
	T52		1	6113	
	T62	L SENSOR ON	0006	3611	
	T72		1	6113	
	T82	I SENSOR ON	0007	3612	
	T53	DETECT CAN	0003	3701	
	T63	2 SECS NOW	T012	3702	
	T73	DETECT CAN	0004	3703	
	T83	2 SECS NOW	T013	3704	
	T58	1 CAN	0003	3709	I
	T64		1	6113	
	T78	1 CAN	0004	3710	I
	T84		1	6113	
	T54	<4	C102	3713	I
	T55	=4	C102	3714	
	T74	<3	C103	3715	I
	T75	=3	C103	3800	
	T56		1	6113	
	T76		1	6113	
	T57		1	6113	
	T77		1	6113	

```
LD  6115
OR  3200
OUT 3200
LD  6115
OR  3202
OUT 3202
LD  6115
OR  3203
OUT 3203
LD  6115
OR  3204
OUT 3204
LD  6115
OR  3512
OUT 3512
LD  6115
OR  3513
OUT 3513
LD  6115
OR  3514
OUT 3514
LD  6115
OR  3515
OUT 3515
LD  0000
OUT 3201
LD  2000
OUT 3206
LD  2008
OUT 3207
LD  2002
OUT 3208
LD  2003
OUT 3209
LD  0005
OUT 3300
LD  2001
OUT 3301
LD  TIM 000
OUT 3307
LD  TIM 004
OUT 3308
LD  0002
OUT 3309
LD  TIM 011
OUT 3310
LD  NOT 0002
OUT 3400
LD  2003
OUT 3401
LD  TIM 001
OUT 3406
LD  TIM 005
OUT 3407
LD  NOT CNT 101
OUT 3408
LD  CNT 101
OUT 3409
LD  2005
OUT 3414
LD  TIM 002
OUT 3504
LD  TIM 006
OUT 3505
LD  2007
OUT 3508
LD  TIM 021
OUT 3510
LD  2004
OUT 3601
LD  2005
OUT 3602
LD  2006
OUT 3603
LD  2007
OUT 3604
LD  2001
OUT 3605
LD  0006
OUT 3611
LD  0007
OUT 3612
LD  0003
OUT 3701
LD  TIM 012
OUT 3702
LD  0004
OUT 3703
LD  TIM 013
OUT 3704
LD  NOT 0003
OUT 3709
LD  NOT 0004
OUT 3710
LD  NOT CNT 102
OUT 3713
LD  CNT 102
OUT 3714
LD  NOT CNT 103
OUT 3715
LD  CNT 103
OUT 3800
LD  3200
LD  3605
AND 3600
OR  LD
AND NOT 3205
OUT 3200
LD  3202
LD  6113
AND 3506
OR  LD
AND NOT 3210
AND NOT 3211
OUT 3202
LD  3203
LD  6113
AND 3502
AND 3501
OR  LD
AND NOT 3212
AND NOT 3213
OUT 3203
LD  3204
LD  6113
AND 3314
OR  LD
AND NOT 3214
OUT 3204
LD  3205
LD  3201
AND 3200
OR  LD
AND NOT 3215
OUT 3205
LD  3210
LD  3206
AND 3202
OR  LD
AND NOT 3302
OUT 3210
LD  3211
LD  3207
AND 3202
OR  LD
AND NOT 3303
OUT 3211
LD  3212
LD  3208
AND 3203
OR  LD
AND NOT 3304
OUT 3212
LD  3213
LD  3208
AND 3203
OR  LD
AND NOT 3304
OUT 3213
LD  3214
LD  3209
AND 3204
OR  LD
AND NOT 3305
OUT 3214
LD  3215
LD  6113
AND 3205
OR  LD
AND NOT 3306
OUT 3215
LD  3302
LD  6113
AND 3210
OR  LD
AND NOT 3311
OUT 3302
LD  3303
LD  6113
AND 3211
OR  LD
AND NOT 3312
OUT 3303
LD  3304
LD  6113
AND 3212
AND 3213
OR  LD
LD  3408
AND 3404
OR  LD
AND NOT 3313
OUT 3304
LD  3305
LD  3300
AND 3214
OR  LD
AND NOT 3314
OUT 3305
LD  3306
LD  3301
AND 3215
OR  LD
AND NOT 3315
OUT 3306
LD  3311
LD  3307
AND 3302
OR  LD
AND NOT 3402
OUT 3311
LD  3312
LD  3308
AND 3303
OR  LD
AND NOT 3403
OUT 3312
LD  3313
LD  3309
AND 3304
OR  LD
AND NOT 3404
OUT 3313
LD  3314
LD  3310
AND 3305
OR  LD
AND NOT 3204
OUT 3314
LD  3315
LD  6113
AND 3306
OR  LD
AND NOT 3405
OUT 3315
LD  3402
LD  6113
AND 3311
OR  LD
AND NOT 3410
```

```
OUT 3402
LD  3403
LD  6113
AND 3312
OR  LD
AND NOT 3411
OUT 3403
LD  3404
LD  3400
AND 3313
OR  LD
AND NOT 3304
AND NOT 3412
OUT 3404
LD  3405
LD  3401
AND 3315
OR  LD
AND NOT 3413
OUT 3405
LD  3410
LD  3406
AND 3402
OR  LD
AND NOT 3415
OUT 3410
LD  3411
LD  3407
AND 3403
OR  LD
AND NOT 3500
OUT 3411
LD  3412
LD  3409
AND 3404
