CMOS single-chip 8-bit microcontroller — 87C751/87C751-16

FEATURES

- EPROM version of 83C751
- 80C51 based architecture
- Inter-Integrated Circuit (I^2C) serial bus interface
- Small package sizes
  - 24 pin DIP (300mil-wide "skinny DIP")
- Available in erasable quartz lid or one-time programmable ceramic package
- Wide oscillator frequency range
- Low power consumption
  - Normal operation, less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 2K × 8 EPROM, 64 × 8 RAM
- 16-bit auto reloadable counter/timer
- Fixed-rate timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, industrial applications, and state machine applications
- Field programmable using standard equipment

DESCRIPTION

The Philips 87C751 offers many of the advantages of the 80C51 architecture in a small package and at low cost

The 87C751 Microcontroller is fabricated with Philips high-density CMOS technology. Philips expitaxial substrate minimizes CMOS latch-up sensitivity

The 87C751 contains a 2K x 8 EPROM, 64 × 8 RAM, 19 I/O lines, a 16-bit auto-reload counter/timer, a fixed-rate timer, a five-source fixed-priority interrupt structure, a bidirectional Inter-Intergrated Circuit (I^2C) serial bus interface, and an on-chip oscillator

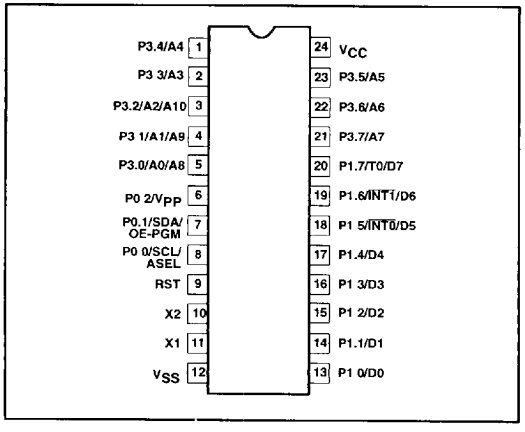
The onboard inter-integrated circuit (I^2C) bus interface allows the 87C751 to operate as a master or slave device on the I^2C on the I^2C small area network This capability facilitates I/O and RAM expansion, access to EEPROM, processor-to-processor communication, and efficient interface to a wide variety of dedicated I^2C peripherals.

ORDERING INFORMATION

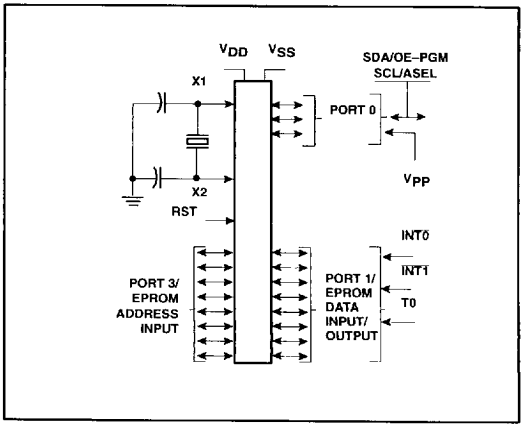
DESCRIPTION	ORDER CODE	PKG DESIGNATOR*
24-Pin ceramic Dual In-line Package (DIP) 300mil-wide with quartz window	87C751/BLA 87C751-16/BLA	GDIP3-T24
24-Pin ceramic Dual In-line Package (DIP) 300mil-wide without quartz window	87C751/BLA OT[1] 87C751-16/BLA OT[1]	GDIP3-T24

