EZ-USB TRM Appendices

8051 Introduction A-1

8051 Architectural Overview B-1

8051 Hardware Description C-1

Register Tables D-1

APPENDIX A

8051 Introduction

The EZ-USB contains an 8051 core that is binary compatible with the industry standard 8051 instruction set. This appendix provides an overview of the 8051 core features. the topics are:

- New 8051 Features
- Performance Overview
- Software Compatibility
- 803x/805x Feature Comparison
- 8051/DS80C320 Differences

8051 Features

The 8051 core provides the following design features and enhancements to the standard 8051 micro-controller:

- Compatible with industry standard 803x/805x:
  - Standard 8051 instruction set
  - Two full-duplex serial ports
  - Three timers
- High speed architecture:
  - 4 clocks/instruction cycle
  - 2.5X average improvement in instruction execution time over the standard 8051
  - Runs DC to 25-MHz clock
  - Wasted bus cycles eliminated
  - Dual data pointers
- 256 Bytes internal data RAM
- High-speed external memory interface with 16-bit address bus
- Variable length `MOVX` to access fast/slow RAM peripherals
- Fully static synchronous design
- Supports industry standard compilers, assemblers, emulators, and ROM monitors

Performance Overview

The 8051 core has been designed to offer increased performance by executing instructions in a 4-clock bus cycle, as opposed to the 12-clock bus cycle in the standard 8051 (see Figure A-1). The shortened bus timing improves the instruction execution rate for most instructions by a factor of three over the standard 8051 architectures.

Some instructions require a different number of instruction cycles on the 8051 core than they do on the standard 8051. In the standard 8051, all instructions except for `MUL` and `DIV` take one or two instruction cycles to complete. In the 8051 core, instructions can take between one and five instruction cycles to complete. The average speed improvement for the entire instruction set is approximately 2.5X, calculated as follows:

Number of Opcodes	Speed Improvement
150	3.0X
51	1.5X
43	2.0X
2	2.4X
Total: 255	Average: 2.5X
Note: Comparison is for 8051 and standard 8051 running at the same clock frequency.	

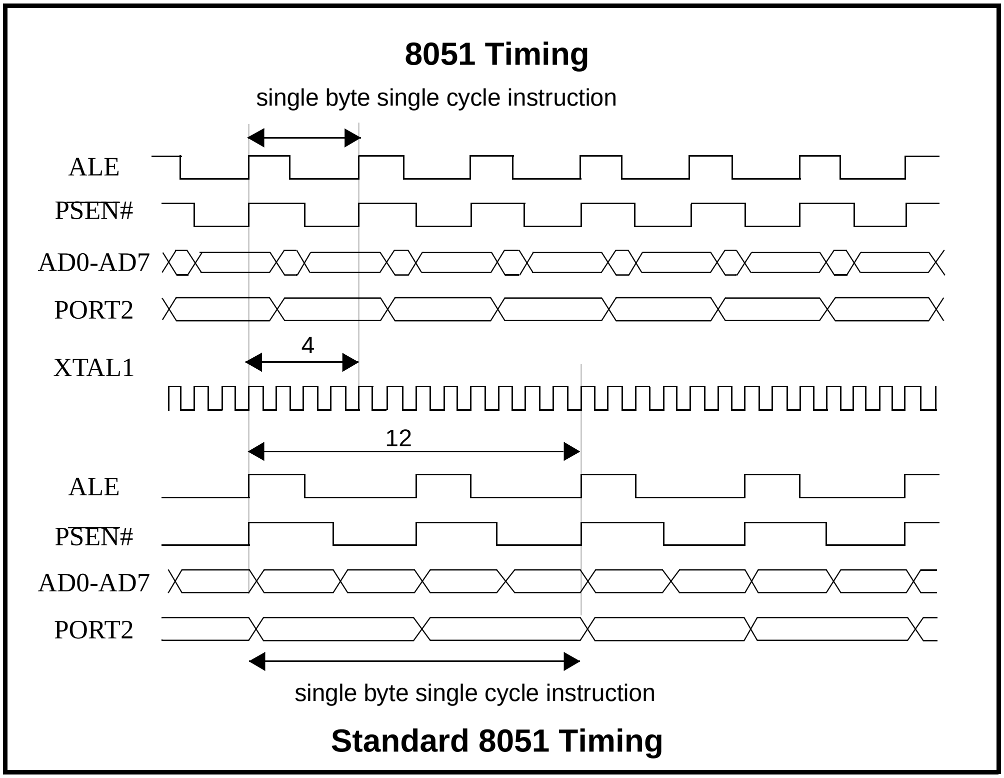

Figure A-1: Comparative Timing of 8051 and Industry Standard 8051

Software Compatibility

The 8051 core is object code compatible with the industry standard 8051 micro-controller. That is, object code compiled with an industry standard 8051 compiler or assembler will execute on the 8051 core and will be functionally equivalent. However, because the 8051 core uses a different instruction timing than the standard 8051, existing code with timing loops may require modification.

The “Instruction Set” in Table B-2 on page B-7 lists the number of instruction cycles required to perform each instruction on the 8051 core. The 8051 instruction cycle timing and number of instruction cycles required for each instruction are compatible with the Dallas Semiconductor DS80C320.

803x/805x Feature Comparison

Table A-1 provides a feature-by-feature comparison of the 8051 core and several common 803x/805x configurations.