OR  LD
AND NOT 3502
AND NOT 3501
OUT 3412
LD  3413
LD  6113
AND 3405
OR  LD
AND NOT 3503
OUT 3413
LD  3415
LD  6113
AND 3410
OR  LD
AND NOT 3506
OUT 3415
LD  3500
LD  6113
AND 3411
OR  LD
AND NOT 3506
OUT 3500
LD  3501
LD  6113
AND 3412
OR  LD
AND NOT 3203
OUT 3501
LD  3502
LD  6113
AND 3412
OR  LD
AND NOT 3203
OUT 3502
LD  3503
LD  3414
AND 3413
OR  LD
AND NOT 3507
OUT 3503
LD  3506
LD  3504
AND 3415
OR  LD
LD  3505
AND 3500
OR  LD
AND NOT 3202
OUT 3506
LD  3507
LD  6113
AND 3503
OR  LD
AND NOT 3509
OUT 3507
LD  3509
LD  3508
AND 3507
OR  LD
AND NOT 3511
OUT 3509
LD  3511
LD  3510
AND 3509
OR  LD
AND NOT 3600
OUT 3511
LD  3512
LD  6113
AND 3804
AND 3803
OR  LD
AND NOT 3606
AND NOT 3607
OUT 3512
LD  3513
LD  6113
AND 3706
OR  LD
AND NOT 3608
OUT 3513
LD  3514
LD  6113
AND 3805
OR  LD
AND NOT 3609
OUT 3514
LD  3515
LD  6113
AND 3708
OR  LD
AND NOT 3610
OUT 3515
LD  3600
LD  6113
AND 3511
OR  LD
AND NOT 3200
OUT 3600
LD  3606
LD  3601
AND 3512
OR  LD
AND NOT 3613
OUT 3606
LD  3607
LD  3601
AND 3512
OR  LD
AND NOT 3613
OUT 3607
LD  3608
LD  3602
AND 3513
OR  LD
AND NOT 3614
OUT 3608
LD  3609
LD  3603
AND 3514
OR  LD
AND NOT 3615
OUT 3609
LD  3610
LD  3604
AND 3515
OR  LD
AND NOT 3700
OUT 3610
LD  3613
LD  6113
AND 3606
AND 3607
OR  LD
LD  3713
AND 3711
OR  LD
AND NOT 3705
OUT 3613
LD  3614
LD  3611
AND 3608
OR  LD
AND NOT 3706
OUT 3614
LD  3615
LD  6113
AND 3609
OR  LD
LD  3715
AND 3712
OR  LD
AND NOT 3707
OUT 3615
LD  3700
LD  3612
AND 3610
OR  LD
AND NOT 3708
OUT 3700
LD  3705
LD  3701
AND 3613
OR  LD
AND NOT 3711
OUT 3705
LD  3706
LD  3702
AND 3614
OR  LD
AND NOT 3513
OUT 3706
LD  3707
LD  3703
AND 3615
OR  LD
AND NOT 3712
OUT 3707
LD  3708
LD  3704
AND 3700
OR  LD
AND NOT 3515
OUT 3708
LD  3711
LD  3709
AND 3705
OR  LD
AND NOT 3613
AND NOT 3801
OUT 3711
LD  3712
LD  3710
AND 3707
OR  LD
AND NOT 3615
AND NOT 3802
OUT 3712
LD  3801
LD  3714
AND 3711
OR  LD
AND NOT 3804
AND NOT 3803
OUT 3801
```

```
LD  3802
LD  3800
AND 3712
OR  LD
AND NOT 3805
OUT 3802
LD  3803
LD  6113
AND 3801
OR  LD
AND NOT 3512
OUT 3803
LD  3804
LD  6113
AND 3801
OR  LD
AND NOT 3512
OUT 3804
LD  3805
LD  6113
AND 3802
OR  LD
AND NOT 3514
OUT 3805
LD  3205
OUT 2000
LD  3210
OR  0206
AND NOT 3211
OUT 0206
LD  3212
OR  0201
AND NOT 3314
OUT 0201
LD  3213
OR  0211
AND NOT 3502
OUT 0211
LD  3302
TIM 000
  #0010
LD  3303
TIM 004
  #0010
LD  3305
TIM 011
  #0020
LD  3306
OUT 2002
LD  3311
OR  0204
AND NOT 3312
OUT 0204
LD  3313
LD  6115
OR  3501
CNT 101
  #0005
LD  3402
TIM 001
  #0010
LD  3403
TIM 005
  #0010
LD  3405
OUT 2004
LD  3410
OR  0205
AND NOT 3411
OUT 0205
LD  3412
OUT 2003
LD  3415
TIM 002
  #0010
LD  3500
TIM 006
  #0010
LD  3503
OUT 2006
LD  3506
OUT 2001
LD  3509
TIM 021
  #0050
LD  3511
OUT 2008
LD  3606
OR  0202
AND NOT 3706
OUT 0202
LD  3607
OR  0212
AND NOT 3801
OUT 0212
LD  3609
OR  0203
AND NOT 3708
OUT 0203
LD  3614
TIM 012
  #0020
LD  3700
TIM 013
  #0020
LD  3705
LD  6115
OR  3803
CNT 102
  #0004
LD  3707
LD  6115
OR  3805
CNT 103
  #0003
LD  3802
OUT 2007
LD  3804
OUT 2005
END
```

