

DATA SHEET

80C31/80C51

CMOS SINGLE-CHIP 8 BIT MICROCONTROLLER

80C31/80C51 : 0 TO 12 MHz 80C31/80C51-1 : 0 TO 16 MHz 80C31-S/80C51-S : 0 TO 20 MHz 80C31/80C51-L:0 TO 6 MHz WITH 2.7 V < V_{CC} < 6 V 80C51F:80C51 WITH PROTECTED ROM

FEATURES

- POWER CONTROL MODES
- . 128 x 8 BIT RAM
- 4 K BYTES OF ROM (80C51)
- 32 PROGRAMMABLÉ VO LÍNES
- TWO 16 BIT TIMER/COUNTER
- 64 K PROGRAM MEMORY SPACE
- FULLY STATIC DESIGN

- BOOLEAN PROCESSOR
- 5 INTERRUPT SOURCES
- PROGRAMMABLE SERIAL PORT
- **64 K DATA MEMORY SPACE**
- TEMPERATURE RANGE: COMMERCIAL, INDUSTRIAL, AUTOMOTIVE AND MILITARY

INTRODUCTION

MHS's 80C31 and 80C51 are high performance CMOS versions of the 8031/8051 NMOS single chip 8 bit μ C and are manufactured using a self-aligned silicon gate CMOS process (SAJI VI).

The fully static design of the MHS 80C31/80C51 allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The 80C51 retains all the features of the 8051:4 K bytes of ROM; 128 bytes of RAM; 32 I/O lines; two 16 bit timers; a 5-source, 2-level interrupt structure; a full

duplex serial port; and on-chip oscillator and clock circuits.

In addition, the 80C51 has two software-selectable modes of reduced activity for further reduction in power consumption. In the Idle Mode the CPU is frozen while the RAM, the timers, the serial port, and the interrupt system continue to function. In the Power Down Mode the RAM is saved and all other functions are inoperative.

The 80C31 is identical to the 80C51 except that it has no on-chip ROM.

The information contained herein is subject to change without notice. No responsibility is assumed by MATRA MHS SA for using this publication and/or circuits described herein; nor for any possible infringements of patents or other rights of third parties which may result from its use.

80C31/Rev.1.0

5-3

. 5868456 0003710 254

5

INTERFACE

PIN CONFIGURATION

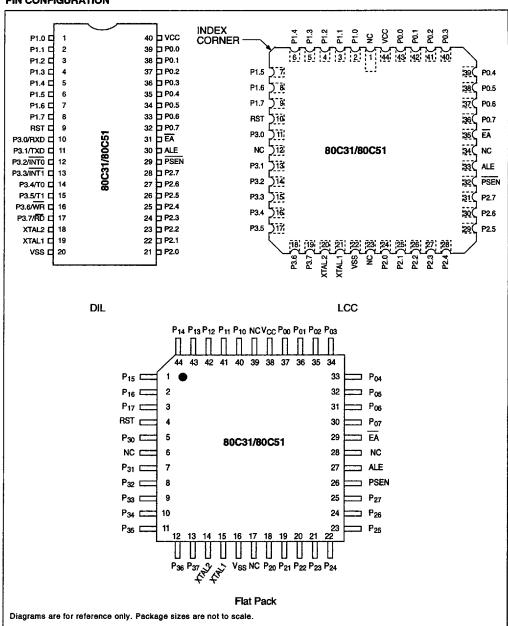


Figure 1.



80C31/Rev.1.0

■ 5868456 0003711 190 **■**

PIN DESCRIPTION

Vss

Circuit ground potential

Vcc

Supply voltage during normal, Idle, and Power Down operation.

Port 0

Port 0 is an 8 bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

Port 1

Port 1 is an 8 bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (III., on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address byte during program verification. In the 80C51, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 2

Port 2 is an 8 bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (III., on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8 bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during program verification in the 80C51. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 3

Port 3 is an 8 bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the

function of various special features of the MHS 51 Family, as listed below.

Alternate Function
RXD (serial input port)
TXD (serial output port)
INTO (external interrupt 0)
INT1 (external interrupt 1)
T0 (Timer 0 external input)
T1 (Timer 1 external input)
WR (external Data Memory write strobe)
RD (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST

A high level on this for two machine cycles while the oscillator is running resets the device. An internal pull-down resistor permits Power-On reset using only a capacitor connected to $V_{\rm CC}$.

ALE

Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time on ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

PSFN

Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

FΔ

When EA is held high, the CPU executed out of internal Program Memory (unless the Program Counter exceeds 0FFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1

Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2

Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

80C31/Rev.1.0

MHS

FUNCTIONAL DESCRIPTION

BLOCK DIAGRAM

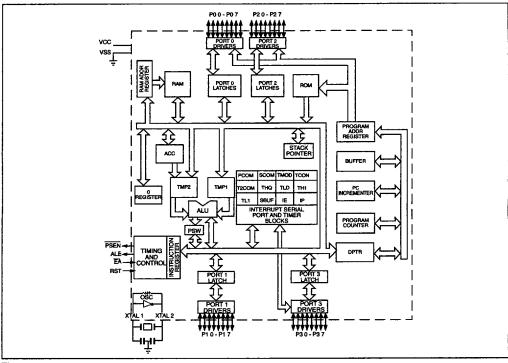


Figure 2.

IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off.

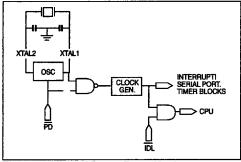


Figure 3: idle and Power Down Hardware.

PCON: Power Control Register

(MSB)							(LSB)
SMOD	_	_	_	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
_	PCON.6	(Reserved)
_	PCON.5	(Reserved)
_	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit
IDL	PCON.0	activates power down operation. Idle mode bit. Setting this bit activates idle mode operation.



80C31/Rev.1.0

5868456 0003713 T63

These special modes are activated by software via the Special Function Register, its hardware address is 87H. PCON is not bit addressable.

If 1's are written to PD and IDL at the same time. PD takes precedence. The reset value of PCON is (0XXX0000).

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other register maintain their data during Idle. Table 1 describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following the instruction that wrote 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiates the Special Function Register (see *Table 1*).

In the Power Down mode, V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released un²⁷ the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in Figure 4.

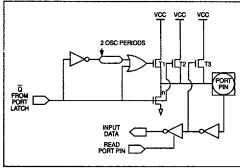


Figure 4: I/O Buffers in the 80C51 (Ports 1, 2, 3).

STOP CLOCK MODE

Due to static design, the MHS 80C31/C51 clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

I/O PORTS

The I/O port drive of the 80C51 is similar to the 8051. The I/O buffers for Ports 1, 2 and 3 are implemented as shown in figure 4.

When the port latch contains a 0, all pFETS in figure 4 are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the loh source current. This inverter and T3 form a latch which holds the 1 and is supported by T2.

When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

MODE	PROGRAM MEMORY	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
ldle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Down	External	0	0	Floating	Port Data	Port Data	Port Data

Table 1 : Status of the ex. all pins during Idle and Power Down modes.

80C31/Rev.1.0





OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in figure 5. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in figure 6. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

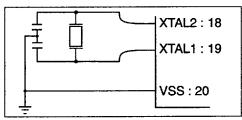


Figure 5 : Crystal Oscillator.

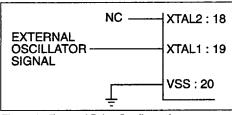


Figure 6: External Drive Configuration.

80C51 WITH PROTECTED ROM

MHS provides a new member in the 80C51 Family named "80C51F" which permits full protection of the internal ROM contents.

With a non protected 80C51, it is very easy to read out the contents of the internal 4 K bytes of ROM.

Three methods exist, two of them are special test modes and the last one is by means of MOVC instructions.

 Test mode "VER": Using this special test mode, the internal ROM contents are output on port P0; the address being applied on ports P2 (AD15...AD8) and P1 (AD7...AD0).

- Test mode "TMB": With this second test mode, the contents of the 80C51 internal bus is presented on port P1 during the PH2 clock phases.
- Using MOVC instructions: If EA = 0, and following a reset, the 80C51 fetches its instructions from external program memory. It is then possible to write a small program whose purpose is to dump the internal ROM contents by means of MOVC A, @A + DPTR and MOVC A, @A + PC instructions.

80C51F with program protection features

This version adds ROM protection features in some strategic points of the 80C51F in order to eliminate the possibility of reading the ROM contents (once the protection has been programmed) by one if the three forementioned methods (VER and TMB test modes, or MOVC instructions).

Nevertheless the customer must note the following:

- Once the protection has been programmed, the 80C51F program always starts at address 0 in the internal ROM.
- The application program must be self contained in the internal 4 K of ROM, otherwise it would be possible to trap the program counter address in the external PROM/EPROM (beyond 4 K) and then to dump the internal ROM contents by means of a patch using MOVC instructions.

Thus, if an extra EPROM is necessary, it is advised to ensure that it will contain only constants or tables.

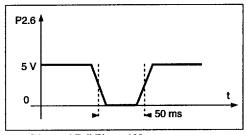
Test of the on-chip program memory

- Before protection is activated: The 80C51F can be tested as any normal 80C51 (using test equipment or any other methods).
- After protection is activated: It is then no longer possible to dump the internal ROM contents.

How to program the protection mechanism

- To burn correctly the fuse a specific configuration of inputs must be settled as below:
 - RST = ALE = 1
 - -P2.7 = 1

Furthermore PSEN signal must be tied at $+9 \text{ V} \pm 5 \%$ level voltage and a pulse must be applied on P2.6 input Port. The timing on P2.6 is shown below :



Time Rise and Fall Rise \leq 100 μ s.

• The electrical schematic shows a typical application to deliver P2.6 signal.

MHS

80C31/Rev.1.0

5868456 0003715 836

5

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Ambiant Temperature Under Bias :

* Notice

Stresses at or 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

 $T_A = 0 \text{ to } + 70^{\circ}\text{C}$; $V_{CC} = 5 \text{ V} \pm 20 \text{ %}$; $V_{SS} = 0 \text{ V}$; F = 0 to 16 MHz. $V_{CC} = 5 \text{ V} \pm 10 \text{ %}$; $V_{SS} = 0 \text{ V}$; F = 16 to 20 MHz. $T_A = -40 \text{ to } 85^{\circ}\text{C}$; $V_{CC} = 5 \text{ V} \pm 10 \text{ %}$; $V_{SS} = 0 \text{ V}$; F = 0 to 16 MHz.

PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
Input Low Voitage	– 0.5 V	0.2 VCC - 0.1	٧	
Input High Voltage (Except XTAL and RST)	0.2 VCC + 0.9	VCC + 0.5	٧	
Input High Voltage (RST and XTAL1)	0.7 VCC	VCC + 0.5	٧	
Output Low Voltage		0.3	٧	IOL = 100 μA
(Ports 1, 2, and 3)		,		IOL = 1.6 mA (note 3)
			-	IOL = 3.5 mA
			•	IOL = 200 μA
(Port U, ALE, PSEN)			V	IOL = 3.2 mA (note 3)
Output High Voltage Ports 1, 2, 3	Vcc - 0.3		V	IOH = - 10 μA
	Vcc - 0.7		٧	IOH = - 30 μA
	Vcc 1.5		٧	IOH = - 60 μA VCC = 5 V ± 10 %
Output High Voltage	Vcc - 0.3		V	IOH = - 200 μA
(Port 0, ALE, PSEN)	Vcc - 0.7		٧	IOH = - 3.2 mA
	V∞ – 1,5		٧	IOH = - 7.0 mA VCC = 5 V ± 10 %
Logical 0 Input Current (Ports 1, 2, 3)		C -50 1 -60	μА	Vin = 0.45 V
Input Leakage Current (Port 0, EA)		± 10	μА	0.45 < Vin < VCC
Logical 1 to 0 Transition Current (Ports 1, 2, 3)		- 650	μА	Vin = 2.0 V
Power Down Current		50	μА	VCC = 2.0 V to 6 V (note 2)
RST Pulldown Resistor	50	150	kΩ	
Capacitance of I/O Buffer		10	pF	f _c = 1 MHz, T _A = 25°C
Power Supply Current				(notes 1, 2)
· · · · · · · · · · · · · · · · · · ·		20	mA	
20 MHz		8	mA	
	Input Low Voltage Input High Voltage (Except XTAL and RST) Input High Voltage (RST and XTAL 1) Output Low Voltage (Ports 1, 2, and 3) Output Low Voltage (Port 0, ALE, PSEN) Output High Voltage Ports 1, 2, 3 Output High Voltage Ports 1, 2, 3 Output High Voltage Ports 1, 2, 3 Input Leakage Current (Port 0, EA) Logical 0 Input Current (Port 0, EA) Logical 1 to 0 Transition Current (Ports 1, 2, 3) Power Down Current RST Pulldown Resistor Capacitance of I/O Buffer Power Supply Current Active Mode 12 MHz 16 MHz Idle Mode 12 MHz 16 MHz	Input Low Voltage Input High Voltage (Except XTAL and RST) Input High Voltage (RST and XTAL1) Output Low Voltage (Ports 1, 2, and 3) Output Low Voltage (Port 0, ALE, PSEN) Output High Voltage Ports 1, 2, 3 Vcc – 0.7 Vcc – 1.5 Output High Voltage (Port 0, ALE, PSEN) Output High Voltage (Port 0, ALE, PSEN) Output High Voltage (Port 0, ALE, PSEN) Toc – 0.3 Vcc – 0.7 Vcc – 1.5 Logical 0 Input Current (Ports 1, 2, 3) Input Leakage Current (Port 0, EA) Logical 1 to 0 Transition Current (Ports 1, 2, 3) Power Down Current RST Pulldown Resistor Capacitance of I/O Buffer Power Supply Current Active Mode 12 MHz 16 MHz 10 MHz 11 MHz 11 MHz 11 MHz 11 MHz 11 MHz 11 MHz 12 MHz 13 MHz 14 MHz 15 MHz	Input Low Voltage	Input Low Voltage