NOTE:
* MIL-STD 1835 or Appendix A of 1995 Military Data Handbook

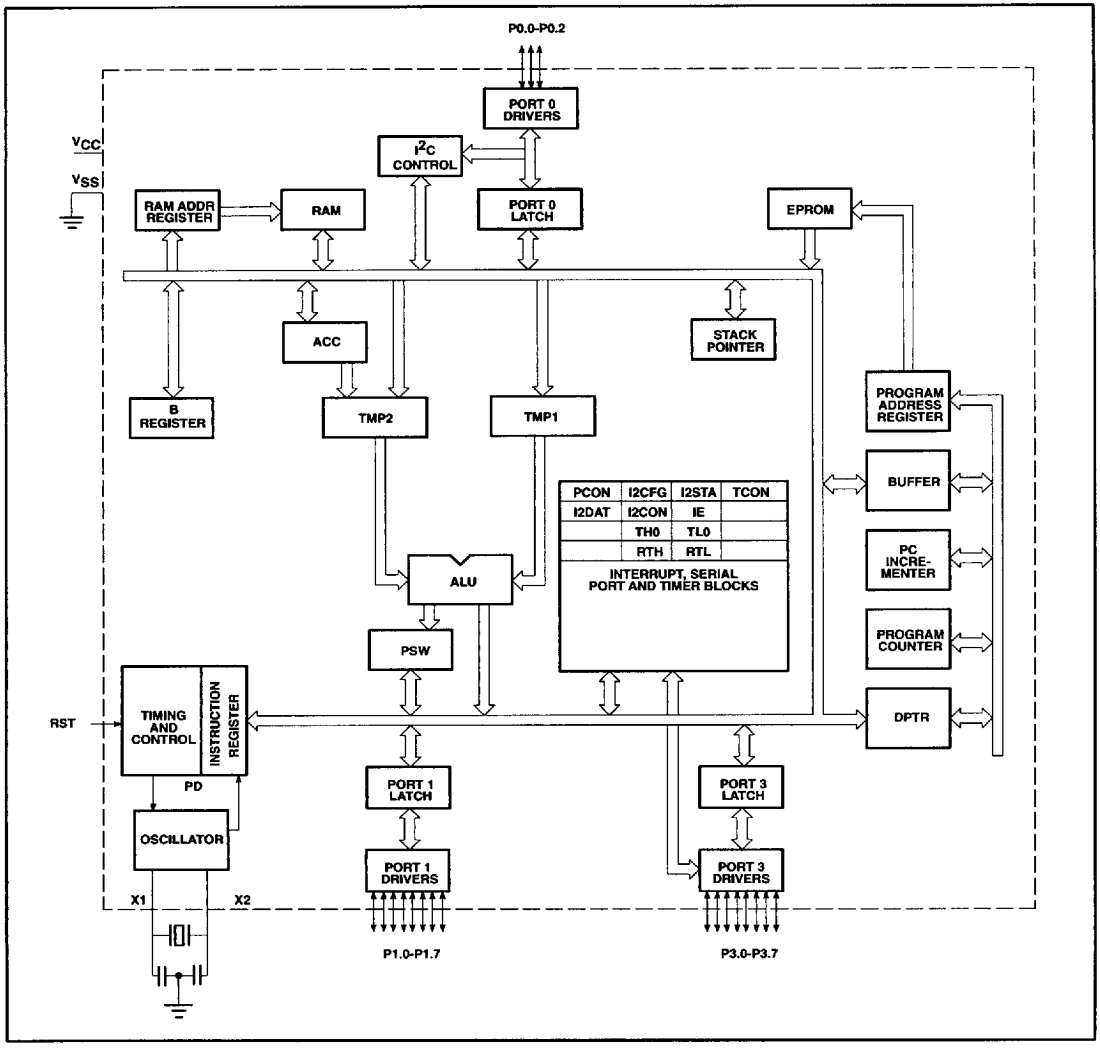
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN CONFIGURATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V_{SS}	12	I	**Circuit ground potential.**
V_{CC}	24	I/O	**Supply voltage during normal, idle, and power-down operation.**
P0.0-P0.2	8-6	I/O	Port 0: Port 0 is a 3-bit open-drain bidirectional port. Port 0 pins that have ones written to them float, and in that state can be used as high-impedance inputs. Port 0 also serves as the serial I^2C interface as shown in the pinout diagram. When this feature is activated by software, SCL and SDA are driven low in accordance with the 12C protocol. These pins are driven low if the port register bit is written with a 0 or if the I^2C subsystem presents a 0 The state of the pin can always be read from the port register by the program.
	7	I/O	To comply with the I^2C specification, P0.0 and P0.1 are open drain bidirectional I/O pins with the electrical characteristics listed in the tables that follow. While these differ from 'standard TTL' characteristics, they are close enough for the pins to still be used as general-purpose I/O in non-I^2C applications.
	8	I/O	**SPA (P0.1)** I^2C data. **SCL (P0.0)** I^2C clock Port 0 also provides alternate functions for programming the EPROM memory as follows:
	8	O	**P0.0/ASEL** - output which selects which byte of the EPROM address is to be applied to port 3. P0.0 - 0 presents the low address byte to port 3. P0 0 - 1 presents the high address byte to port 3 (only the three least significant bits are used).
	7	I	**P0.1/OE/PGM** - input, OE/PGM, which specifies verify mode (output enable) or the program mode OE/PGM - 1 output enabled (verify mode) OE/PGM - 0 program mode.
	6		**P0.2/V_{PP}** Programming voltage input.
P1.0-P1.7	13-20	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have ones written to them are pulled high by the internal pullups and can be used as inputs As inputs, port 1 pins that are externally pulled low will source current because of the internal pullups. (See DC electrical characteristics: I_{IL}). Port 1 also serves the special function features of the 80C51 family as listed below:
	18		**$\overline{INT0}$ (P1.5):** External interrupt
	19		**$\overline{INT1}$ (P1.6):** External interrupt
	20		**T0 (P1.7):** Timer 0 external input
			Port 1 serves to output the addressed EPROM contents in the verify mode and accepts as inputs the value to program into the selected address during the program mode.
P3.0-P3.7	5-1 23-21	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have ones written to them are pulled and can be used as inputs. As inputs, port 3 pins that are external ly being pulled low will source current because of the pullups (See DC electrical characteristics: I_{IL}) Port 3 also functions as the address input for the EPROM memory location to be programmed (or verified) The 11-bit address is multiplexed into this port as specified by P0.0.
RST	9	I	**Reset:** A high on this pin for two machine cycles while the oscillator is running resets the device An internal diffused resistor to V_{SS} permits a power-on RESET using only an external capacitor to V_{CC}. After the device is reset, a 10-bit serial sequence, sent LSB first, applied to RESET places the device in the programming state allowing programming address, data and V_{PP} to be applied for programming or verification purposes. The RESET serial sequence must be synchronized with the X1 input.
X1	11	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. X1 also serves as the clock to strobe in a serial bit stream into RESET to place the device in the programming state.
X2	10	O	**Crystal:** Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTIC

X1 and X2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, X1 should be driven while X2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

IDLE MODE

In the idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power- down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. The control bits for the reduced power modes are in the special function register PCON.

Table 4. External Pin Status during Idle and Power-down Modes

Mode	Port 0	Port 1	Port 2
Idle	Data	Data	Data
Power-down	Data	Data	Data

DESIGN CONSIDERATIONS

At power-down, the voltage on V_{CC} and RST must come up at the same time for a proper start-up. Before entering the power-down mode, the contents of the carry bit and B.7 must be equal (See Table 4).

DIFFERENCES BETWEEN THE 87C751 AND THE 80C51

Program Memory

On the 87C751, program memory is 2048 bytes long and is not externally expandable. Program memory can contain 87C751 instructions and constant data. The only fixed locations in program memory are the addresses at which execution is taken up in response to reset and to interrupts, which are as follows:

Event	Program Memory Address
Reset	000
External $\overline{INT0}$	003
Counter/timer 0	00B
External $\overline{INT1}$	013
Timer I	01B
I^2C serial	023

Counter/Timer Subsystem

The 87C751 has one counter/timer called timer/counter 0. Its operation is similar to mode 2 operation on the 80C51, but is extended to 16 bits with 16 bits of autoload. The controls for this counter are centralized in a single register called TCON.