AUX	I/O
3200	
3204	
3208	2002
3211	0206
3215	
3300	0005
3304	
3308	T004
3311	0204
3315	
6113	1
3404	
3408	C101
3411	0205
6113	1
3500	T006
3504	T002
3506	2001
3509	T021
3512	
3600	
3604	2007
3608	
3611	0006
3614	T012
3702	T012
3706	0202

AUX	I/O
3201	0000
3205	2000
3209	2003
3212	0201
6113	1
3301	2001
3305	T011
3309	0002
3312	0204
6113	1
3401	2003
3405	2004
3409	C101
3412	2003
6113	1
3501	C101
3505	T006
3507	
3510	T021
3513	
3601	2004
3605	2001
3609	0203
6113	1
3615	
3703	0004
3707	C103

AUX	I/O
3202	
3206	2000
6113	1
3213	0211
6113	1
3302	T000
3306	2002
3310	T011
3313	C101
6113	1
3402	T001
3406	T001
6113	1
3413	
3414	2005
3502	0211
6113	1
6113	1
3511	2008
3514	
3602	2005
3606	0202
3610	
3612	0007
3700	T013
3704	T013
3708	0203

AUX	I/O
3203	
3207	2008
3210	0206
3214	
6113	1
3303	T004
3307	T000
6113	1
3314	0201
3400	0002
3403	T005
3407	T005
3410	0205
6113	1
3415	T002
3503	2006
6113	1
3508	2007
6113	1
3515	
3603	2006
3607	0212
6113	1
3613	
3701	0003
3705	C102
3709	0003