80C31/Rev.1.0





ABSOLUTE MAXIMUM RATINGS*

Ambiant Temperature Under	Bias :
A = Automotive	
Storage Temperature	65°C to + 150°C
Voltage on Any Pin to Vss	0.5 V to VCC + 0.5 V
Voltage on Vcc to Vss	0.5 V to 6.5 V
Power Dissipation	

* Notice

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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

 $T_A = -40^{\circ}$ to 125°C; $V_{CC} = 5 \text{ V} \pm 10 \text{ %}$; $V_{SS} = 0 \text{ V}$; F = 0 to 12 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	- 0.5 V	0.2 VCC - 0.1	٧	
VIH	Input High Voltage (Except XTAL1, RST)	0.2 VCC + 0.9	VCC + 0.5	٧	
VIH1	Input High Voltage (XTAL1, RST)	0.7 VCC	VCC + 0.5	V	
VOL	Output Low Voltage (Ports 1, 2, and 3)		0.3 0.45 1.0	V V V	IOL = 100 μA IOL = 1.6 mA (note 3) IOL = 3.5 mA
VOL1	Output Low Voltage (Port 0, ALE, PSEN)		0.3 0.45 1.0	٧	IOL = 200 μA IOL = 3.2 mA (note 3) IOL = 7.0 mA
VOH	Output High Voltage (Ports 1, 2, 3)	Vcc - 0.3		V	IOH = - 10 μA
		Vcc - 0.7		٧	IOH = - 30 μA
		Vcc - 1.5		٧	IOH = - 60 μA
VOH1	Output High Voltage	Vcc - 0.3		٧	IOH = - 200 μA
	(Port 0 in External Bus Mode, ALE, PSEN)	Vcc - 0.7		٧	IOH = - 3.2 mA
		Vcc - 1,5		٧	IOH = -7.0 mA
IIL	Logical 0 Input Current Ports 1, 2, 3		- 75	μА	Vin = 0.45 V
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		- 750	μА	Vin = 2.0 V
ILI	Input Leakage Current (Port 0, EA)		± 10	μА	0.45 < Vin < VCC
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Pin Capacitance		10	рF	Test Freq = 1 MHz, T _A = 25°C
IPD	Power Down Current		75	μА	VCC = 2 V to 5.5 V (note 2)
ICC	Power supply current Active mode 12 MHz Idle mode 12 MHz		21 7	mA mA	VCC = 5.5 V (notes 1, 2 VCC = 5.5 V (notes 1, 2



80C31/Rev.1.0

ABSOLUTE MAXIMUM RATINGS*

Ambiant Temperature Under Bias :

M =Military	55°C to + 125°C
Storage Temperature	
Voltage on Any Pin to Vss.	
Voltage on Vcc to Vss	
Power Dissipation	

* Notice

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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended period may affect device reliability.

DC CHARACTERISTICS

 $T_A = -55^{\circ}$ to 125°C; VSS = 0 V; VCC = 5 V ± 10 %; F = 0 to 12 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	- 0.5 V	0.2 VCC - 0.1	٧	
VIH	Input High Voltage (Except XTAL1, RST)	0.2 VCC + 0.9	VCC + 0.5	٧	
VIH1	Input High Voltage (XTAL1, RST)	0.7 VCC	VCC + 0.5	٧	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	٧	IOL = 1.6 mA (note 3)
VOL1	Output Low Voltage (Port 0, ALE, PSEN)		0.45	٧	IOL = 3.2 mA (note 3)
VOH	Output High Voltage (Ports 1, 2, 3)	2.4		٧	IOH = ~ 60 μA VCC = 5 V ± 10 %
		0.75 VCC		٧	IOH = - 25 μA
		0.9 VCC		٧	IOH = - 10 μA
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, PEN)	2.4		٧	IOH = - 800 μA VCC = 5 V ± 10 %
		0.75 VCC		٧	IOH = - 300 μA
		0.9 VCC		٧	IOH = – 80 μA
IIL	Logical 0 Input Current Ports 1, 2, 3		- 75	μA	Vin = 0.45 V
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		- 750	μΑ	Vin = 2 V
ILI	Input Leakage Current (Port 0, EA)		± 10	μА	0.45 < Vin < VCC
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Pin Capacitance		10	рF	Test Freq = 1 MHz, T _A = 25°C
IPD	Power Down Current		75	μA	VCC = 2 V to 5.5 V (note 2)
ICC	Power supply current Active mode 12 MHz Idle mode 12 MHz		21 7	mA mA	VCC = 5.5 V (notes 1, 2) VCC = 5.5 V (notes 1, 2)

80C31/Rev.1.0





ABSOLUTE MAXIMUM RATINGS*

Ambiant Temperature Under Bias:	
C = commercial	
I = industrial	40°C to +85°C
Storage Temperature	
Voltage on Vcc to Vss	
Voltage on Any Pin to Vss 0	
Power Dissipation	
"This value is based on the max	cimum allowable die

temperature and the thermal resistance of the package

* Notice

Stresses at or 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 above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

 $T_A = -40^{\circ}$ to 85°C; $V_{CC} = 2.7$ V to 6 V; VSS = 0 V; F = 0 to 6 MHz

SYMBOL	PARAMETER	MIN	MAX	UNIT	TEST CONDITIONS
VIL	Input Low Voltage	- 0.5 V 0.2 Vcc - 0.1		٧	
VIH	Input High Voltage (Except XTALs and RST)	0.2 V _{CC} + 0.9			
VIH1	Input High Voltage to RST for Reset	0.7 Vcc	Vcc + 0.5	V	
VIH2	Input High Voltage to XTAL1	0.7 Vcc	V _∞ + 0.5	V	
VPD	Power Down Voltage to Vcc in PD Mode	2.0	6.0	V	
VOL	Output Low Voltage (Ports 1, 2, and 3)		0.45		IOL = 800 μA (note 3)
VOL1	Output Low Voltage (Port 0, ALE, PSEN)	•	0.45	٧	IOL = 1.6 mA (note 3)
VOH	Output High Voltage Ports 1, 2, 3	0.9 Vcc		٧	IOH = - 10 μA
VOH1	Output High Voltage (Port 0 in External Bus Mode), ALE, PEN	0.9 Vcc		V	IOH = - 80 μA
IIL	Logical 0 Input Current Ports 1, 2, 3		C -50	ДΑ	
			I -6	0	Vin = 0.45 V
L	Input Leakage Current		± 10	μА	0.45 < Vin < VCC
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		- 650	μА	Vin = 2.0 V
IPD	Power Down Current		50	μА	V _{CC} = 2.0 V to 6 V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f _c = 1 MHz, T _A = 25°C

MAXIMUM Icc (mA)

· · ·	OP	ERATING (NOT	TE 3)		IDLE (NOTE 4)	
FREQ. VCC	2.7 V	5 V	6 V	2.7 V	5 V	6 V
1 MHz 6 MHz	0.8 mA 4 mA	1.5 mA 8 mA	1.8 mA 10 mA	400 μA 1.2 mA	800 μA 3.5 mA	1 mA 3.8 mA

5-12



80C31/Rev.1.0

5868456 0003719 481 📟

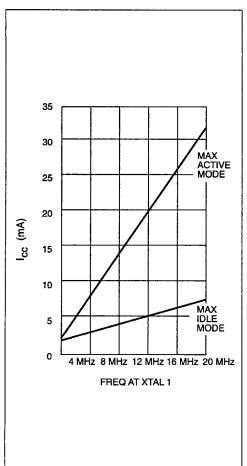


Figure 1 : ICC vs. Frequency. Valid only within frequency specifications of the device under test.

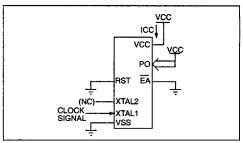


Figure 12 : ICC Test Condition, Idle Mode.
All other pins are disconnected.

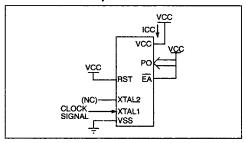


Figure 13 : ICC Test Condition, Active Mode. All other pins are disconnected.

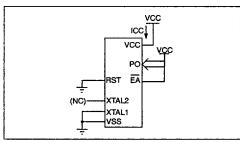


Figure 15 : ICC Test Condition, Power Down Mode. All other pins are disconnected.

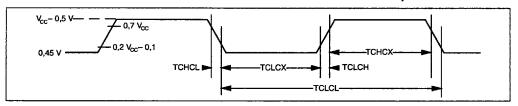


Figure 14: Clock Signal Waveform for ICC Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

M M M S

80C31/Rev.1.0

Note 1: ICC max is given by:

Active Mode: ICCMAX = 1.47 x FREQ + 2.35 Idle Mode: ICCMAX = 0.33 x FREQ + 1.05

where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See figures 1 through 5 for ICC test conditions.

Note 2: ICC is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS + .5 V, VIH = VCC - .5 V; XTAL2 N.C.; EA = RST = Port 0 = VCC. ICC would be slightly higher if a crystal oscillator used.

Idle ICC is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, VIL = VSS +

EXPLANATION OF THE AC SYMBOL

Each timing symbol has 5 characters. The first character is always a "T" (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

.5 V, VIH = VCC - .5 V; XTAL2 N.C; Port 0 = VCC; EA = RST = VSS.

Power Down ICC is measured with all output pins disconnected; EA = PORT 0 = VCC; XTAL2 N.C.; RST = VSS.

Note 3: Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VoLS of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45 V may exceed 0,45 V with maxi VOL peak 0.6 V. A Schmitt Trigger use is not necessary.

Example:

TAVLL = Time for Address Valid to ALE low.
TLLPL = Time for ALE low to PSEN low.

A : Address.

C : Clock. D : Input data.

H : Logic level HIGH.

i : Instruction (program memory contents).

L : Logic level LOW, or ALE.

P : PSEN.

Q : Output data.

R: READ signal.

T: Time.

V : Valid. W : WRITE signal.

X : No longer a valid logic level.

Z: Float.

AC PARAMETERS

 $T_A = 0 \text{ to } + 70 \text{ °C}$; VSS = 0 V; VCC = 5 V \pm 20 %; 0 to 16 MHz $T_A = 0 \text{ to } + 70 \text{ °C}$; VSS = 0 V; VCC = 5 V \pm 10 %; 16 to 20 MHz

 $T_A = -40 \text{ to} + 85 ^{\circ}\text{C}$; VSS = 0 V; VCC = 5 V ± 10 %; 0 to 16 MHz

 $T_A = -55 \text{ to} + 125 \text{ °C}$; VSS = 0 V; VCC = 5 V ± 10 %; 0 to 12 MHz

(Load Capacitance for PORTO, ALE and PSEN = 100 pf; Load Capacitance for all other outputs = 80 pf.)