A watchdog timer, called Timer I, is for use with the I^2C subsystem. In I^2C applications, this timer is dedicated to time-generation and bus monitoring of the I^2C. In non-I^2C applications, it is available for use as a fixed timebase.

Interrupt Subsytem - Fixed Priority

The IP register and the 2-level interrupt system of the 80C51 are eliminated. Simultaneous interrupt conditions are resolved by a single-level, fixed priority as follows:

Highest priority:	Pin $\overline{INT0}$
	Counter/timer flag 0
	Pin $\overline{INT1}$
	Timer I
Lowest priority:	Serial I^2C

Serial Communications

The 87C751 contains an I^2C serial communications port instead of the 80C51 UART. The I^2C serial port is a single bit hardware interface with all of the hardware necessary to support multimaster and slave operations. Also included are receiver digital filters and timer (timer I) for communication watch dog purposes. The I^2C serial port is controlled through four special function registers; I^2C control, I^2C data, I^2C status, and I^2C configuration.

Special Function Register Addresses

Special function register addresses for the 87C751 are identical to those of the 80C51, except for the changes listed below:

80C51 special function registers not present in the 87C751 are TMOD (89), P2 (A0) and IP (B8). The 80C51 registers TH1, TL1, SCON, and SBUF are replaced with the 87C751 registers RTH, RTL, I2CON, and 12DAT, respectively. Additional special function registers are 12CFG-(D8) and 12STA-(FB).

Table 5. I^2C Special Function Register Addresses

Register Address			Bit Address							
Name	Symbol	Address	MSB							LSB
I^2C control	I2CON	98	9F	9E	9D	9C	9B	9A	99	98
I^2C data	12DAT	99	–	–	–	–	–	–	–	–
I^2C configuration	12CFG	D8	DF	DE	DD	DC	DB	DA	D9	D8
I^2C status	12STA	F8	FF	FE	FD	FC	FB	FA	F9	F8

ABSOLUTE MAXIMUM RATINGS[2, 3]

SYMBOL	PARAMETER	RATING	UNIT
T_{amb}	Operating ambient temperature range[3]	-55 to +125	°C
T_{STG}	Storage temperature range	-65 to + 150	°C
V_{CC}	Voltage from V_{CC} to V_{SS}[4], except pin 6	-0.5 to +6 5	V
V_{CC}	Voltage from Pin 6 to V_{SS}[4]	-0.5 to +13.0	V
V_S	Voltage from any pin to V_{SS}[4]	-0 5 to V_{CC} +0.5	V
P_D	Power dissipation	1.0	W

DC ELECTRICAL CHARACTERISTICS

$-55°C \le T_A \le 125°C$, $4.5V \le V_{CC} \le 5.5V$, $V_{SS} = 0V$[4].

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
V_{IL} V_{IH} V_{IH1}	Input low voltage, except SDA, SCL Input high voltage, except X1, RST Input high voltage, X1, RST		-0.5 $0.2V_{CC}$+1.1 $0.7V_{CC}$+.2	$0.2V_{CC}$-0.1 V_{CC}+0.5 V_{CC}+0.5	V V V
 V_{IL1} V_{IH2}	SDA, SCL: Input low voltage Input high voltage		 -0.5 $0.7V_{CC}$	 $0.3V_{CC}$ V_{CC}+0 5	 V V
V_{OL} V_{OL1}	Output low voltage, ports 1 and 3 Output low voltage, port 0.2	I_{OL} = 1.6mA I_{OL} = 3.2mA		0 45 0.45	V V
V_{OH}	Output high voltage, ports 1 and 3	I_{OH} = -60µA I_{OH} = -25µA I_{OH} = -10µA	2.4 $0.75V_{CC}$ $0.9V_{CC}$		V V V
 V_{OL2}	Port 0.0 and 0.1 (I^2C) - Drivers Output low voltage	 I_{OL} = 3mA (over V_{CC} range)		 0 4	 V
I_{IL} I_{TL} I_{LI}	Logical 0 input current, ports 1 and 3 Logical 1 to 0 transition current, ports 1 and 3 Input leakage current, port 0	V_{IN} = 0 45V V_{IN} = 2V $0.45 < V_{IN} < V_{CC}$		75 -750 ±10	µA µA µA
R_{RST}	Reset pull-down resistor		25	175	kΩ
C_{IO}[9]	Pin capacitance	Test freq - 1MHz, T_A - 25°C		10	pF
C[9]	Driver, receiver combined: Capacitance	Test freq – 1MHz, T_A = 25°C		10	pF
I_{PD}	Power-down current[5]	V_{CC} = 3 to 5.5V		50	µA
V_{PP}[10]	V_{PP} program voltage	V_{SS} = 0V V_{CC} = 5V±10% T_A = 21°C - 27°C	12 5	13.0	V
I_{PP}[10]	Program current	V_{PP} = 13.0V		10	mA
I_{CC}[11]	Supply current	See Figure 18			

AC SYMBOL DESIGNATIONS

Each timir g symbol has five characters. The first character is always 't' (= time) The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

C – Clock
D – Input rlata
H – Logic level high
L – Logic level low
Q – Output data
T – Time
V – Valid
X – No longer a valid logic level
Z – Float

AC ELECTRICAL CHARACTERISTICS

$-55°C \leq T_A \leq 125°C$, $4.5 \leq V_{CC} \leq 5.5V_{SS} = 0V$[4, 8]

SYMBOL	PARAMETER		12MHz		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	Oscillator frequency	87C751			3.5	12	MHz
		87C751-16			3.5	16	MHz
	External clock (Figure 16)						
t_{CHCX}	High time		20				ns
t_{CLCX}	Low time		20				ns
t_{CLCH}	Rise time[10]			20			ns
t_{CHCL}	Fall time[10]			20			ns