AUX	I/O
6113	1
3712	
3800	C103
6113	1

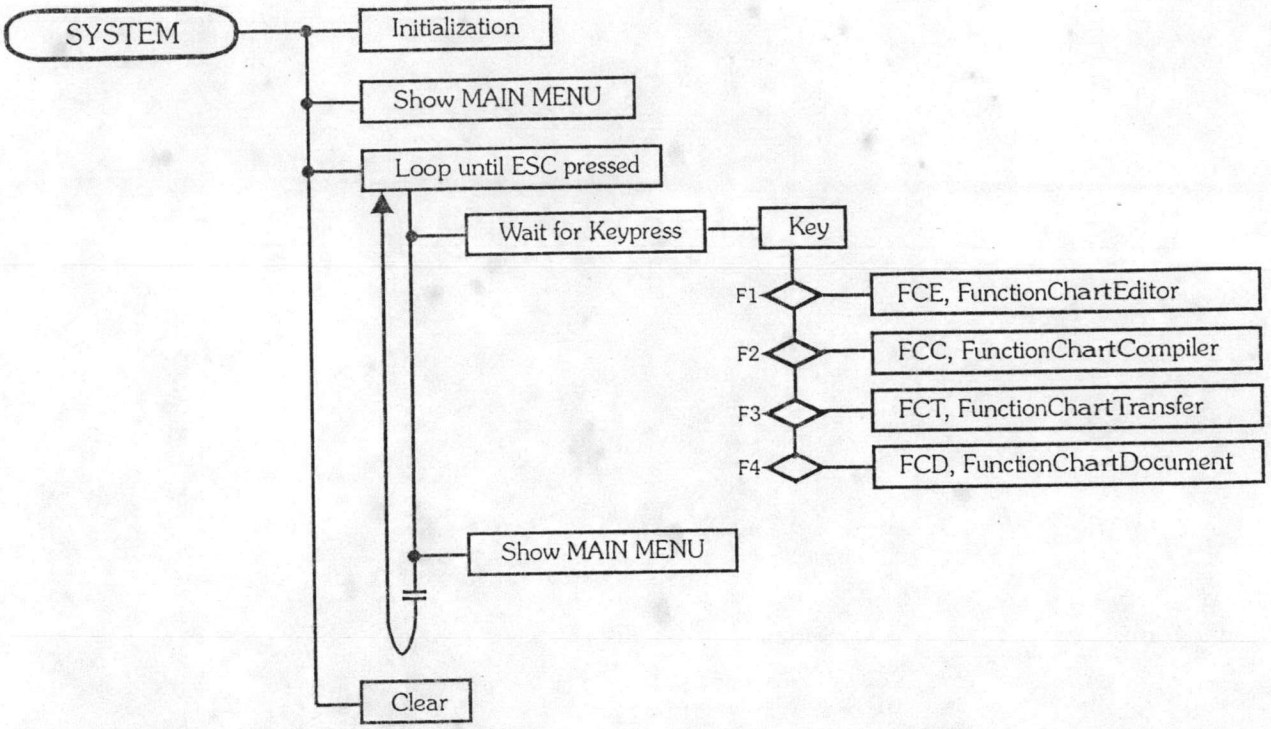
AUX	I/O
3710	0004
3713	C102
3801	0212
3803	C102