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

OVALDOL	DADAMETED	0 TO 12 MHz		16 MHz		20 MHz	
SYMBOL	PARAMETER	MIN	MAX	MIN	MAX	MIN	MAX
TLHLL	ALE pulse width	2TCLCL-40		110		70	
TAVLL	Address Valid to ALE	TCLCL-40		30		25	
TLLAX	Address Hold After ALE	TCLCL-30		35		25	
TLLIV	ALE to Valid Instr In		4TCLCL-100		185		140_
TLLPL	ALE to PSEN	TCLCL-30		45		30	
TPLPH	PSEN Pulse Width	3TCLCL-45		165		130	
TPLIV	PSEN to Valid Instr IN		3TCLCL-105		125		80
TPXIX	Input Instr Hold After PSEN	0		0		0	
TPXIZ	Input Instr Float After PSEN		TCLCL-25		22		10
TPXAV	PSEN to Address Valid	TCLCL-8		55		45	
TAVIV	Address to Valid Instr In		5TCLCL-105		230		180
TPLAZ	PSEN Low to Address Float		10		10		10

5-14

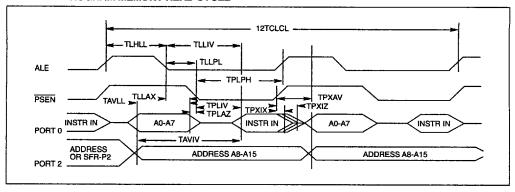


80C31/Rev.1.0

5868456 0003721 03T **=**

<u>5</u>

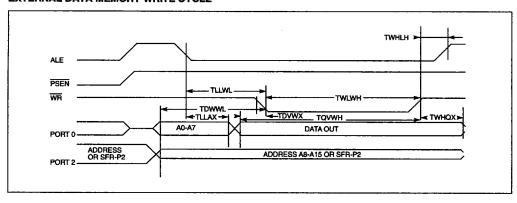
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY CHARACTERISTICS

SYMBOL	PARAMETER	0 TO 1	2 MHz 16 MHz		MHz	z 20 MHz	
31 MDCL	PANAMETER	MIN	MAX	MIN	MAX	MIN	MAX
TRLRH	RD Pulse Width	6TCLCL-100		340		260	
TWLWH	WR Pulse Width	6TCLCL-100		340		260	
TRLDV	RD to Valid Data In		5TCLCL-165		240		180
TRHDX	Data Hold After RD	0		0		0	
TRHDZ	Data Float After RD		2TCLCL-60		90		70
TLLDV	ALE to Valid Data In		8TCLCL-150		435		360
TAVDV	Address to Valid Data In		9TCLCL-165		480		400
TLLWL	ALE to WR or RD	3TCLCL-50	3TCLCL+50	150	250	125	185
TAVWL	Address to WR or RD	4TCLCL-130		180		170	
TQVWX	Data Valid to WR Transition	TCLCL-50		15		10	
TQVWH	Data Set-Up to WR High	7TCLCL-150		380		310	
TWHQX	Data Hold After WR	TCLCL-50		40		30	
TRLAZ	RD Low to Address Float		0		0		0
TWHLH	RD or WR High to ALE High	TCLCL-40	TCLCL+40	35	90	30	65

EXTERNAL DATA MEMORY WRITE CYCLE



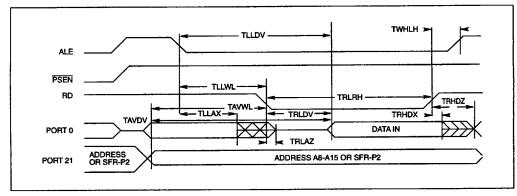
80C31/Rev.1.0

5-15

M M-IS

■ 5868456 0003722 T76 **■**

EXTERNAL DATA MEMORY READ CYCLE

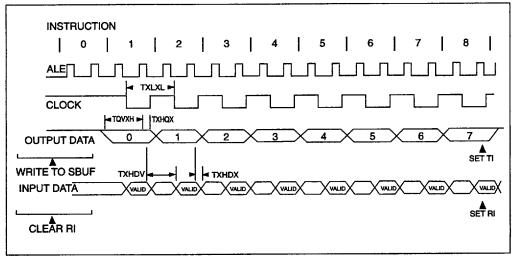


SERIAL PORT TIMING - SHIFT REGISTER MODE

 $\begin{array}{l} T_A = 0 \text{ to} + 70 \text{ °C} \text{ ; VSS} = 0 \text{ V ; VCC} = 5 \text{ V} \pm 20 \text{ % ; 0 to } 16 \text{ MHz} \\ T_A = 0 \text{ to} + 70 \text{ °C} \text{ ; VSS} = 0 \text{ V ; VCC} = 5 \text{ V} \pm 10 \text{ % ; } 16 \text{ to } 20 \text{ MHz} \\ T_A = -40 \text{ to} + 85 \text{ °C} \text{ ; VSS} = 0 \text{ V ; VCC} = 5 \text{ V} \pm 20 \text{ % ; 0 to } 16 \text{ MHz} \\ T_A = -55 \text{ to} + 125 \text{ °C} \text{ ; VSS} = 0 \text{ V ; VCC} = 5 \text{ V} \pm 10 \text{ % ; 0 to } 12 \text{ MHz} \\ \end{array}$

0)////001	DADAMETED	0 TO 12 MHz		16 MHz		20 MHz	
SYMBOL.	PARAMETER	MIN	MAX	MIN	MAX	MIN	MAX
TXLXL	Serial port clock cycle time	12TCLCL		750		600	
TQVHX	Output data setup to clock rising edge	10TCLCL-133		563		450	
TXHQX	Output data hold after clock rising edge	2TCLCL-117		63	}	50	
TXHDX	Input data hold after clock rising edge	0		0		0	
TXHDV	Clock rising edge to input data valid		10TCLCL-133		563		450

SHIFT REGISTER TIMING WAVEFORMS



MAS

5-16

80C31/Rev.1.0

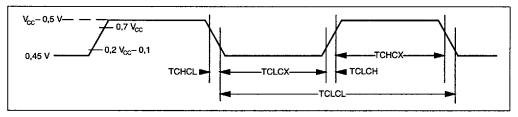
■ 5868456 0003723 902 ■

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL1)

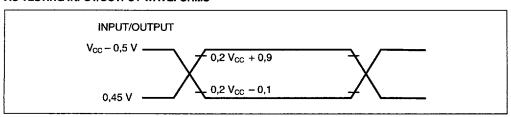
SYMBOL	PARAMETER	М	IN	M	AX	UNIT
TCLCL	Oscillator Period	50	(5)			ns
TCHCX	High Time	20	(5)			ns
TCLCX	Low Time	20	(5)			ns
TCLCH	Rise Time			20	(5)	ns
TCHCL	Fall Time			20	(5)	ns

(5) AT 20 MHz

EXTERNAL CLOCK DRIVE WAVEFORMS

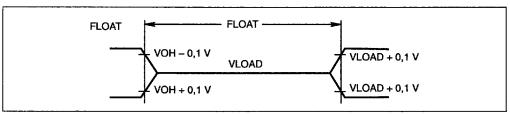


AC TESTING INPUT/OUTPUT WAVEFORMS



AC inputs during testing are driven at $V_{CC} - 0.5$ for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at VIH min for a logic "1" and VIL max for a logic "0".

FLOAT WAVEFORMS

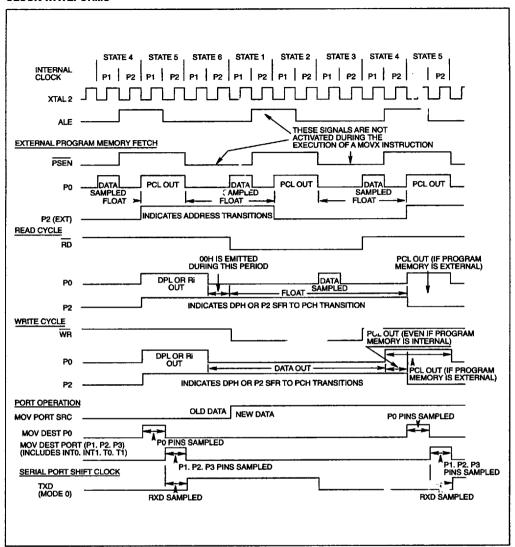


For timing purposes as port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. $IoV/IoH \ge \pm 20$ mA.

80C31/Rev.1.0



CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though (T_A = 25°C fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