NOTES:
1. Erase characteristics do not apply for one time programming (OT).
2. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified. All voltages with respect to V^{SS} unless otherwise noted.
5. Power-down Icc is measured with all output pins disconnected; port 0 = V_{CC}; X2, X1 n.c.; RST =V_{SS}.
6. I_{CC} is measured with all output pins disconnected; X1 driven with t_{CLCH}, t_{CHCL} = 5ns, $V_{IL} = V_{SS} + 0.5V$, $V_{IH} = V_{CC} - 0.5V$; X2 n.c.; RST = port 0 = Vcc. Icc will be slightly higher if a crystal is used.
7. Idle I_{CC} is measured with all output pins disconnected; X1 driven with t_{CLCH}, t_{CHCL} = 5ns, $V_{IL} = V_{SS} + 0.5V$, $V_{IH} = V_{CC} - 0.5V$; X2 n.n; port 0 = V_{CC}; RST = V_{SS}.
8. Load capacitance for ports = 80pF.
9. C_{IO} is tested initially and after each design or process changes which may effect capacitance.
10. Guaranteed but not tested.
11. Tested at maximum frequency only.

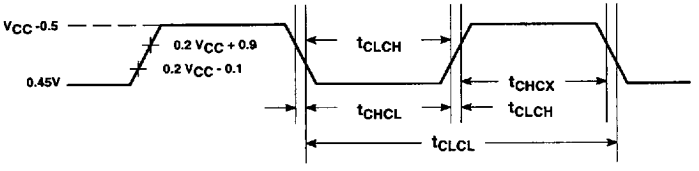


Figure 16. External Clock Drive Waveform

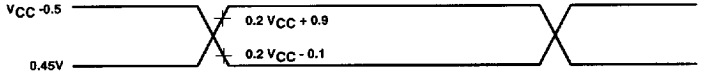
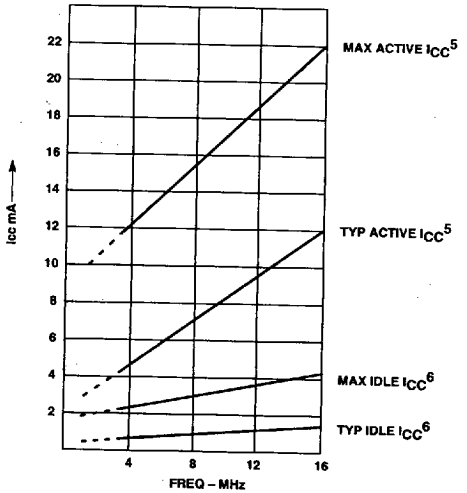


Figure 17. AC Testing Input/Output Waveform



Maximum I_{CC} values taken at V_{CC} = 5.5V and worst case temperature. Typical I_{CC} values taken at V_{CC} = 5.0V and 25°C.
Notes 5 and 6 refer to AC Electrical Characteristics.

Figure 18. I_{CC} versus Frequency

PROGRAMMING CONSIDERATIONS

EPROM Characteristics

The 87C751 is programmed by using a modified Quick-Pulse Programming algorithm similar to that used for devices such as the 87C451 and 87C51. It differs from these devices in that a serial data stream is used to place the 87C751 in the programming mode.

Figure 19 shows a block diagram of the programming configuration for the 87C751. Port pin PO.2 is used as the programming voltage supply input (V_{PP} signal). Port pin PO.1 is used as the program (PCM/) signal. This pin is used for the 25 programming pulses.

Port 3 is used as the address input for the byte to be programmed and accepts both the high and low components of the eleven bit address. During one portion of the programming cycle, Port 3 will accept the low component of the address (bits A0 - A7) and during another portion of the cycle, the high component of the address (bits A8 - A10). The timing of the high and low components is derived internal to the 87C751 and P0.0 is provided as an address select signal (ASEL) to indicate when the device is ready to accept which portion of the address. By using tri-statable drivers as shown in Figure 18, this multiplexing of the address components is transparent to the user. P0.0 is an open drain output and a pullup resistor is required. P0.0 is low for the lower byte of the address and high for the high order bits of the address.

Port 1 is used as a bidirectional data bus during programming mode, it accepts the byte to be programmed. During verify mode, it provides the contents of the EPROM location specified by the address which has been supplied to Port 3.

The XTAL1 pin is the ocsillator input and receives the master system clock. This clock should be between 4 and 6 MHz.

The RESET pin is used to accept the serial data stream that places the 87C751 into various programming modes. This pattern consists of a 10-bit code with the LSB sent first. Each bit is synchronized to the clock input, X1.

Programming Operation

Figures 20 and 21 show the timing diagrams for the program/verify cycle. RESET should initially be held high for at least two machine cycles. P0.1 (PGM/) and P0.2 (V_{PP}) will be at V_{OH} as a result of the RESET operation. At this point, these pins function as normal quasi-bidirectional I/O ports and the programming equipment may pull these lines low. However, prior to sending the 10 bit code on the reset pin, the programming equipment should drive these pins high (V_{IH}). The RESET pin may now be used as the serial data input for the data stream which places the 87C751 in the programming mode. Data bits are sampled during the clock high time and thus should only change during the time that the clock is low. Following transmission of the last data bit, the RESET pin should be held low.

Next the address information for the location programmed is placed at the inputs of the buffers used to perform the address multiplexing function. At this time Port 1 functions as an output.

A high voltage V_{PP} level is then applied to the V_{PP} input (P0.2). (This sets Port 1 as an input port.) The data to be programmed into the EPROM array is then placed on Port 1. This is followed by a series of programming pulses applied to the PGM/ pin (P0.1). These pulses are created by driving P0.1 low and then high. This pulse is

repeated until a total of twenty-five programming pulses have occurred. At the conclusion of the last pulse, the PGM/ signal should remain high.

The V_{PP} signal may now be driven to the V_{OH} level, placing the 87C751 in the verify mode. (Port 1 is now used as an output port.) After 4 machine cycles (48 clock periods) the contents of the addressed location in the EPROM array will appear on Port 1.