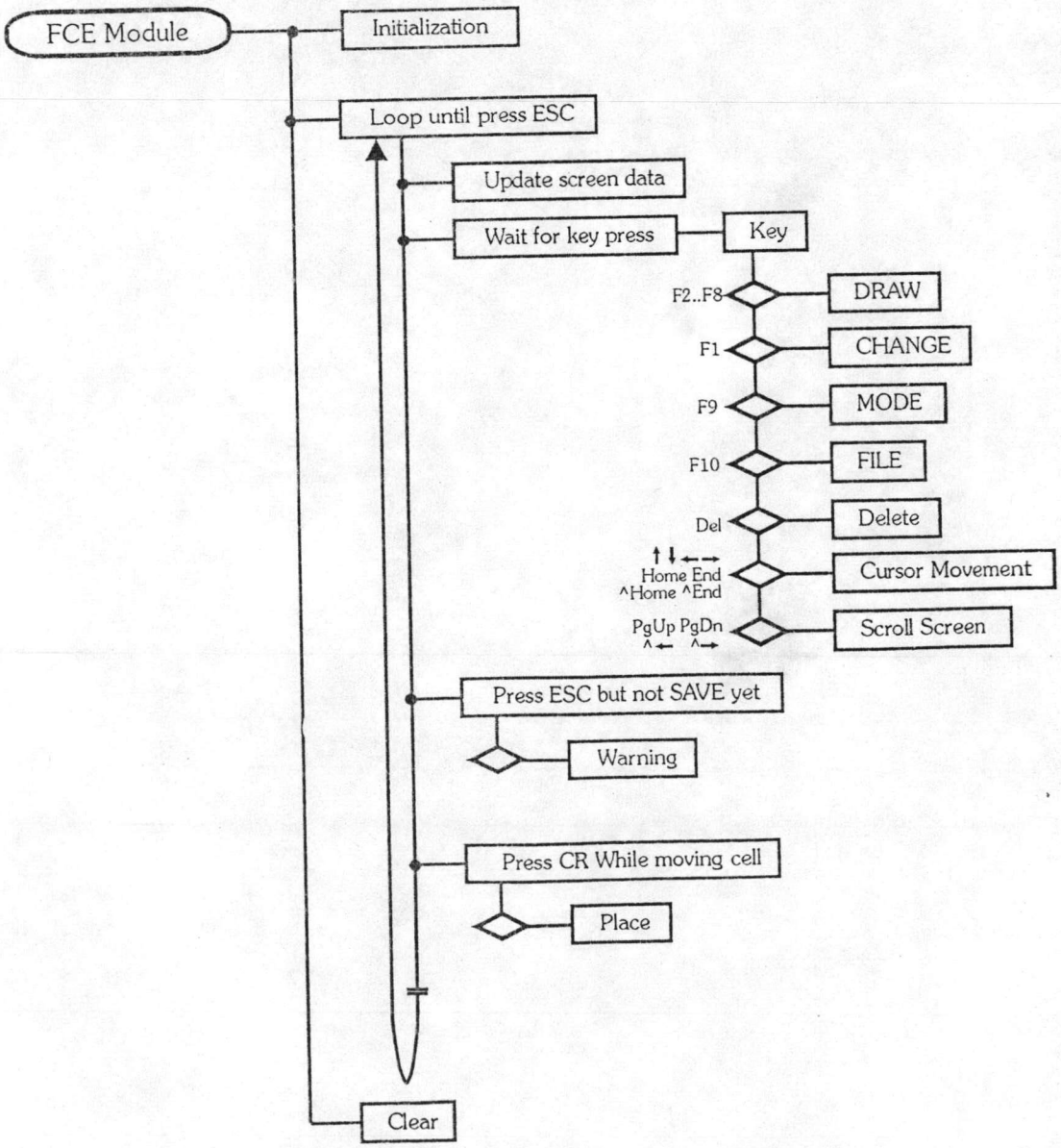
AUX	I/O
6113	1
3714	C102
3802	2007
3804	2005