5-18



80C31/Rev.1.0

5

INSTRUCTION OPCODES

MHS C51 INSTRUCTION SET DESCRIPTION

ARTHMETIC OPERATIONS MNEMONIC DESCRIPTION BYTE CYC		UCTION SET DESC	AIT HON		
ADD A, Rinc Add registrer to Accumulator 1 1 1 ADD A, direct Add direct bytes to Accumulator 2 1 1 ADD A, ØRi Add indirect RAM to Accumulator 1 1 ADD A, #data Add inmediate data of Accumulator 2 1 ADDC A, Rin Add registrer to Accumulator with Carry 1 1 ADDC A, Griect Add direct byte to A with Carry 1 1 ADDC A, ØRi Add inmediate data to A with Carry 1 1 ADDC A, ØRi Add indirect RAM to A with Carry 1 1 ADDC A, ØRi Add indirect RAM to A with Carry 1 1 1 ADDC A, ØRi Add indirect RAM to A with Carry 1 1 1 ADDC A, ØRi Add indirect RAM to A with Carry 1 1 1 ADDC A, ØRi Add indirect RAM to A with Carry 1 1 1 ADDC A, ØRi Add indirect Subtract direct byte 1 1 1 SUBB A, direct Subtract direct byte 1 1 1 SUBB A, direct Subtract direct byte 1 1 1 SUBB A, ØRi Subtract inmed. data from A with Borrow 2 1 INC A Increment Accumulator 1 1 1 INC A Increment Register 1 1 1 INC direct Increment indirect RAM 1 1 1 INC direct Increment indirect RAM 1 1 1 INC DPTR Increment Data Pointer 2 1 INC DPTR Increment Data Pointer 1 2 DEC A Decrement Accumulator 1 1 1 DEC direct Decrement Accumulator 1 1 1 DEC direct Decrement Indirect RAM 1 1 1 DEC direct A RA RAD Indirect RAM 1 1 1 DEC direct A RAD Indirect RAM 1 1 1 DEC direct A RAD Indirect RAM 1 1 1 1 DEC direct A RAD Indirect RAM 1 1 1 1 DEC direct A RAD Indirect RAM 1 1 1 1 DEC di	1	OPERATIONS	B-50 amin-mail:		
ADD					CYC
ADD A, @Ri Add indirect RAM to Accumulator 1 1 1 ADDC A, #data Add immediate data of Accumulator 2 1 1 1 ADDC A, Rn Add register to Accumulator with Carry 1 1 1 1 ADDC A, @Ri Add indirect RAM to A with Carry 1 1 1 1 ADDC A, @Ri Add indirect RAM to A with Carry 1 1 1 1 ADDC A, @Ri Add indirect RAM to A with Carry 1 1 1 1 ADDC A, @Ri Add indirect RAM to A with Carry 1 1 1 1 ADDC A, #data Add immediate data to A with Carry 1 1 1 1 ADDC A, #data Add immediate data to A with Carry 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			Add registrer to Accumulator		_
ADD					
ADDC		•			
ADDC A, direct Add direct byte to A with Carry flag 1 ADDC A, @Ri Add indirect RAM to A with Carry flag 1 1 ADDC A, #data Add indirect RAM to A with Carry flag 2 1 SUBB A, Fin Subtract register from A with borrow 1 1 SUBB A, direct Subtract direct byte from A with Borrow 2 1 SUBB A, data Subtract indirect RAM from A with Borrow 2 1 INC A Increment Accumulator 1 1 INC Rn Increment Accumulator 1 1 INC GPRI Increment Indirect RAM 1 1 INC DPTR Increment Indirect RAM 1 1 INC DPTR Increment Accumulator 1 1 INC DPTR Increment Accumulator 1 1 DEC A Decrement Accumulator 1 1 DEC Rn Decrement Egister 1 1				_	
ADDC		•		•	-
ADDC A, #data Add immediate data to A with Carry flag 2 1 SUBB A, Rn Subtract direct byte from A with borrow 1 1 SUBB A, direct Subtract direct byte from A with Borrow 2 1 SUBB A, data Subtract direct byte from A with Borrow 2 1 INC A Increment accumulator 1 1 INC A Increment eduction 1 1 INC Rn Increment direct byte 2 1 INC Bri Increment direct byte 2 1 INC Bri Increment direct BAM 1 1 INC Bri Increment Data Pointer 1 2 1 DEC A Decrement Data Pointer 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 </td <td></td> <td></td> <td></td> <td></td> <td></td>					
SUBB A, Rin Subtract register from A with borrow 1 SUBB A, direct Subtract direct byte from A with Borrow 2 SUBB A, @Ri Subtract direct byte from A with Borrow 1 SUBB A, data Subtract immed, data from A with Borrow 1 INC A Increment Accumulator 1 INC Rin Increment Accumulator 1 INC direct Increment direct byte 2 INC BRI Increment direct byte 2 INC DPTR Increment data Pointer 2 INC DPTR Increment Data Pointer 1 INC DPTR Increment Accumulator 1 INC DEC A Decrement Accumulator 1 INC DEC Rin Decrement register 1 INC DEC A Decrement register 1 INC DEC Gran Decrement indirect RAM 1 INC DEC Gran Decrement register 1 INC DEC Gran Decrement finderet RAM 1 INC DEC Gran Decrement register 1 INC DEC Gran Decrement indirect RAM 1 INC DEC Gran Decrement register 1 INC DEC Gran Decrement indirect RAM 1 INC DECREMENT IN INCREMENT IN INCREM					
SUBB A, direct Subtract direct byte from A with Borrow 2 1 SUBB A, @Ri Subtract infriect RAM from A with Borrow 1 1 1 INC A Increment Accumulator 1 1 1 INC Rn Increment Accumulator 1 1 1 INC direct Increment direct byte 2 1 INC @Ri Increment direct byte 2 1 INC @Ri Increment Data Pointer 1 1 2 INC @Ri Increment Data Pointer 1 1 2 INC DPTR Increment Data Pointer 1 1 2 INC Geri Increment Data Pointer 1 1 2 INC DEC A Decrement Accumulator 1 1 1 INC DEC A Decrement Accumulator 1 1 1 INC DEC Girect Decrement direct byte 2 1 INC Geri Decrement direct byte 2 1 1 INC Geri Decrement direct byte 3 1 4 INC Geri Decrement direct Dyte Accumulator 1 1 1 INC Geri Decrement direct Dyte 3 2 INC Geri Decrement d		A, #data	Add immediate data to A with Carry flag	2	
SUBB A, data Subtract indirect RAM from A with Borrow 1 1 1 SUBB A, data Subtract immed. data from A with Borrow 2 1 1 INC A Increment Accumulator 1 1 1 INC Rn Increment direct byte 2 1 INC Rn Increment direct BAM 1 1 INC DPTR Increment Data Pointer 1 2 DEC A Decrement Accumulator 1 1 1 INC DEC Rn Decrement Accumulator 1 1 1 INC DEC Rn Decrement Rnice RAM 1 1 INC DEC Rn Decrement Register 1 1 1 INC DEC Rn Decrement Rnice RAM 1 1 INC DEC Rn Decrement Indirect RAM 1 1 INC DEC RNICE RN	SUBB	A, Rn	Subtract register from A with borrow		
SUBB A data Subtract immed. data from A with Borrow 2 1 INC A Increment Accumulator 1 1 INC Rn Increment Accumulator 1 1 INC direct Increment direct byte 2 1 INC @Ri Increment indirect RAM 1 1 INC DPTR Increment Data Pointer 1 2 DEC A Decrement Accumulator 1 1 1 DEC direct Decrement direct byte 2 1 DEC Rn Decrement direct byte 2 1 DEC @Ri Decrement direct byte 2 1 DEC @Ri Decrement indirect RAM 1 1 1 DEC direct Decrement indirect RAM 1 1 1 DEC DEC @Ri Decrement indirect RAM 1 1 1 DEC DEC @Ri Decrement indirect RAM 1 1 1 DEC DEC @Ri Decrement indirect RAM 1 1 1 DEC DEC @Ri Decrement indirect RAM 1 1 1 DEC DEC @Ri Decrement indirect RAM 1 1 1 DEC DEC @Ri Decrement indirect RAM 1 1 1 DEC DEC @Ri Decrement direct byte 1 1 4 DIV AB Divide A by B 1 1 4 DIV AB Divide A by B 1 1 4 DIV AB Divide A by B 1 1 4 DIV AB Divide A by B 1 1 4 DA Decimal Adjust Accumulator 1 1 1 LOGICAL OPERATIONS MNEMONIC DESTINATION MIL A, direct AND direct byte to Accumulator 1 1 1 ANIL A, direct AND direct byte to Accumulator 1 1 1 ANIL A, #data AND immediate data to Accumulator 2 1 ANL A, #data AND immediate data to Accumulator 2 1 ANL direct, #data AND immediate data to Accumulator 2 1 ANL direct, #data AND immediate data to Girect byte 3 2 ORL A, Rn OR register to Accumulator 1 1 1 ORL A, #data OR immediate data to direct byte 3 2 AND Accumulator 1 1 1 ORL A, #data OR immediate data to direct byte 2 1 ORL A, #data OR immediate data to direct byte 2 1 ORL A, #data OR immediate data to direct byte 2 1 ORL A, #data OR immediate data to direct byte 2 1 AND Accumulator 0 1 1 1 AND Accumulator 0			Subtract direct byte from A with Borrow		
INC An Increment Accumulator 1 1 1 1 INC Rn Increment register 1 1 1 1 INC Rn Increment register 1 1 1 1 INC Rn Increment direct byte 2 2 1 INC @Ri Increment direct byte 2 1 INC DPTR Increment indirect RAM 1 1 1 1 INC DPTR Increment Indirect RAM 1 1 1 1 INC DPTR Increment Indirect RAM 1 1 1 1 INC DEC A Decrement Accumulator 1 1 1 INC DEC Rn Decrement Register 1 1 1 INC DEC Rn Decrement Girect byte 2 1 INC DEC GIRECT DECREMENT REGISTER 1 1 1 INDEC GIRECT DECREMENT REGISTER 1 1 1 INDEC GIRECT DECREMENT REGISTER 1 1 1 INDEC GIRECT DECREMENT REGISTER 1 1 1 INDECREMENT REGISTER 1 1 INDECREMENT REGISTER 1 1 INDECREMENT REGISTER 1 1 INDECREMENT REGISTER INDECREMENT REGI	SUBB	A, @Ri	Subtract indirect RAM from A with Borrow	1	1
INC Rn Increment register 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	SUBB	A, data	Subtract immed. data from A with Borrow	2	1
INC direct Increment direct byte 2 1 INC @Ri Increment indirect RAM 1 1 1 1 1 2 2 2 2 2	INC	Α	Increment Accumulator	1	
INC ØRI Increment indirect RAM 1 1 1 INC DPTR Increment Data Pointer 1 2 DEC A Decrement Accumulator 1 1 1 DEC Rn Decrement Accumulator 1 1 1 DEC Girect Decrement derect byte 2 1 1 MUL AR Multiply A & B 1 4 DIV AB Divide A by B 1 4 DA A Decimal Adjust Accumulator 1 1 1 LOGICAL OPERATIONS MNEMONIC DESTINATION BYTE CYC ANL A, Rn AND register to Accumulator 1 1 ANL A, direct AND direct byte to Accumulator 2 1 ANL A, #data AND immediate data to direct byte 2 1 ANL direct, #data AND immediate data to direct byte 3 2 ORL A, Rn OR register to Accumulator 2 1 ORL A, Girict OR direct byte to Accumulator 2 1 ORL A, Walta AND immediate data to direct byte 3 2 ORL A, Rn OR register to Accumulator 1 1 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Rn OR register to Accumulator 2 1 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Rn OR register to Accumulator 2 1 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Bn OR register to Accumulator 2 1 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Rn OR register to Accumulator 2 1 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Walta OR immediate data to direct byte 3 2 ORL A, Walta OR immediate data to direct byte 3 2 INCRL A, Walta OR immediate data to direct byte 3 2 INCRL A, Walta OR immediate data to direct byte 3 2 INCRL A, Walta OR immediate data to direct byte 3 2 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR immediate data to direct byte 3 3 INCRL A, Walta OR Immediate data to direct byte 3 3 INCRL A,	INC	Rn	Increment register	1	1
INC	INC	direct	Increment direct byte	2	1
DEC A Decrement Accumulator 1 1 1 DEC Rn Decrement register 1 1 1 DEC direct Decrement direct byte 2 1 DEC @Ri Decrement indirect RAM 1 1 1 MUL AR Multiply A & B 1 4 DIV AB Divide A by B 1 4 DA A Decimal Adjust Accumulator 1 1 1 LOGICAL OPERATIONS MNEMONIC DESTINATION BYTE CYC ANL A, Girect AND direct byte to Accumulator 1 1 1 ANL A, direct AND direct byte to Accumulator 2 1 ANL A, #data AND immediate data to Accumulator 2 1 ANL A, idirect, # AND immediate data to direct byte 3 2 ORL A, Rn OR register to Accumulator 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 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Girect OR direct byte to Accumulator 2 1 ORL A, Rn OR immediate data to Accumulator 2 1 ORL A, Rn Cacumulator 0 1 1 1 ORL A, A GRI OR immediate data to Accumulator 2 1 ORL DRI	INC	@Ri	Increment indirect RAM	1	
DEC Rn Decrement register 1 1 1 DEC direct Decrement direct byte 2 1 DEC @Ri Decrement indirect RAM 1 1 1 MUL AR Multiphy A & B 1 4 DIV AB Divide A by B 1 4 DA A Decimal Adjust Accumulator 1 1 1 LOGICAL OPERATIONS MNEMONIC DESTINATION BYTE CYC ANL A, Rn AND register to Accumulator 1 1 1 ANL A, direct AND direct byte to Accumulator 2 1 ANL A, Roli AND indirect RAM to Accumulator 1 1 1 ANL A, #data AND immediate data to Accumulator 2 1 ANL Direct, #data AND immediate data 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, #data OR immediate data to direct byte 3 2 ORL A, #data OR immediate data to direct byte 3 2 ORL A, #data OR immediate data to Accumulator 2 1 ORL A, #data OR immediate data to Direct byte 3 2 XRL A, #data OR immediate data to Accumulator 1 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL A, #data OR immediate data to Direct byte 3 2 XRL A, Rn Exclusive-OR register to Accumulator 1 1 XRL A, #data Exclusive-OR register to Accumulator 1 1 XRL A, #data Exclusive-OR indirect byte to Accumulator 1 1 XRL A, #data Exclusive-OR indirect byte to Accumulator 1 1 XRL A, #data Exclusive-OR indirect byte to Accumulator 1 1 XRL A, #data Exclusive-OR indirect than to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A, #data Exclusive-OR indirect RAM to A 1 1 XRL A Rotate Accumulator Left 1 1 XRL A Rotate Accum	INC	DPTR	Increment Data Pointer	1	2
DEC direct Decrement direct byte 2 1 DEC @Ri Decrement indirect RAM 1 1 1 MUL AR Multiply A & B 1 1 4 DIV AB Divide A by B 1 1 4 DA A Decimal Adjust Accumulator 1 1 1 LOGICAL OPERATIONS MINEMONIC DESTINATION BYTE CYC ANL A, GRi AND direct byte to Accumulator 1 1 1 ANL A, @Ri AND indirect RAM to Accumulator 1 1 1 ANL A, #data AND immediate data to Accumulator 2 1 ANL Direct, #data AND immediate data 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 1 ORL A, @Ri OR indirect byte to Accumulator 1 1 1 ORL A, Grect OR direct byte to Accumulator 2 2 1 ANL direct, #data AND immediate data to direct byte 2 1 ANL direct, #data OR immediate data to direct byte 2 2 1 ORL A, @Ri OR indirect byte to Accumulator 1 1 1 ORL A, @Ri OR indirect BAM to Accumulator 2 2 1 ORL A, @Ri OR indirect byte to Accumulator 2 2 1 ORL A, #data OR immediate data to direct byte 3 2 XRL A, Bn OR Accumulator 0 1 1 ORL Direct, #data OR immediate data to Accumulator 2 1 ORL Direct, #data OR immediate data to Accumulator 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, #data Exclusive-OR indirect byte 0 3 2 XRL A, Bn Exclusive-OR indirect byte 0 3 2 XRL A, #data Exclusive-OR indirect Dyte 0 Accumulator 1 1 XRL A, #data Exclusive-OR interect byte 0 Accumulator 1 1 XRL A, #data Exclusive-OR interect byte 0 Accumulator 1 1 XRL A, #data Exclusive-OR interect byte 0 Accumulator 1 1 XRL A, #data Exclusive-OR interect byte 0 Accumulator 1 1 XRL A, #data Exclusive-OR interect byte 0 Accumulator 1 1 XRL A, #data Exclusive-OR immediate data to A 1 1 XRL A, #data Exclusive-OR immediate data to A 1 1 XRL A, #data Exclusive-OR immediate data to A 1 1 XRL A, #data Exclusive-OR immediate data to A 1 1 XRL A, #data Exclusive-OR immediate data to A 1 1 XRL A, #data Exclusive-OR immediate data to A 1 1 XRL A, #data Exclusive-OR immediate data to A 1 1 XRL A Rotate Accumulator 1 1 1 XRL A Rotate Accumulator Left 1 1 XRL A Rotate Accum	DEC	Α	Decrement Accumulator	1	1