The next programming cycle may now be initiated by placing the address information at the inputs of the multiplexed buffers, driving the V_{PP} pin to the V_{PP} voltage level, providing the byte to be programmed to Port 1 and issuing the 25 programming pulses on the PGM/ pin, bringing V_{PP} back down to the V_{CC} level and verifying the byte.

Programming Modes

The 87C751 has four programming features incorporated within its EPROM array. These include the USER EPROM for storage of the application's code, a sixteen byte encryption KEY array and two security bits. Programming and verification of the these four elements are selected by a combination of the serial data stream applied to the RESET pin and the voltage levels applied to port pins P0.1 and P0.2. The various combinations are shown in Table 1.

Encryption Key Table

The 87C751 includes a 16 byte EPROM array that is programmable by the end user. The contents of this array can then be used to encrypt the program memory contents during a program memory verify operation. When a program memory verify operation is performed, the contents of the program memory location is XNOR'ed with one of the bytes in the 16 byte encryption table. The resulting data pattern is then provided to port 1 as the verify data. The encryption mechanism can be disabled, in essence, by leaving the bytes in the encryption table in their erased state (FFH) since the XNOR product of a bit with a logical one will result in the original bit. The encryption bytes are mapped with the code memory in 16 byte groups. The first byte in code memory will be encrypted with the first byte in the encryption table; the second byte in code memory will be encrypted with the second byte in the encryption table and so forth up to and including the sixteenth byte. The encryption repeats in 16 byte groups; the seventeenth byte in code memory will be encrypted with the first byte in the encryption table, and so forth.

Security Bits

Two security bits, security bit 1 and security bit 2, are provided to limit access to the USER EPROM and encryption key arrays, Security bit 1 is the program inhibit bit, and once programmed performs the following functions:

3. Additional programming of the USER EPROM is inhibited.
4. Additional programming of the encryption is inhibited.
5. Verification of the encryption key is inhibited.
6. Verification of the USER EPROM and the security bit levels may still be performed.

(If the encryption key array is being used, the security bit should be programmed by the user to prevent unauthorized parties from reprogramming the encryption key to all logical zero bits. Such programming would provide data during a verify cycle that is the logical compliment of he USER EPROM contents.)

Security bit 2, the verify inhibit bit, prevents verification of both the USER EPROM array and the encryption key arrays. The security bit levels may still be verified.

Programming and Verifying Security Bits

Security bits are programmed employing the same techniques used to program the USER EPROM and KEY arrays using serial data streams and logic levels on port pins indicated in Table 3. When programming either security bit, it is not necessary to provide address or data information to the 87C751 on ports 1 and 3.

Verification occurs in a similar manner using the RESET serial stream shown in Table 1. Port 3 is not required to be driven and the results of the verify operation will appear on ports 1.6 and 1.7.

Port 1.7 contains the security bit 1 data and is a logical one if programmed and logical zero i erased. Likewise, P1.6 contains the security bit 2 data and is a logical one if programmed and a logical zero if erased.

Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. **For this and secondary effects, it is recommended that an opaque label be placed over the window.** For elevated temperature of solvent environments, use Kapton tape Fluorglas part number 2345-5 or equivalent.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 angstroms) to an integrated dose at least 15W-sec/cm^2. Exposing the EPROM to and ultraviolet lamp of 12,000^W/cm^2 rating for 20 to 39 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

Table 6. Implementing Program/Verify Modes

Operation	Serial Code	P0.1 (PGM/)	P0.2 (V_{PP})
Program user EPROM	296H	~*	V_{PP}
Verify user EPROM	296H	V_{IH}	V_{IH}
Program key EPROM	292H	~*	V_{PP}
Verify key EPROM	292H	V_{IH}	V_{IH}
Program security bit 1	29AH	~*	V_{PP}
Program security bit 2	298H	~*	V_{PP}
Verify security bits	29AH	V_{IH}	V_{IH}

NOTE: Pulsed from V_{IH} to V_{IL} and returned to V_{IH}.

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T_A = 21°C TO +27°C, V_{CC} = 5V±10, V_{SS} = 0V

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN	MAX	
$1/t_{CLCL}$	Oscillator/clock frequency	4	6	MHz
t_{AVGL}*	Address setup to P0.1 (PROG-) Low	$10\mu s + 24t_{CLCL}$		
t_{GHAX}	Address hold after P0.1 (PROG-) High	$48t_{CLCL}$		
t_{DVGL}	Data setup to P0.1 (PROG-) Low	$38t_{CLCL}$		
t_{GHDX}	Data hold after P0.1 (PROG-) High	$36t_{CLCL}$		
t_{SHGL}	V_{PP} setup to P0.1 (PROG-) Low	10		µs
t_{GHSL}	V_{PP} hold after P0.1 (PROG-)	10		µs
t_{GLGH}	P0.1 (PROG-) width	90	110	µs
t_{AVQV}**	V_{PP} Low (V_{CC}) to data valid		$48t_{CLCL}$	
t_{GHGL}	P0.1 (PROG-) High to P0.1 (PROG-) Low	10		µs
t_{SYNL}	P0.0 (sync pulse) Low	$4t_{CLCL}$		
t_{SYNH}	P0.0 (sync pulse) High	$8t_{CLCL}$		

NOTES:
* Address should be valid at least 24 t_{CLCL} before rising edge of P0.2 (V_{PP}).
** For a pure verify mode, i.e., no program mode in between, t_{AVQV} is 14 t_{CLCL} maximum.