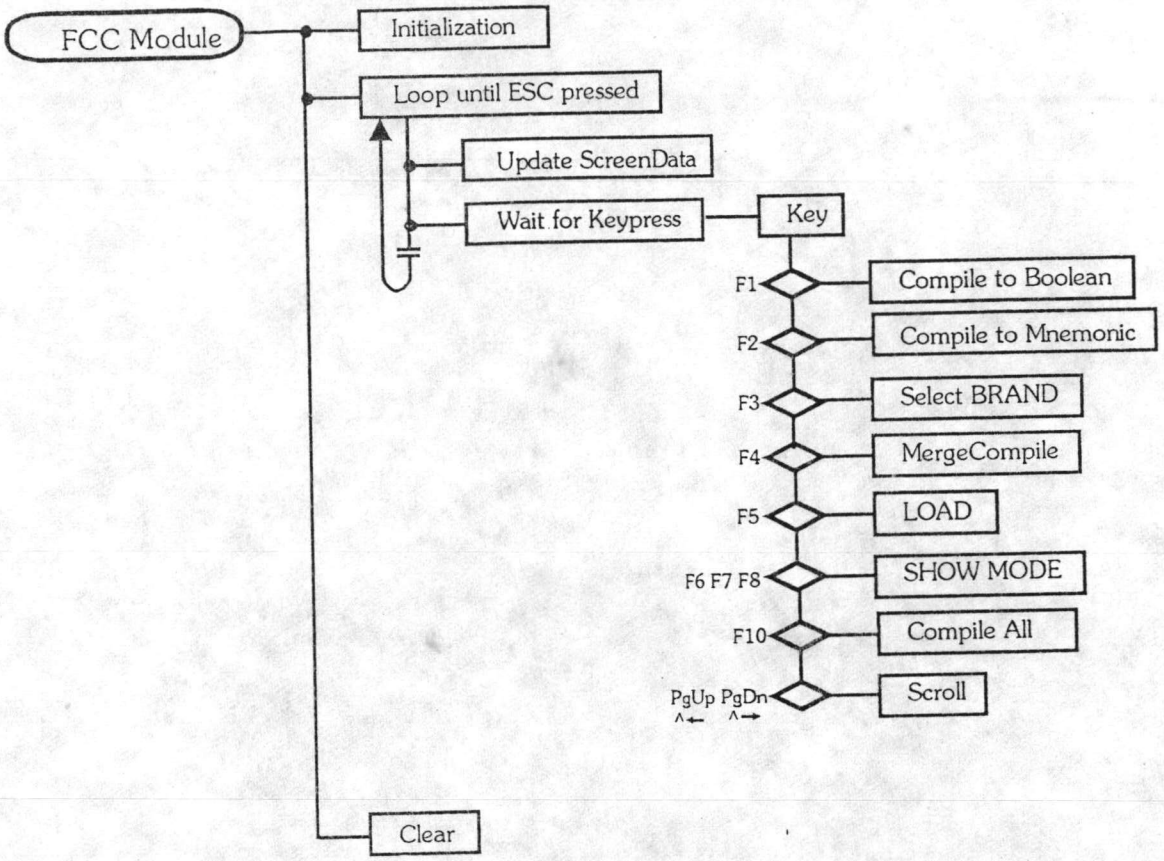
AUX	I/O
3711	
3715	C103
6113	1
3805	C103