DEC direct Decrement idirect byte 2 1 DEC @Ri Decrement indirect RAM 1 1 1 MUL AR Multiply A & B 1 4 DIV AB Divide A by B 1 4 DA A Decimal Adjust Accumulator 1 1 LOGICAL OPERATIONS MNEMONIC DESTINATION BYTE CYC ANL A, Rn AND register to Accumulator 1 1 1 ANL A, direct AND direct byte to Accumulator 2 1 ANL A, @Ri AND indirect RAM to Accumulator 1 1 1 ANL A, direct, A AND immediate data to Accumulator 2 1 ANL Direct, #data AND immediate data to direct byte 2 1 ANL DRL A, Rn OR register to Accumulator 2 1 ANL Direct, #data AND immediate data to direct byte 3 2 CORL A, Rn OR register to Accumulator 1 1 ORL A, direct OR direct byte to Accumulator 1 1 ORL A, Walata OR immediate data to direct byte 3 2 CORL A, Walata OR immediate data to direct byte 3 2 CORL A, Walata OR immediate data to Accumulator 1 1 CORL A, Walata OR immediate data to Accumulator 1 1 CORL A, Walata OR immediate data to Accumulator 2 1 CORL A, Walata OR immediate data to direct byte 2 1 CORL DIRECT OR direct byte to Accumulator 2 1 CORL DIRECT OR direct byte to Accumulator 2 1 CORL DIRECT OR direct byte to Accumulator 2 2 CORL DIRECT OR direct byte to Accumulator 2 1 CORL DIRECT OR direct byte to Accumulator 2 2 CORL DIRECT OR direct byte to Accumulator 2 3 CORL DIRECT OR DIRECT OR REGISTER AND ACCUMULATOR 3 2 CORL DIRECT OR ACCUMULATOR 3 2 CORL DIRECT DIRECT OR REGISTER AND A 1 1 CORL DIRECT DIRECT DIRECT DIRECT DIRECT DATE OR ACCUMULATOR 3 2 CORL DIRECT DIRECT DIRECT DATE OR ACCUMULATOR 3 2 CORL DIRECT DIRECT DATE OR ACCUMULATOR 3 2 CORL DIRECT DATE OR ACCUMULATOR 3 2 CORL DIRECT DATE OR ACCUMULATOR 3 3 CORL DATE OR ACCUMULAT	DEC	Rn	Decrement register	1	1
DEC @Ri Decrement indirect RAM 1 1 1 1 MULL AR Multiply A & B 1 1 4 DIV AB Divide A by B 1 1 4 DIV AB Divide A by B 1 1 4 DIV AB Divide A by B 1 1 1 4 DIV AB DECIMAL ACCUMULATOR 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	DEC	direct		2	1
DIV AB Divide A by B DA A Decimal Adjust Accumulator LOGICAL OPERATIONS MNEMONIC DESTINATION MIL A, Rin AND register to Accumulator ANL A, Griect AND direct byte to Accumulator ANL A, Griect AND indirect RAM to Accumulator ANL A, #data AND immediate data to Accumulator ANL Direct, A AND immediate data to Accumulator ANL Direct, A AND immediate data to direct byte ANL Direct, A AND immediate data to direct byte ANL Direct, A AND immediate data to direct byte ANL Direct, A AND Accumulator ANL Direct, A ACCUMULATOR AND Accumulator	DEC	@Ri		1	
DIV AB Divide A by B DA A Decimal Adjust Accumulator LOGICAL OPERATIONS MNEMONIC DESTINATION MIL A, Rin AND register to Accumulator ANL A, direct AND direct byte to Accumulator ANL A, @ Ri AND indirect RAM to Accumulator ANL A, #data AND immediate data to Accumulator ANL direct, A AND accumulator 1 ANL direct, A AND Accumulator 2 ANL direct, A AND immediate data to direct byte 2 ORL A, Rin OR register to Accumulator 1 ORL A, direct OR direct byte to Accumulator 1 ORL A, #data OR immediate data to direct byte 3 ORL A, #data OR immediate data to irect byte 1 ORL A, #data OR immediate data to Accumulator 2 ORL A, #data OR immediate data to Accumulator 1 ORL A, #data OR immediate data to Accumulator 2 ORL DIRECT A OR Accumulator 1 ORL DIRECT A OR Accumulator 1 ORL DIRECT A OR Accumulator 2 INDICATE A OR Accumulator 1 ORL DIRECT A OR Accumulator 2 INDICATE A OR ACCUMULATOR 1 INDICATE A OR IMMEDIATE AND IMMED	MUL	AR	Multiply A & B	1	4
DA A Decirnal Adjust Accumulator 1 1 1 LOGICAL OPERATIONS MNEMONIC DESTINATION BYTE CYC ANL A, Rn AND register to Accumulator 1 1 1 ANL A, direct AND direct byte to Accumulator 2 1 ANL A, ®Ri AND indirect RAM to Accumulator 1 1 1 ANL A, #data AND immediate data to Accumulator 2 1 ANL direct, A AND Accumulator 1 1 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 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 2 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL DRL A, #data OR immediate data to Accumulator 2 1 ORL DRL A, Rn OR register to Accumulator 2 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL DRL DRL DRL DRL DRL DRL DRL DRL DRL D	DIV	AB		1	
MNEMONIC ANL A, Rn AND register to Accumulator ANL A, direct AND direct byte to Accumulator ANL A, #data AND immediate data to Accumulator ANL A, #data AND immediate data to direct byte ANL A, direct, A AND Accumulator to direct byte ANL A, #data AND immediate data to direct byte ANL ANL A, #data AND immediate data to direct byte ANL ANL AIH AND AND ACCUMULATOR ANL AIH AIH AIH AIH AIH AIH AIH AI	DA	A		1	
MNEMONIC ANL A, Rn AND register to Accumulator ANL A, direct AND direct byte to Accumulator ANL A, & Ri AND immediate data to Accumulator ANL A, Wata AND immediate data to direct byte ANL A, Rn AND immediate data to direct byte ANL A, Rn OR register to Accumulator ANL ANL A, #data AND immediate data to direct byte ANL ANL A, Rn OR register to Accumulator A, Rn OR Register to Accumulator A, Rn OR DRI CRL A, A, Wata OR immediate data to direct byte A, Wata ORL A, Wata OR immediate data to Accumulator ANL A, Wata OR immediate data to Accumulator ANL A, Wata OR immediate data to direct byte ANL A, Rn ARN Exclusive-OR register to Accumulator ANL A, Wata	LOGICAL OPE	PATIONS			
ANL A, Rn AND register to Accumulator 1 1 1 ANL A, direct AND direct byte to Accumulator 2 1 ANL A, @Ri AND indirect RAM to Accumulator 1 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 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 2 1 ORL A, #data OR immediate data to direct byte 2 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL DRL DRL DRL DRL DRL DRL DRL DRL DRL D		MATIONS	DESTINATION	BYTE	CYC
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 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 2 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL DIRECT A OR Accumulator 2 1 ORL DIRECT A OR Accumulator 3 2 ORL DIRECT A OR Accumulator 4 3 2 AND immediate data to Accumulator 5 1 AND ACCUMULATOR ACCUMULATOR 5 1 AND ACCUMULATOR 5 1 ACCUMULATOR 5 1 AND ACCUMULAT		A Do			
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 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 2 1 ORL A, #data OR immediate data to Accumulator 1 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL DRL DRL DRL DRL DRL DRL DRL DRL DRL D	1	•			
ANL A, #data AND immediate data to Accumulator 2 1 ANL direct, A AND Accumultaor 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 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, ®Ri OR indirect RAM to Accumulator 1 1 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL DIRECT, #data OR immediate data to Accumulator 2 1 ORL DIRECT, #data OR immediate data 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 1 1 XRL DIRECT, #data Exclusive-OR immediate data to A 2 1 XRL DIRECT, #data Exclusive-OR immediate data to A 2 1 XRL DIRECT, #data Exclusive-OR immediate data to A 2 1 XRL DIRECT, #data Exclusive-OR immediate data to A 2 1 XRL DIRECT, #data Exclusive-OR immediate data to DIRECT, #data Exclusive-OR immediate DIRECT, #data Exc	1				
ANL direct, A AND Accumultaor 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 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 1 1 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL Direct A OR Accumulator 1 1 1 ORL Direct A OR Accumulator 1 1 1 ORL Direct A OR Accumulator 2 1 ORL Direct, #data OR immediate data to direct byte 2 1 ORL Direct, #data OR immediate data to direct byte 3 2 XRL Direct Exclusive-OR register to Accumulator 1 1 XRL Direct Exclusive-OR direct byte to Accumulator 2 1 XRL Direct Exclusive-OR indirect RAM to Direct Dire	1	,		•	-
ANL direct, #data AND immediate data to direct byte 3 2 ORL A, Rn OR register to Accumulator 1 1 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 1 1 1 ORL A, #data OR immediate data to Accumulator 2 1 ORL Direct A OR Accumulator 2 1 ORL Direct A OR Accumulator 1 1 1 ORL Direct A OR Accumulator 1 2 1 ORL Direct A OR Accumulator 1 2 1 ORL Direct, #data OR immediate data to direct byte 2 1 ORL Direct, #data OR immediate data to direct byte 3 2 XRL Direct, #data OR immediate data to Direct byte 3 2 XRL Direct Exclusive-OR register to Accumulator 1 1 XRL Direct Exclusive-OR indirect byte to Accumulator 2 1 XRL Direct Exclusive-OR indirect RAM to A 1 1 1 XRL Direct, A Exclusive-OR immediate data to A 2 1 XRL Direct, A Exclusive-OR Accumulator to direct byte 2 1 XRL Direct, A Exclusive-OR immediate data to Direct Dyte 2 1 XRL Direct, #data Exclusive-OR immediate data to Direct Dyte 2 1 XRL Direct, #data Exclusive-OR immediate data to Direct Dyte 2 1 XRL Direct, #data Exclusive-OR immediate data to Direct Dyte 2 1 XRL Direct, #data Exclusive-OR immediate data to Direct Dyte 2 1 XRL Direct, #data Exclusive-OR immediate Dyte 2 1 XRL Direct, #data Exclusive-OR immediate Dyte 3 2 CLR Direct A Clear Accumulator 1 1 XRL Direct					
ORL A, Rn OR register to Accumulator 1 1 ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 1 1 ORL 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 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 direct RAM to A 1 1 XRL A, #data Exclusive-OR immediate data to A 2 1 XRL A, #data Exclusive-OR Accumulator 2 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 A 2 1 XRL Direct, #data Exclusive-OR immediate data to direct 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		•			
ORL A, direct OR direct byte to Accumulator 2 1 ORL A, @Ri OR indirect RAM to Accumulator 1 1 ORL 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 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 A, #data Exclusive-OR Accumulator 2 1 XRL Direct, A Exclusive-OR Accumulator 1 1 XRL Direct, A Exclusive-OR immediate data to A 2 1 XRL Direct, A Exclusive-OR immediate data to A 2 1 XRL Direct, A Exclusive-OR immediate data to Direct byte 2 1 XRL Direct, #data Exclusive-OR immediate data to direct 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					
ORL A, @Ri OR indirect RAM to Accumulator 1 1 1 ORL 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 1 XRL A, direct Exclusive-OR direct byte to Accumulator 2 1 XRL A, @Ri Exclusive-OR indirect RAM to A 1 1 1 XRL A, #data Exclusive-OR immediate data to A 2 1 XRL A, #data Exclusive-OR Accumulator 2 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 3 2 CLR A Clear Accumulator 1 1 1 CPL A Complement Accumulator 1 1 1 RL A Rotate Accumulator Left 1 1 RLC A Rotate Accumulator Left 1 1 RLC A Rotate Accumulator Right 1 1					
ORL 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 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 3 2 CLR A Clear Accumulator 3 2 CLR A Clear Accumulator 1 1 CPL A Complement Accumulator 1 1 RL A Rotate Accumulator Left 1 1 RLC A Rotate Accumulator Right 1 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 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 3 2 CLR A Clear Accumulator 3 2 CLR A Clear Accumulator 1 1 CPL A Complement Accumulator 1 1 RL A Rotate Accumulator Right 1 1	T				
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 AAM 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, A Exclusive-OR immediate data to direct 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					
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 3 2 CLR A Clear Accumulator 1 1 CPL A Complement Accumulator 1 1 RL A Rotate Accumulator 1 1 RL A Rotate Accumulator 1 1 RLC A Rotate A Left through the Carry flag 1 1 RR A Rotate Accumulator Right 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 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					
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 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	—			-	
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 3 2 CLR A Clear Accumulator 1 1 CPL A Complement Accumulator 1 1 RL A Rotate Accumulator 1 1 RLC A Rotate A Left through the Carry flag 1 RR A Rotate Accumulator Right 1					
XRL direct, A Exclusive-OR Accumulator to direct byte 2 1 XRL direct, #data Exclusive-OR immediate data to direct 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				•	
XRL direct, #data Exclusive-OR immediate data to direct 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		,			
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		•			
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				_	
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				•	
RLC A Rotate A Left through the Carry flag 1 1 1 RR A Rotate Accumulator Right 1 1					
RR A Rotate Accumulator Right 1 1	ı Di	Α		•	
	_ · · · · ·				4
	RLC		Rotate A Left through the Carry flag	•	
The state of the s	RLC RR	Ä	Rotate Accumulator Right	1	1
SWAP A Swap nibbles within the Accumulator 1 1	RLC RR RRC	A A	Rotate Accumulator Right Rotate A Flight through Carry flag	1 1	1