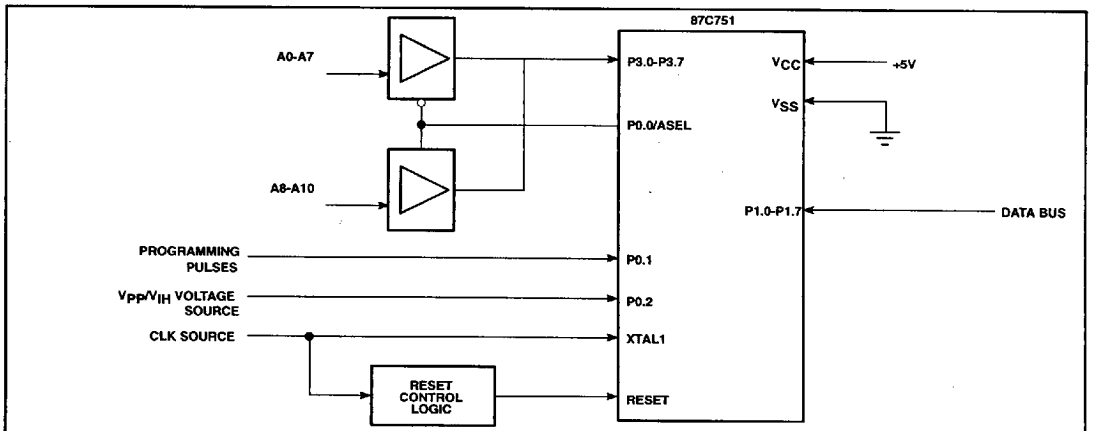


Figure 19. Programming Configuration

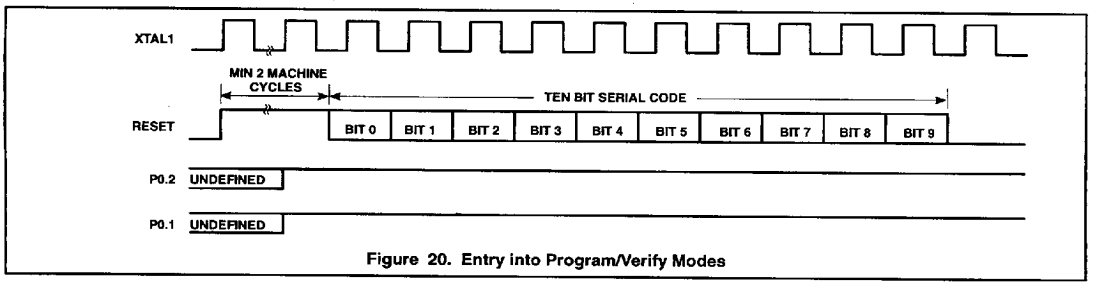


Figure 20. Entry into Program/Verify Modes

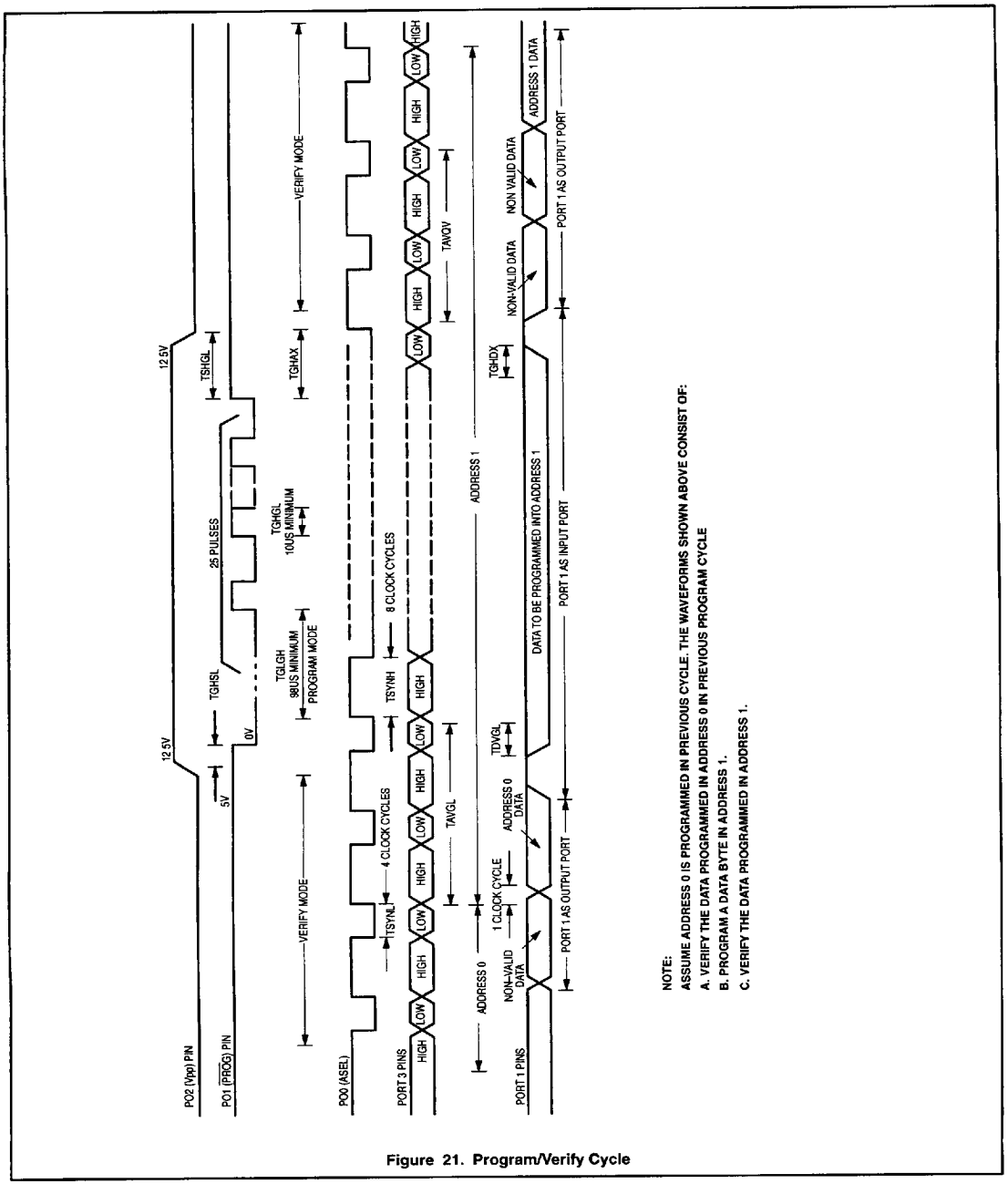


Figure 21. Program/Verify Cycle

INSTRUCTION SET

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Arithmetic Operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A,@Ri	Add indirect RAM to accumulator	1	1
ADD	A,#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A,#data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w/borrow	1	1
SUBB	A,#data	Subtract immediate data from A w/borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1
Logical Operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A, Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORI	A,@Ri	OR indirect RAM to accumulator	1	1
ORI	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate A left through the carry flag	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate A right through carry flag	1	1
SWAP	A	Swap nibbles within the accumulator	1	1
Data Transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1