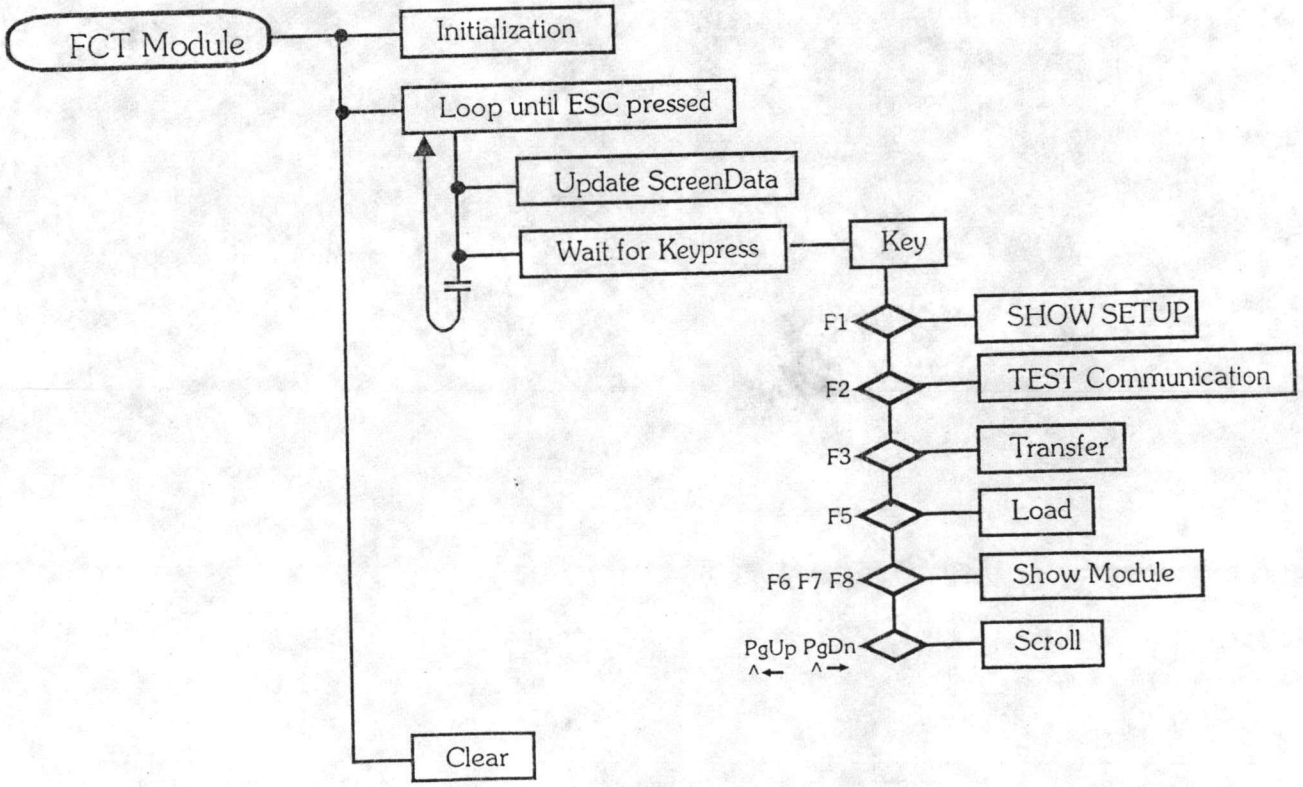
ภาคผนวก จ

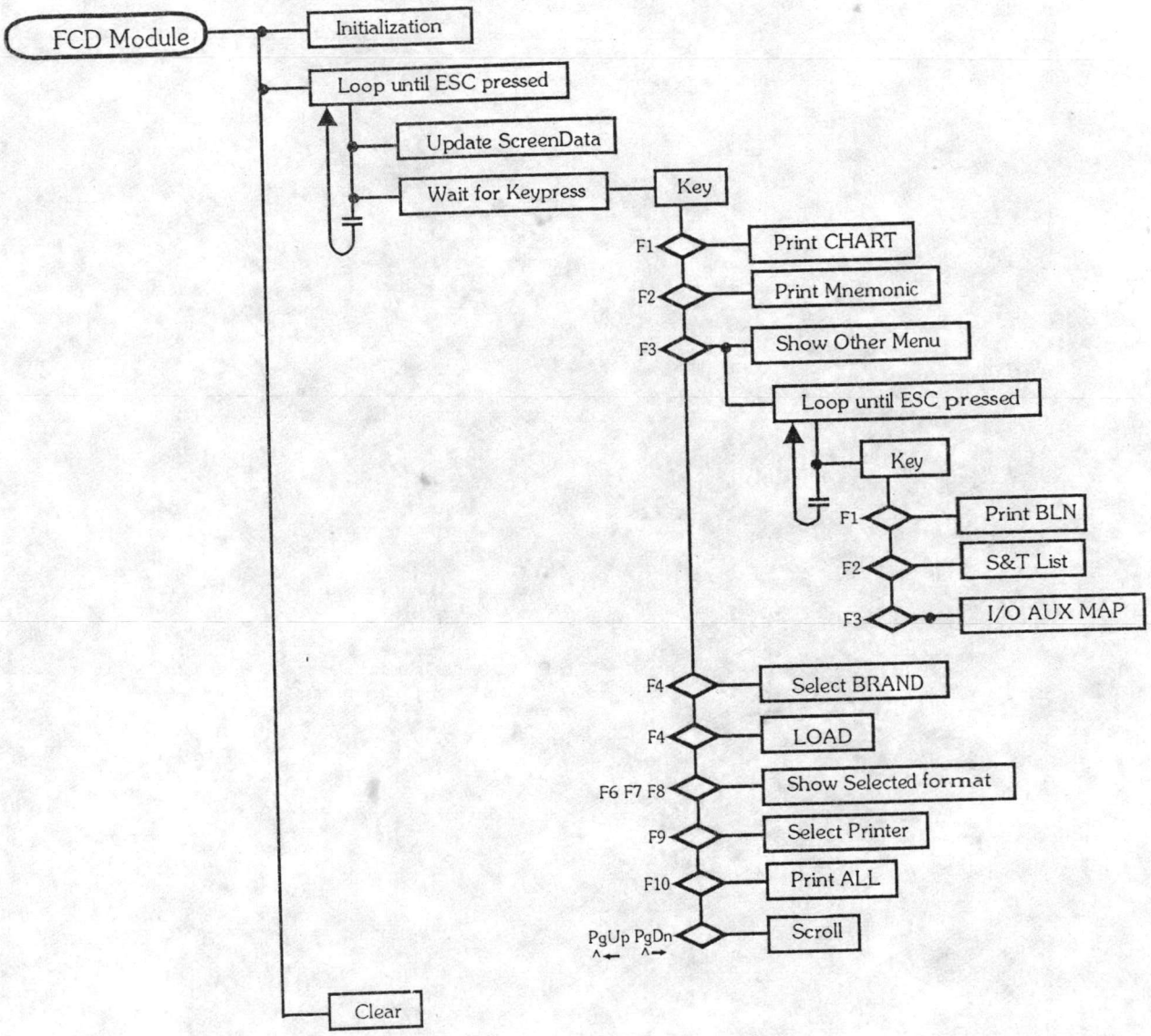
แผนภาพของระบบ


SYSTEM
Initialization
Show MAIN MENU
Loop until ESC pressed
Wait for Keypress
Key
F1
FCE, FunctionChartEditor
F2
FCC, FunctionChartCompiler
F3
FCT, FunctionChartTransfer
F4
FCD, FunctionChartDocument
Show MAIN MENU
Clear


FCE Module
Initialization
Loop until press ESC
Update screen data
Wait for key press
Key
F2..F8
DRAW
F1
CHANGE
F9
MODE
F10
FILE
Del
Delete
↑↓←→
Home End
^Home ^End
Cursor Movement
PgUp PgDn
^← ^→
Scroll Screen
Press ESC but not SAVE yet
Warning
Press CR While moving cell
Place
Clear


FCC Module
Initialization
Loop until ESC pressed
Update ScreenData
Wait for Keypress
Key
F1
Compile to Boolean
F2
Compile to Mnemonic
F3
Select BRAND
F4
MergeCompile
F5
LOAD
F6 F7 F8
SHOW MODE
F10
Compile All
PgUp PgDn
Scroll
Clear


FCT Module
Initialization
Loop until ESC pressed
Update ScreenData
Wait for Keypress
Key
F1
SHOW SETUP
F2
TEST Communication
F3