80C31/Rev.1.0







MNEMONIC DESCRIPTION BYTE CYC					
MOV		RANSFER		D)/75	01/0
MOV A, direct Move direct byte to Accumulator 2 1 MOV A, deat Move indirect RAM to Accumulator 2 1 MOV A, deata Move indirect RAM to Accumulator 2 1 MOV Rh, deata Move Accumulator to dresister 1 1 MOV Rh, deata Move direct byte to register 2 2 MOV Grect, A Move direct byte to register 2 2 MOV direct, A Move Accumulator to direct byte 2 2 MOV direct, direct Move direct byte to direct byte 2 2 MOV direct, deata Move direct byte to direct byte 2 2 MOV GRI, deata Move direct byte to indirect RAM 1 1 MOV GRI, direct Move direct byte to indirect RAM 1 1 MOV GRI, direct Move direct byte to indirect RAM 1 1 MOV GRI, direct Move direct byte to indirect RAM 1 1 MOV					
MOV A, @Ri Move indirect PAM to Accumulator 1 1 1 1 MOV A, #data Move immediate data to Accumulator 2 1 MOV A, #data Move immediate data to register 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 2 2 2 <td></td> <td>•</td> <td></td> <td></td> <td></td>		•			
MOV A, #data Move immediate data to Accumulator 2 1 MOV Rn, A Move Accumulator to register 1 1 MCV Rn, direct Move direct byte to register 2 2 MCV direct, A Move immediate data to register 2 1 MCV direct, A Move incent diversity to direct byte 2 1 MCV direct, direct Move direct byte to direct byte 2 2 MCV direct, direct Move direct byte to direct byte 2 2 MCV direct, direct Move direct byte to direct byte 2 2 MCV direct, diata Move direct byte to direct byte 3 2 MCV direct, diata Move direct byte to direct byte 2 2 MCV direct, data Move direct byte to direct byte 3 2 MCV direct, data Move at a can direct byte to indirect RAM 1 1 MCV direct, data Move data byte to indirect DAM 2 2		•			-
MOV Rn, A Move Accumulator to register 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 2 1 1 2 2 1 1 2 1 1 2 2		•			-
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, Girect Move direct byte 2 2 MOV direct, Giret Move direct byte to direct byte 2 2 MOV direct, Girla Move immediate data to direct byte 3 2 MOV direct, Hata Move immediate data to direct byte 3 2 MOV @RI, direct Move direct byte to indirect RAM 1 1 MOV @RI, direct Move direct byte to indirect RAM 2 2 MOV @RI, direct Move direct byte to indirect RAM 2 1 MOV QRI, direct Move direct byte to indirect RAM 2 1 MOV QRI, direct Move direct byte to indirect RAM 2 1 MOVC A, @A + DCT Move external RAM (16-bit addr) to A 1 2		•		_	-
MOV Rn, #data Move namediate 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 2 2 MOV direct, direct Move direct byte to direct byte 3 2 MOV direct, #data Move immediate data to direct byte 3 2 MOV @Ri, drect Move documulator to indirect RAM 1 1 1 MOV @Ri, #data Move immediate data to indirect RAM 2 2 MOV DPTR, #data to Move immediate data to indirect RAM 2 1 MOV DPTR, #data to Move immediate data to indirect RAM 2 2 MOV DPTR, #data to Move immediate data to indirect RAM 2 1 MOV A @Ri, #data Move Experience Fall to to Indirect RAM 2 1 MOV A @Ri Move A DETR Move Experience Fall to Indirect RAM 1 2					
MOV direct, A Move Accumulator to direct byte 2 1 MOV direct, Rn Move register to direct byte 2 2 MOV direct, direct Move indirect byte to direct 3 2 MOV direct, direct Move inmediate data to direct byte 2 2 MOV GRI, direct Move inmediate data to indirect RAM 1 1 MOV GRI, direct Move direct byte to indirect RAM 2 2 MOV GRI, direct Move direct byte to indirect RAM 2 2 MOV GRI, direct Move direct byte to indirect RAM 2 1 MOV DPTR, #data 16 Load Data Pointer with a 16-bit constant 3 2 MOV A, @A+ PC Move Code byte relative to PC to A 1 2 MOVX A, @RI Move External RAM (16-bit addr) to A 1 2 MOVX A, @API Move A to External RAM (16-bit addr) 1 2 MOVX QDPTR, A Move A to External RAM (16-bit addr) 1 2					
MOV direct, Rn Move register to direct byte 2 2 MOV direct, direct Move direct byte to direct 3 2 MOV direct, @Ri Move indirect RAM to direct byte 2 2 MOV direct, #data Move inmediate data to direct byte 3 2 MOV @Ri, 4 Move Accumulator to indirect RAM 1 1 MOV @Ri, 4data Move immediate data to indirect RAM 2 2 MOV DPTR, #data 16 Move immediate data to indirect RAM 2 1 MOVC A, @A + DPTR Move Code byte relative to DPTR to A 1 2 MOVX A, @A + PC Move Code byte relative to DPTR to A 1 2 MOVX A, @RI Move External RAM (8-bit addr) to A 1 2 MOVX QRI, A Move External RAM (8-bit addr) to A 1 2 MOVX @RI, A Move A to External RAM (8-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (8-bit addr) 1 1					
MOV direct, direct Move direct byte to direct 3 2 MOV direct, #data Move immediate data to direct byte 3 2 MOV direct, #data Move immediate data to direct byte 3 2 MOV ØRI, A Move Accumulator to indirect RAM 1 1 MOV ØRI, #data Move Code byte to indirect RAM 2 2 MOV ØRI, #data Move immediate data to indirect RAM 2 1 MOV OPTR, #data 16 Load Data Pointer with a 16-bit constant 3 2 MOVC A, @A + DPTR Move Code byte relative to DPTR to A 1 2 MOVX A, @RI Move External RAM (16-bit addr) to A 1 2 MOVX A, @DPTR Move A to External RAM (16-bit addr) to A 1 2 MOVX @DPTR, A Move External RAM (16-bit addr) to A 1 2 VBUSH direct Push direct byte onto stack 2 2 VCH A, Rn Exchange to stack byte and stack 2 2		•			
MOV direct, @Ri Move indirect (ARM 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 1 MOV @Ri, data Move direct byte to indirect RAM 2 1 MOV DPTR, data 16 Load Data Pointer with a 16-bit constant 3 2 MOVC A, @A + PPT Move Code byte relative to PCT to A 1 2 MOVX A, @FI Move External RAM (8-bit addr) to A 1 2 MOVX QRI, A Move External RAM (8-bit addr) to A 1 2 MOVX @RI, A Move A to External RAM (16-bit addr) 1 2 MOVX @RI, A Move A to External RAM (16-bit addr) 1 2 WOVX @RI, A Move A to External RAM (16-bit addr) 1 2 WOVX @RI, A Move A to External RAM (16-bit addr) 1 2		•			2
MOV direct, #data Move immediate data to direct byte 3 2 MOV ®Ri, A Move Accumulator to indirect RAM 1 1 1 MOV @Ri, idrect Move direct byte to indirect RAM 2 2 MOV @RI, #data Move immediate data to indirect RAM 2 1 MOV DPTR, #data 16 Load Data Pointer with a 16-bit constant 3 2 MOVC A, @A + DPTR Move Code byte relative to DPTR to A 1 2 MOVX A, @RI Move External RAM (8-bit addr) to A 1 2 MOVX A, @RI Move External RAM (16-bit addr) to A 1 2 MOVX BPTR, A Move A to External RAM (16-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 PUSH direct byte orto stack 2 2 2 VCH A, Rn Exchange tregister with Accumulator 1 1 1 XCH A, GRI Exchange low-order nibble ind RAM with A		•			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 control of the property of t					2
MOV ® Ri, direct Move direct byte to indirect RAM 2 2 MOV ® Ri, ifedata Move immediate data to indirect RAM 2 1 MOV Q Ri, ifedata Move Code byte relative to DPTR to A 1 2 MOVC A, @ A + DPTR Move Code byte relative to DPTR to A 1 2 MOVX A, @ Ri Move Code byte relative to DPTR to A 1 2 MOVX A, @ DPTR Move Code byte relative to DPTR to A 1 2 MOVX A, @ DPTR Move External RAM (8-bit addr) to A 1 2 MOVX Ø PRI, A Move A to External RAM (8-bit addr) 1 2 MOVX @ DPTR, A Move A to External RAM (8-bit addr) 1 2 PUSH direct Pop direct byte from stack 2 2 2 XCH A, Rn Exchange register with Accumulator 1 1 1 XCH A, Q Ri Exchange direct byte with Accumulator 2 1 XCH A, Q Ri Exchange direct byte with Accumulator <td></td> <td>,</td> <td>· · · · · · · · · · · · · · · · · · ·</td> <td></td> <td></td>		,	· · · · · · · · · · · · · · · · · · ·		
MOV					
MOV DPTR, #data 16 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 MOVX A, @Ri Move External RAM (8-bit addr) to A 1 2 MOVX QRI, A Move A to External RAM (8-bit addr) to A 1 2 MOVX @DPTR, A Move A to External RAM (6-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (6-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (6-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (6-bit addr) 1 2 PUSH direct Push direct byte from stack 2 2 2 VOP direct Push direct byte order of the form stack 2 2 2 XCH A, Rn Exchange register with Accumulator 1 1 1 XCH A, @Ri Exchange register wit					
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 MOVX A, @Fi Move External RAM (8-bit addr) to A 1 2 MOVX A, @DPTR Move external RAM (16-bit addr) to A 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 PUSH direct Push direct byte onto stack 2 2 PUSH direct Push direct byte onto stack 2 2 POP direct Push direct byte onto stack 2 2 XCH A, Rn Exchange legister with Accumulator 1 1 XCH A, direct Exchange legister with Accumulator 2 1 XCH A, QRi Exchange legister With Accumulator 2 1 XCH A, QRi Exchange legister ARM with A 1 1 XCH	1 = .				
MOVC A, @A + PC Move Code byte relative to PC to A 1 2 MOVX A, @Ri Move External RAM (8-bit addr) to A 1 2 MOVX A, @DPTR Move A to External RAM (6-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 PUSH direct Push direct byte onto stack 2 2 POP direct Pop direct byte from stack 2 2 YCH A, Rn Exchange register with Accumulator 1 1 XCH A, Gri Exchange indirect Pyte with Accumulator 2 1 XCH A, @Ri Exchange direct byte with Accumulator 2 1 XCH A, @Ri Exchange direct byte with Accumulator 2 1 XCH A, @Ri Exchange direct byte with Accumulator 2 1 XCH A, @Ri Exchange direct byte with Accumulator 2 2 <t< td=""><td></td><td></td><td></td><td>_</td><td></td></t<>				_	
MOVX A, @Ri Move External RAM (8-bit addr) to A 1 2 MOVX A, @DPTR Move external RAM (16-bit addr) to A 1 2 MOVX @Ri, A Move A to External RAM (16-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 PUSH direct Push direct byte onto stack 2 2 POP direct Push direct byte from stack 2 2 XCH A, Rn Exchange lirect byte with Accumulator 1 1 1 XCH A, direct Exchange indirect RAM with A 1 1 1 XCH A, direct Exchange indirect RAM with A 1 1 1 XCH A, direct Exchange indirect RAM with A 1 1 1 XCH A, direct Exchange indirect RAM with A 1 1 1 XCH A, direct Exchange indirect RAM with A				•	2
MOVX A. @DPTR Move external RAM (16-bit addr) 1 2 MOVX @Ri, A Move A to External RAM (8-bit addr) 1 2 MOVX @DPTR, A Move A to External RAM (16-bit addr) 1 2 PUSH direct Push direct byte onto stack 2 2 2 POP direct Push direct byte from stack 2 2 2 XCH A, Rn Exchange register with Accumulator 1 1 1 XCH A, direct Exchange indirect byte with Accumulator 2 1 1 XCH A, @Ri Exchange indirect byte with Accumulator 2 1 1 XCHD A, @Ri Exchange lindirect byte with Accumulator 2 1 1 XCHD A, @Ri Exchange lindirect byte with Accumulator 2 1 1 XCHD A, @Ri Exchange lindirect byte with Accumulator 2 1 1 XCHD A, @Ri Exchange lindirect byte with Accumulator 2 1 1		,		-	
MOVX ® Ri, A Move A to External RAM (8-bit addr) 1 2 MOVX ® DPTR, A Move A to External RAM (16-bit addr) 1 2 PUSH direct Push direct byte onto stack 2 2 POP direct Pop direct byte from stack 2 2 XCH A, Rn Exchange ergister with Accumulator 2 1 XCH A, @ Ri Exchange ergister with Accumulator 2 1 XCH A, @ Ri Exchange low-order nibble ind RAM with A 1 1 XCHD A, @ Ri Exchange low-order nibble ind RAM with A 1 1 XCHD A, @ Ri Exchange low-order nibble ind RAM with A 1 1 XCHD A, @ Ri Exchange low-order nibble ind RAM with A 1 1 XCHD A, @ Ri Exchange low-order nibble ind RAM with A 1 1 XCHD A, @ Ri Exchange low-order nibble ind RAM with A 1 1 XCHD A & Brit Exchange low-order nibble ind RAM with A 1 1	MOVX	A, @Ri		•	2