INSTRUCTION SET (Continued)

MNEMONIC		DESCRIPTION	BYTE	CYCLES
Data Transfer (cont)				
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to A	1	2
MOVC	A,@A + PC	Move code byte relative to PC to A	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
SCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order digit indirect RAM with A	1	1
Boolean variable manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and machine control				
ACALL	addr11	Absolute subroutine call	2	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP*	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if carry flag not set	2	2
JB	bit,rel	Jump if direct bit set	3	2
JNB	bit,rel	Jump if direct bit not set	3	2
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3	2
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3	2
CJNE	Rn,#data,rel	Compare immediate to register and jump if not equal	3	2
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	2
DJNZ	Rn,rel	Decrement register and jump if not zero	2	2
DJNZ	direct,rel	Decrement direct and jump if not zero	3	2
NOP		No operation	1	1

Notes on following page.

Notes on data addressing modes:

Rn Working register R0-R7.
direct 64 internal RAM locations, any I/O port, control or status register.
@Ri indirect internal RAM location addressed by register R0 or R1.
#data 8-bit constant included in instruction.
#data16 16-bit constant included as bytes 2 and 3 of instruction.
bit 128 software flags, any I/O pin, control or status bit.

Notes on program addressing modes:

addr16 Destination address for LJMP. Only 11 bits of this 16-bit field are used.
addr11 Destination address for ACALL and AJMP will be within the same 2Kbyte page of program memory as the first byte of the following instruction.
rel SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 - 128 bytes relative to first byte of the following instruction.

INSTRUCTION OPCODES IN HEXADECIMAL ORDER

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS	HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
00	1	NOP		2C	1	ADD	A,R4
01	2	AJMP	code addr	2D	1	ADD	A,R5
02	3	LJMP*		2E	1	ADD	A,R6
03	1	RR	A	2F	1	ADD	A,R7
04	1	INC	A	30	3	JNB	bit addr,code addr
05	2	INC	data addr	31	2	ACALL	code addr
06	1	INC	@R0	32	1	RETI	
07	1	INC	@R1	33	1	RLC	A
08	1	INC	R0	34	2	ADDC	A,#data
09	1	INC	R1	35	2	ADDC	A,data addr
0A	1	INC	R2	36	1	ADDC	A,@R0
0B	1	INC	R3	37	1	ADDC	A,@R1
0C	1	INC	R4	38	1	ADDC	A,R0
0D	1	INC	R5	39	1	ADDC	A,R1
0E	1	INC	R6	3A	1	ADDC	A,R2
0F	1	INC	R7	3B	1	ADDC	A,R3
10	3	JBC	bit addr,code addr	3C	1	ADDC	A,R4
11	2	ACALL	code addr	3D	1	ADDC	A,R5
12	3	LCALL*		3E	1	ADDC	A,R6
13	1	RRC	A	3F	1	ADDC	A,R7
14	1	DEC	A	40	2	JC	code addr
15	2	DEC	data addr	41	2	AJMP	code addr
16	1	DEC	@R0	42	2	ORL	data addr,A
17	1	DEC	@R1	43	3	ORL	data addr,#data
18	1	DEC	R0	44	2	ORL	A,#data
19	1	DEC	R1	45	2	ORL	A,data addr
1A	1	DEC	R2	46	1	ORL	A,@R0
1B	1	DEC	R3	47	1	ORL	A,@R1
1C	1	DEC	R4	48	1	ORL	A,R0
1D	1	DEC	R5	49	1	ORL	A,R1
1E	1	DEC	R6	4A	1	ORL	A,R2
1F	1	DEC	R7	4B	1	ORL	A,R3
20	3	JB	bit addr,code addr	4C	1	ORL	A,R4
21	2	AJMP	code addr	4D	1	ORL	A,R5
22	1	RET		4E	1	ORL	A,R6
23	1	RL	A	4F	1	ORL	A,R7
24	2	ADD	A,#data	50	2	JNC	code addr
25	2	ADD	A,data addr	51	2	ACALL	code addr
26	1	ADD	A,@R0	52	2	ANL	data addr,A
27	1	ADD	A,@R1	53	3	ANL	data addr,#data
28	1	ADD	A,R0	54	2	ANL	A,#data
29	1	ADD	A,R1	55	2	ANL	A,data addr
2A	1	ADD	A,R2	56	1	ANL	A,@R0
2B	1	ADD	A,R3	57	1	ANL	a,@r1

INSTRUCTION OPCODES IN HEXADECIMAL ORDER (Continued)