Transfer
F5
Load
F6 F7 F8
Show Module
PgUp PgDn
Scroll
Clear


FCD Module
Initialization
Loop until ESC pressed
Update ScreenData
Wait for Keypress
Key
F1
Print CHART
F2
Print Mnemonic
F3
Show Other Menu
Loop until ESC pressed
Key
F1
Print BLN
F2
S&T List
F3
I/O AUX MAP
F4
Select BRAND
F4
LOAD
F6 F7 F8
Show Selected format
F9
Select Printer
F10
Print ALL
PgUp PgDn
Scroll
Clear

ประวัติผู้เขียน

นาย ธเนศ พาณิชพัฒน์ เกิดเมื่อวันที่ 11 มีนาคม พ.ศ. 2509 ที่กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาอิเลคทรอนิคส์ ภาควิชาวิศวกรรมไฟฟ้า จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2530 และเข้าทำงานที่บริษัท ไมโครเอ็กซ์ จำกัด ในตำแหน่งโปรแกรมเมอร์ เป็นเวลา 1 ปี จากนั้นเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต ณ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2532 ระหว่างการศึกษาได้ลาพักการศึกษา 1 ภาคเรียน เพื่อไปฝึกงานเรื่องการประยุกต์ใช้งานเครื่องควบคุมซนิดโปรแกรมได้ในอุตสาหกรรม ที่บริษัท VONESCO จำกัด (VONESCO AG) ประเทศสวิสเซอร์แลนด์