MOVX ® DPTR, A direct Move A to External RAM (16-bit addr) 1 2 PUSH direct Push direct byte onto stack 2 2 POP direct Pop direct byte from stack 2 2 XCH A, Rn Exchange register with Accumulator 1 1 XCH A, @Ri Exchange direct byte with Accumulator 2 1 XCH A, @Ri Exchange direct byte with Accumulator 2 1 XCH A, @Ri Exchange direct byte with Accumulator 2 1 XCHD A, @Ri Exchange low-order nibble ind RAM with A 1 1 BOOLEAN VARIABLE MANIPULATION BYTE CYC CLR C Clear Carry flag 1 1 CLR C Clear direct bit 2 1 SETB bit Clear direct bit 2 1 SETB bit Set direct bit 2 1 CPL C Complement Carry flag 1 1 CPL Dit C bit AND direct bit to Carry flag 2	MOVX	A, @DPTR		-	2
PUSH direct Push direct byte onto stack 2 2 2 2 2 2 2 2 2				•	2
POP direct Pop direct byte from stack 2 2 XCH A, Rn Exchange register with Accumulator 1 1 XCH A, direct Exchange incert byte with Accumulator 2 1 XCH A, @Ri Exchange indirect RAM with A 1 1 XCHD A, @Ri Exchange low-order nibble ind RAM with A 1 1 BOOLEAN VARIABLE MANIPULATION BYTE CYC CLR C Clear Carry flag 1 1 CLR C Clear darry flag 1 1 CLR bit Clear direct bit 2 1 SETB C Set Carry flag 1 1 1 SETB bit Clear direct bit 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2	MOVX	@DPTR, A	Move A to External RAM (16-bit addr)		
XCH A, Rn Exchange register with Accumulator XCH A, direct Exchange direct byte with Accumulator XCH A, direct Exchange indirect RAM with A 1 XCHD A, @Ri Exchange indirect RAM with A 1 XCHD A, @Ri Exchange low-order nibble ind RAM with A 1 XCHD A, @Ri Exchange low-order nibble ind RAM with A 1 BOOLEAN VARIABLE MANIPULATION MNEMONIC DESCRIPTION METE CYC CLR C Clear Carry flag 1 CLR bit Clear direct bit 2 1 SETB C Set Carry flag 1 1 SETB C C Set Carry flag 1 1 SETB bit Set direct Bit 2 1 CPL C Complement Carry flag 1 1 CPL C Complement Carry flag 1 1 ANL C, bit AND direct bit to Carry flag 2 2 ANL C, bit AND complement of direct bit to Carry 2 2 ANL C, bit OR direct bit to Carry flag 2 2 CRL C, bit OR direct bit to Carry flag 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit OR complement of direct bit to Carry 2 2 CRL C, bit Move direct bit to Carry flag 2 2 1 MOV bit, C Move direct bit to Carry flag 2 2 2 2 CRL C, bit Move direct bit to Carry flag 2 2 1 MOV bit, C Move direct bit to Carry flag 2 2 2 2 2 2 3 4 4 CYC ACALL addr 11 Absolute subroutine Call 2 2 2 4 CALL addr 11 Absolute subroutine Call 3 2 2 2 3 4 4 AJMP addr 16 Long Subroutine Call 3 2 2 3 4 AJMP addr 16 Long Jump 3 3 2 3 4 AJMP addr 16 Long Jump 3 3 2 3 4 AJMP addr 16 Long Jump 3 4 4 AJMP addr 16 Long Jump 3 4 4 AJMP addr 16 Long Jump 6 4 ACALL rel Jump indirect relative to the DPTR 1 4 AJMP addr 16 Long Jump indirect relative to the DPTR 1 4 AJMP addr 16 Long Jump indirect relative to the DPTR 1 4 AJMP addr 16 Jump indirect relative to the DPTR 1 4 AJMP addr 16 Jump indirect relative to the DPTR 1 4 AJMP addr 16 Jump indirect relative to the DPTR 1 4 AJMP addr 16 Jump i	PUSH	direct	Push direct byte onto stack		
XCH	POP	direct	Pop direct byte from stack		
XCH A, @Ri Exchange indirect RAM with A 1 1 XCHD A, @Ri Exchange low-order nibble ind RAM with A 1 1 BOOLEAN VARIABLE MANIPULATION MNEMONIC DESCRIPTION BYTE CYC CLR C Clear Carry flag 1 1 CLR bit Clear direct bit 2 1 SETB C Set Carry flag 1 1 SETB C Set direct Bit 2 1 CPL C Complement Carry flag 1 1 CPL D. C. Complement direct bit 2 1 ANL C, bit AND direct bit to Carry flag 2 2 ANL C, bit AND direct bit to Carry flag 2 2 ORL C, bit OR direct bit to Carry flag 2 2 ORL C, bit Move direct bit to Carry flag 2 2 MOV C, bit Move direct bit to Carry flag 2 2 <	XCH	A, Rn	Exchange register with Accumulator	•	
Name	XCH	A, direct	Exchange direct byte with Accumulator		
BOOLEAN VARIABLE MANIPULATION MNEMONIC DESCRIPTION BYTE CYC CLR C Clear darry 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 Carry flag 2 2 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 ORL C, bit OR complement of direct bit to Carry 2 2 ORL C, bit OR complement of direct bit to Carry 2 2 MOV bit, C Move direct bit to Carry flag 2 <	XCH	A, @Ri	Exchange indirect RAM with A	1	1
MNEMONIC DESCRIPTION BYTE CYC 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 2 1 MOV bit, C Move Carry flag to direct bit 2 2 2 PROGRAM AND MACHINE CONTROL MNEMONIC BYTE CYC ACALL addr 1	XCHD	A, @Ri	Exchange low-order nibble ind RAM with A	1	1
MNEMONIC DESCRIPTION BYTE CYC 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 2 1 MOV bit, C Move Carry flag to direct bit 2 2 2 PROGRAM AND MACHINE CONTROL MNEMONIC BYTE CYC ACALL addr 1	BOOLEAN VAR	IABLE MANIPULAT	ION		
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 carry flag 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 OR complement of direct bit to Carry 2 2 MOV C,bit Move direct bit to Carry flag 2 2 1 MOV bit, C Move Carry flag to direct bit 2 2 2 PROGRAM AND MACHINE CONTROL MNEMONIC BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 <th></th> <th></th> <th></th> <th>BYTE</th> <th>CYC</th>				BYTE	CYC
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 MNEMONIC BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 2 2 RET Return from subroutine 1 2 RETI Return from interr	CLR	С	Clear Carry flag		-
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 OR direct bit to Carry flag 2 2 ANL C,bit OR direct bit to Carry flag 2 2 ANL C,bit OR direct bit to Carry flag 2 2 ORL C,bit OR complement of direct bit to Carry 1 2 2 MOV C,bit Move direct bit to Carry flag 2 1 MOV bit, C Move Carry flag 0 2 1 MOV bit, C Move Carry flag to direct bit 2 2 PROGRAM AND MACHINE CONTROL MNEMONIC DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 3 2 LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 Long Jump 3 2 SJMP rel Short Jump (relative addr) 2 JMP @A + DPTR Jump interct leative to the DPTR 1 JZ rel Jump if Accumulator is Zero 2 JNZ rel Jump if Accumulator is Not Zero 2 JC rel Jump if Carry flag is set 2	CLR	bit	Clear direct bit		
CPL C Complement Carry flag 1 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 0 2 1 MOV bit, C Move Carry flag 0 2 1 MOV bit, C Move Carry flag 0 2 1 MOV DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 Long Jump 3 3 SJMP rel Short Jump (relative addr) 2 JMP @A + DPTR Jump indirect relative to the DPTR 1 JZ rel Jump if Accumulator is Zero 2 JNZ rel Jump if Carry flag is set 2	SETB	С	Set 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 direct bit to Carry flag 2 1 MOV bit, C Move Carry flag to direct bit 2 2 PROGRAM AND MACHINE CONTROL Move Carry flag to direct bit 2 2 PROGRAM AND MACHINE CONTROL BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 ACALL addr 11 Absolute subroutine Call 2 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 <	SETB	bit	Set direct Bit		1
ANL C, bit AND direct bit to Carry flag ANL C,/bit AND complement of direct bit to Carry ORL C,bit OR direct bit to Carry flag ORL C,/bit OR complement of direct bit to Carry ORL C,/bit OR complement of direct bit to Carry ORL C,/bit OR complement of direct bit to Carry ORL C,/bit Move direct bit to Carry flag ORU C, bit Move direct bit to Carry flag MOV bit, C Move Carry flag to direct bit ORUPTOR CONTROL MNEMONIC DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call LCALL addr 16 Long Subroutine Call OR CALL ABSOLUTE SUBROUTINE OR CALL A	CPL	С	Complement Carry flag	1	1
ANL C,bit AND complement of direct bit to Carry ORL C,bit OR direct bit to Carry flag ORL C,bit OR complement of direct bit to Carry MOV C,bit Move direct bit to Carry flag MOV bit, C Move Carry flag to direct bit MNEMONIC ACALL addr 11 Absolute subroutine Call LCALL addr 16 Long Subroutine Call LCALL addr 11 Return from subroutine RET Return from subroutine RETI Return from interrupt AJMP addr 11 Absolute Jump LJMP addr 16 Long Jump SJMP rel Short Jump (relative addr) JMP @A + DPTR Jump indirect relative to the DPTR JZ rel Jump if Accumulator is Not Zero JC rel Jump if Carry flag is set	CPL	bit	Complement direct bit		
ANL C,/bit AND complement of direct bit to Carry ORL C,bit OR direct bit to Carry flag ORL C,/bit OR complement of direct bit to Carry MOV C,bit OR complement of direct bit to Carry MOV bit, C Move direct bit to Carry flag MOV bit, C Move Carry flag to direct bit PROGRAM AND MACHINE CONTROL MNEMONIC ACALL addr 11 Absolute subroutine Call LCALL addr 16 Long Subroutine Call RET Return from subroutine RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump LJMP addr 16 Long Jump SJMP rel Short Jump (relative addr) JMP @A + DPTR Jump indirect relative to the DPTR JZ rel Jump if Accumulator is Not Zero JNZ rel Jump if Carry flag is set A carry flag 2 2 2 2 2 2 3 2 4 3 3 5 3 6 4 7 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7 8	ANL	C, bit	AND 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 BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 ACALL addr 11 Absolute subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 Long Jump 3 2 SJMP rel Short Jump (relative addr) 2 2 JMP @A + DPTR Jump inferct 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 s	ANL		AND complement of direct bit to Carry		
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 MNEMONIC DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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 Carry flag is set 2 2	ORL	C,bit	OR direct bit to Carry flag		
MOV bit, C Move Carry flag to direct bit 2 2 PROGRAM AND MACHINE CONTROL MNEMONIC DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	ORL	C./bit	OR complement of direct bit to Carry		
PROGRAM AND MACHINE CONTROL MNEMONIC DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	MOV	C,bit	Move direct bit to Carry flag		1
MNEMONIC DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	MOV	bit, C	Move Carry flag to direct bit	2	2
MNEMONIC DESCRIPTION BYTE CYC ACALL addr 11 Absolute subroutine Call 2 2 LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	PROGRAM AND	MACHINE CONTR	OL .		
LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	MNEMONIC		DESCRIPTION	BYTE	CYC
LCALL addr 16 Long Subroutine Call 3 2 RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	ACALL	addr 11	Absolute subroutine Call		
RET Return from subroutine 1 2 RETI Return from interrupt 1 2 AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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		addr 16	Long Subroutine Call		2
AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	RET				2
AJMP addr 11 Absolute Jump 2 2 LJMP addr 16 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	RETI		Return from interrupt		2
LJMP addr 16 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	AJMP	addr 11	Absolute Jump	2	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	LJMP	addr 16	•	3	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
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					2
JNZ rel Jump if Accumulator is Not Zero 2 2 JC rel Jump if Carry flag is set 2 2					2
JC rel Jump if Carry flag is set 2 2					
			· · · · · · · · · · · · · · · · · · ·		