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS	HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
58	1	ANL	A,R0	8F	2	MOV	data addr,R7
59	1	ANL	A,R1	90	3	MOV	DPTR,#data
5A	1	ANL	A,R2	91	2	ACALL	code addr
5B	1	ANL	A,R3	92	2	MOV	bit addr,C
5C	1	ANL	A,R4	93	1	MOVC	A,@A + DPTR
5D	1	ANL	A,R5	94	2	SUBB	A,#data
5E	1	ANL	A,R6	95	2	SUBB	A,data addr
5F	1	ANL	A,R7	96	1	SUBB	A,@R0
60	2	JZ	code addr	97	1	SUBB	A,@R1
61	2	AJMP	code addr	98	1	SUBB	A,R0
62	2	XRL	data addr,A	99	1	SUBB	A,R1
63	3	XRL	data addr,#data	9A	1	SUBB	A,R2
64	2	XRL	A,#data	9B	1	SUBB	A,R3
65	2	XRL	A,data addr	9C	1	SUBB	A,R4
66	1	XRL	A,@R0	9D	1	SUBB	A,R5
67	1	XRL	A,@R1	9E	1	SUBB	A,R6
68	1	XRL	A,R0	9F	1	SUBB	A,R7
69	1	XRL	A,R1	A0	2	ORL	C,/bit addr
6A	1	XRL	A,R2	A1	2	AJMP	code addr
6B	1	XRL	A,R3	A2	2	MOV	C,bit addr
6C	1	XRL	A,R4	A3	1	INC	DPTR
6D	1	XRL	A,R5	A4	1	MUL	AB
6E	1	XRL	A,R6	A5		reserved	
6F	1	XRL	A,R7	A6	2	MOV	@R0,data addr
70	2	JNZ	code addr	A7	2	MOV	@R1,data addr
71	2	ACALL	code addr	A8	2	MOV	R0,data addr
72	2	ORL	C,bit addr	A9	2	MOV	R1,data addr
73	1	JMP	@A + DPTR	AA	2	MOV	R2,data addr
74	2	MOV	A,#data	AB	2	MOV	R3,data addr
75	3	MOV	data addr,#data	AC	2	MOV	R4,data addr
76	2	MOV	@R0,#data	AD	2	MOV	R5,data addr
77	2	MOV	@R1,#data	AE	2	MOV	R6,data addr
78	2	MOV	R0,#data	AF	2	MOV	R7,data addr
79	2	MOV	R1,#data	B0	2	ANL	C,/bit addr
7A	2	MOV	R2,#data	B1	2	ACALL	code addr
7B	2	MOV	R3,#data	B2	2	CPL	bit addr
7C	2	MOV	R4,#data	B3	1	CPL	C
7D	2	MOV	R5,#data	B4	3	CJNE	A,#data,code addr
7E	2	MOV	R6,#data	B5	3	CJNE	A,data addr,code addr
7F	2	MOV	R7,#data	B6	3	CJNE	@R0,#data,code addr
80	2	SJMP	code addr	B7	3	CJNE	@R1,#data,code addr
81	2	AJMP	code addr	B8	3	CJNE	R0,#data,code addr
82	2	ANL	C,bit addr	B9	3	CJNE	R1,#data,code addr
83	1	MOVC	A,@A + PC	BA	3	CJNE	R2,#data,code addr
84	1	DIV	AB	BB	3	CJNE	R3,#data,code addr
85	3	MOV	data addr,data addr	BC	3	CJNE	R4,#data,code addr
86	2	MOV	data addr,@R0	BD	3	CJNE	R5,#data,code addr
87	2	MOV	data addr,@R1	BE	3	CJNE	R6,#data,code addr
88	2	MOV	data addr,R0	BF	3	CJNE	R7,#data,code addr
89	2	MOV	data addr,R1	C0	2	PUSH	data addr
8A	2	MOV	data addr,R2	C1	2	AJMP	code addr
8B	2	MOV	data addr,R3	C2	2	CLR	bit addr
8C	2	MOV	data addr,R4	C3	1	CLR	C
8D	2	MOV	data addr,R5	C4	1	SWAP	A
8E	2	MOV	data addr,R6	C5	2	XCH	A,data addr

INSTRUCTION OPCODES IN HEXADECIMAL ORDER (Continued)

HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS	HEX CODE	NUMBER OF BYTES	MNEMONIC	OPERANDS
C6	1	XCH	A,@R0	E3	1	MOVX*	
C7	1	XCH	A@R1	E4	1	CLR	A
C8	1	XCH	A,R0	E5	2	MOV	A,data addr
C9	1	XCH	A,R1	E6	1	MOV	A,@R0
CA	1	XCH	A,R2	E7	1	MOV	A,@R1
CB	1	XCH	A,R3	E8	1	MOV	A,R0
CC	1	XCH	A,R4	E9	1	MOV	A,R1
CD	1	XCH	A,R5	EA	1	MOV	A,R2
CE	1	XCH	A,R6	EB	1	MOV	A,R3
CF	1	XCH	A,R7	EC	1	MOV	A,R4
D0	2	POP	data addr	ED	1	MOV	A,R5
D1	2	ACALL	code addr	EE	1	MOV	A,R6
D2	2	SETB	bit addr	EF	1	MOV	A,R7
D3	1	SETB	C	F0	1	MOVX*	
D4	1	DA	A	F1	2	ACALL	code addr
D5	3	DJNZ	data addr,code addr	F2	1	MOVX*	
D6	1	XCHD	A,@R0	F4	1	CPL	A
D7	1	XCHD	A,@R1	F5	2	MOV	data addr,A
D8	2	DJNZ	R0,code addr	F6	1	MOV	@R0,A
D9	2	DJNZ	R1,code addr	F7	1	MOV	@R1,A
DA	2	DJNZ	R2,code addr	F3	1	MOVX*	
DB	2	DJNZ	R3,code addr	F8	1	MOV	R0,A
DC	2	DJNZ	R4,code addr	F9	1	MOV	R1,A
DD	2	DJNZ	R5,code addr	FA	1	MOV	R2,A
DE	2	DJNZ	R6,code addr	FB	1	MOV	R3,A
DF	2	DJNZ	R7,code addr	FC	1	MOV	R4,A
E0	1	MOVX*		FD	1	MOV	R5,A
E1	2	AJMP	code addr	FE	1	MOV	R6,A
E2	1	MOVX*		FF	1	MOV	R7,A

NOTE:

* Missing 80C51 instructions are not applicable to the 87C751. Should these instructions be executed, the appropriate number of ROM bytes will be fetched, but no operation will take place. The use of these codes is not recommended.

The addr16 field is allowed for compatibility with existing software tools, however only the least significant 11 bits of the address field will be used.