80C31/Rev.1.0



PROGRAM MNEMONIC	AND MACHINE CONTR	ROL (cont.) DESCRIPTION	ВУТЕ	CYC
JB	bit. rel	Jump if direct Bit set	3	2
JNB	bit, rel	Jump if direct Bit Not sef	3	2
JBC	bit, rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A, direct, rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A, #data, ref	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn, #data, rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri, #data, rel	Comp. immed. to ind. & jump if Not Equal	3	2
DJNZ	Rn, rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct, rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn - Working register R0-R7

direct - 128 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

#data 16 - 16-bit constant included as bytes 2 & 3 of instruction bit - 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr 16

- Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space

5868456 0003728 494 **III**

Addr 11

rel

- Destination address for ACALL & AJMP will be within the same 2-k page of program

memory as the first byte of the following instruction - SJMP and all conditional jumps include an 8-bit offset byte. Range is + 127 - 128 bytes

relative to the first byte of the following instruction. All mnemonics copyrighted® Intel Corporation 1979

INSTRUCTION OPCODES IN HEXADECIMAL ORDER

HEX	NUMB.	MNEM.	OPERANDS
CODE	OF BYTES		
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06 07	1 1	INC INC	@R0 @R1
08	1	INC	R0
09	i	INC	R1
0A	i	INC	R2
0B	i	INC	R3
OC.	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12 13	3 1	LCALLR RC	code addr A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	i	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D 1E	1 1	DEC DEC	R5 R6
1F	1	DEC	10 187
20	3	JB	bit addr. code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	Α
24	2	ADD	A, data
25	2	ADD	A, data addr
26	1	ADD	A, @ R0
27	1	ADD	A, @R1
28	1	ADD	A, R0
29 2A	1 1	ADD ADD	A, R1 A, R2
2B	1	ADD	A, R3
2C	1	ADD	A. R4
2D	i	ADD	A, R5
2E	1	ADD	A, R6
2F	1	ADD	A, R7
30	3	JNB	bit, addr, code
31	2	ACALL	addr
32	1	RETI	code addr

HEX	NUMB.	MNEM.	OPERANDS
CODE	OF BYTES		
33 34	1 2	RLC ADDC	A A, #data
35	2	ADDC	A, data addr
36	1	ADDC	A, @RD
37	1	ADDC	A, @R1
38	1	ADDC	A, R0
39	1	ADDC	A, R1
3A	1	ADDC	A, R2
3B	1	ADDC ADDC	A, R3
3C 3D	1	ADDC	A, R4 A, R5
3E	i	ADDC	A, R6
3F	i	ADDC	A, R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr A
43	3	ORL	data addr, #data
44	2	ORL	A, #data
45	2 1	ORL ORL	A, data addr
46 47	1	ORL	A, @R0 A, @R1
48	i	ORL	A, R0
49	i	ORL	A, R1
4A	1	ORL	A, R2
4B	1	ORL	A, R3
4C	1	ORL	A, R4
4D	1	ORL	A, R5
4E 4F	1	ORL ORL	A, R6
50	2	JNC	A, R7 code addr
51	2	ACALL	code addr
52	2	ANL	data addr, A
53	3	ANL	data addr, #data
54	2	ANL	A, #data
55	2	ANL	A, data addr
56	1	ANL	A, @R0
57	1	ANL	A, @R1
58 59	1	ANL ANL	A, R0 A. R1
59 5A	i	ANL	A, R1 A, R2
5B	i	ANL	A, R3
5C	i	ANL	A R4
5D	1	ANL	A, R5
5E	1	ANL	A, R6
5F	1	ANL	A, R7
60	2	JZ	code addr
61	2	AJMP	code addr
62 63	2 3	XRL XRL	data addr A data addr. #data
64	2	XRL	A, #data
65	2	XRL	A, data addr



80C31/Rev.1.0

HEX	NUMB.	MNEM.	OPERANDS
CODE	OF BYTES		
66	1	XRL	A. @R0
67	1	XRL	A, @R1
68	1	XRL	A, R0
69	1	XRL	A, R1
6A	1	XRL	A, R2
6B	1	XRL	A, R3
6C	1	XRL	A, R4
6D	1	XRL	A, R5
6E 6F	1	XRL	A, R6
70	2	XRL JNZ	A, R7
71	2	ACALL	code addr code addr
72	2	ORL	C, bit addr
73	ī	JMP	@A + DPTR
74	2	MOV	A, #data
75	3	MOV	data addr, #data
76	2	MOV	@ R0, #data
77	2	MOV	@R1,#data
78	2	MOV	R0, #data
79	2	MOV	R1, #data
7A	2	MOV	R2, #data
7B	2	MOV	R3, #data
7C	2	MOV	R4, #data
7D	2	MOV	R5, #data
7E 7F	2 2	MOV MOV	R6, #data
80	2	SJMP	R7, #data code addr
81	2	AJMP	code addr
82	2	ANL	C, bit addr
83	1	MOVC	A, @A + PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr, @R0
87	2	MOV	data addr, @R1
88	2	MOV	data addr, R0
89	2	MOV	data addr, R1
A8	2	MOV	data addr, R2
8B	2	MOV	data addr, R3
8C 8D	2 2	MOV MOV	data addr, R4
8E	2	MOV	data addr, R5
8F	2	MOV	data addr, R6 data addr, R7
90	3	MOV	DPTR, #data
91	2	ACALL	code adddr
92	2	MOV	bit addr, C
93	1	MOVC	A, @A + DPTR
94	2	SUBB	A, #data
95	2	SUBB	A, data addr
96	1	SUBB	A, @R0
97	1	SUBB	A, @R1
98	1	SUBB	A, RO

HEX	NUMB.	MNEM.	OPERANDS
CODE	OF BYTES		
99	1	SUBB	A, R1
9A	1	SUBB	A, R2
9B 9C	1 1	SUBB	A, R3
9D	1	SUBB SUBB	A, R4 A, R5
9E	i	SUBB	A, R6
9F	1	SUBB	A, R7
A0	2	ORL	C, bit addr
A1	2	AJMP	code addr
A2 A3	2 1	MOV INC	C, bit addr DPTR
A4	i	MUL	AB
A5	•	reserved	7.0
A6	2	MOV	@R0, data addr
A7	2	MOV	@R1, data addr
A8	2	MOV	R0, data addr
A9 AA	2 2	MOV MOV	R1, data addr R2, data addr
AB	2	MOV	R3, data addr
AC	2	MOV	R4, data addr
AD	2	MOV	R5, data addr
AE	2	MOV	R6, data addr
AF	2 2	MOV	R7, data addr
B0 B1	2	ANL ACALL	C, bit addr code addr
B2	2	CPL	Bit addr
В3	1	CPL	C
B4	3	CJNE	A, #data, code addr
B5	3	CJNE	A, data addr, code addr
B6	3 3	CJNE	@R0, #data, code addr
B7 B8	3	CJNE CJNE	@R1, #data, code addr R0, #data, code addr
B9	3	CJNE	R1, #data, code addr
BA	3	CJNE	R2, #data, code addr
BB	3	CJNE	R3, #data, code addr
BC	3	CJNE	R5, #data, code addr
BD BE	3	CJNE	R4, #data, code addr
BF	3 3	CJNE CJNE	R6, #data, code addr R7, #data, code addr
Co	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A data addr
C5 C6	2 1	XCH	A, data addr A, @R0
C7	1	XCH	A, @R1
C8	i	XCH	A, R0
C9	1	XCH	A, R1
CA	1	XCH	A, R2
СВ	1	XCH	A, R3

80C31/Rev.1.0

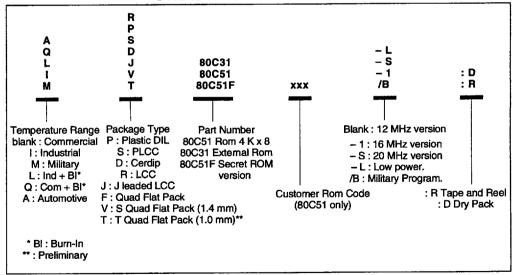




HEX	NUMB. OF BYTES	MNEM.	OPERANDS
CC	1	XCH	A, R4
CD	1	XCH	A, R5
CE	1	XCH	A, R6
CF	1	XCH	A, R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	Α
D5	3	DJNZ	data addr, code addr
D6	1	XCHD	A, @R0
D7	1	XCHD	A, @R1
D8	2	DJNZ	R0, code addr
D9	2	DJNZ	R1, code addr
DA	2 2	DJNZ	R2, code addr
DB	2	DJNZ	R3, code addr
DC	2	DJNZ	R4, code addr
DD	2	DJNZ	R5, code addr
DE	2 2	DJNZ	R6, code addr
DF		DJNZ	R7, code addr
E0	1	MOVX	A, @DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A, @R0
E3	1	MOVX	A, @R1
E4	1	CLR	Α
E5	2	MOV	A, data addr

HEX	NUMB. OF BYTES	MNEM.	OPERANDS
E6	1	MOV	A, @R0
E7	1	MOV	A, @R1
E8	1	MOV	A, RO
E9	1	MOV	A, R1
EA	1	MOV	A, R2
EB	1	MOV	A, R3
EC	1	MOV	A, R4
ED	1	MOV	A, R5
EE	1	MOV	A, R6
EF	1	MOV	A, R7
F0	1	MOVX	@DPTR, A
F1	2	ACALL	code addr
F2	1	MOVX	@R0, A
F3	1	MOVX	@R1, A
F4	1	CPL	Α
F5	2	MOV	data addr, A
F6	1	MOV	@R0, A
F7	1	MOV	@R1, A
F8	1	MOV	R0, A
F9	1	MOV	R1, A
FA	1	MOV	R2, A
FB	1	MOV	R3, A
FC	1	MOV	R4, A
FD	1	MOV	R5, A
FE	1	MOV	R6, A
FF	1	MOV	R7, A

ORDERING INFORMATION





5-24

80C31/Rev.1.0

5868456 0003731 T89 **=**