Table A-1 : Feature Summary of 8051 Core and Common 803x/805x Configurations

Feature	**Intel**				**Dallas DS80C320**	**Anchor 8051**
	8031	**8051**	**80C32**	**80C52**		
Clocks per instruction cycle	12	12	12	12	4	4
Program / Data Memory	-	4 KB ROM	-	8 KB ROM	-	8 K RAM
Internal RAM	128 bytes	128 bytes	256 bytes	256 bytes	256 bytes	256 bytes
Data Pointers	1	1	1	1	2	2
Serial Ports	1	1	1	1	2	2
16-bit Timers	2	2	3	3	3	3
Interrupt sources (total of int. and ext.)	5	5	6	6	13	13
Stretch memory cycles	no	no	no	no	yes	yes

8051 Core/ DS80C320 Differences

The 8051 core is similar to the DS80C320 in terms of hardware features and instruction cycle timing. However, there are some important differences between the 8051 core and the DS80C320.

Serial Ports

The 8051 core does not implement serial port framing error detection and does not implement slave address comparison for multiprocessor communications. Therefore, the 8051 core also does not implement the following SFRs: SADDR0, SADDR1, SADEN0, and SADEN1.

Timer 2

The 8051 core does not implement Timer 2 downcounting mode or the downcount enable bit (TMOD2, bit 0). Also, the 8051 core does not implement Timer 2 output enable (T2OE) bit (TMOD2, bit 1). Therefore, the TMOD2 SFR is also not implemented in the 8051 core.

Also, the 8051 core Timer 2 overflow output is active for one clock cycle. In the DS80C320, the Timer 2 overflow output is a square wave with a 50% duty cycle.

Timed Access Protection

The 8051 core does not implement timed access protection and therefore, does not implement the TA SFR.

Watchdog Timer

The EZ-USB/8051 does not implement a watchdog timer.

APPENDIX

B

8051 Architectural Overview

8051 Architectural Overview

This appendix provides a technical overview and description of the 8051 core architecture.

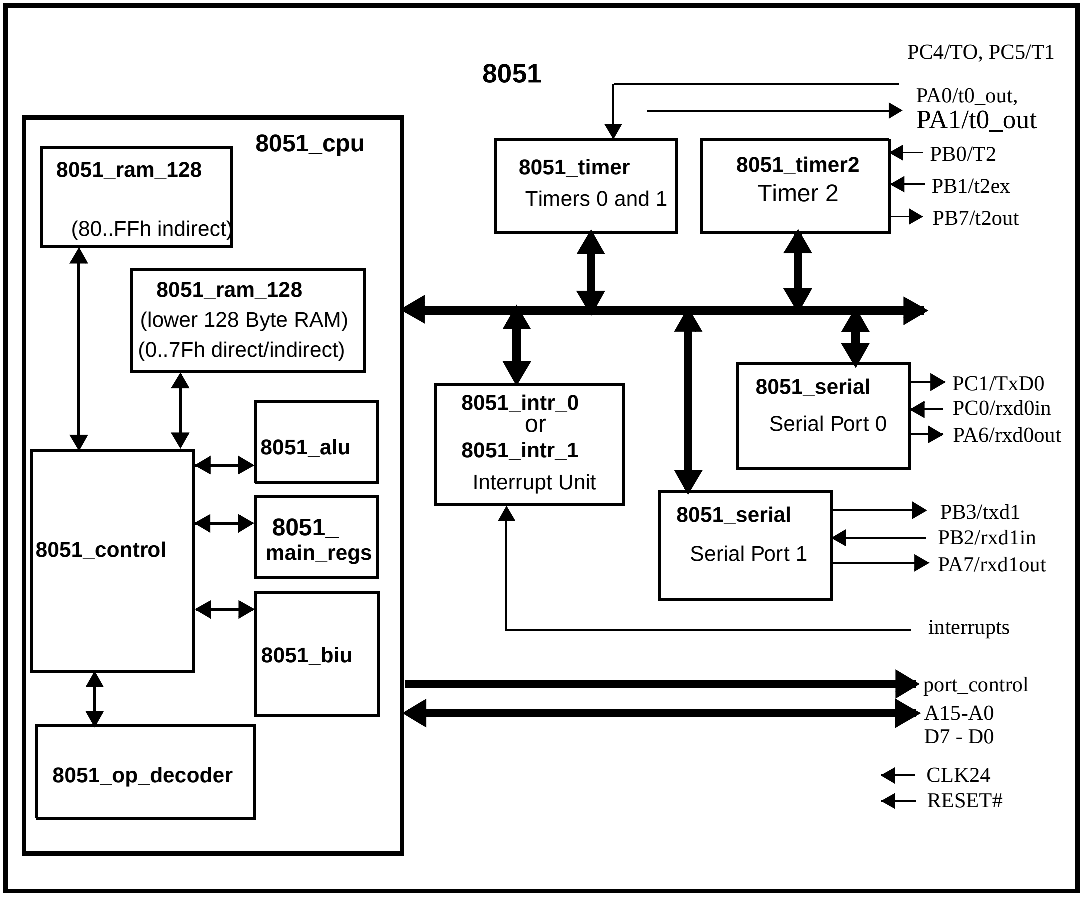

Figure B-1 : 8051 Block Diagram

Memory Organization

Memory organization in the 8051 core is similar to that of the industry standard 8051. There are three distinct memory areas: program memory (ROM), data memory (external RAM), and registers (internal RAM).

Program Memory

The EZ-USB provides 8K of data that is mapped as both program and data memory at addresses 0x0000-0x1B3F. In addition, the bulk endpoint buffers may be used as external data memory if they are not used as endpoint buffers. See Chapter 3, "EZ-USB Memory" for more details.

External RAM

The EZ-USB chip has dedicated address and data pins, so port 2 and port 0 are not used to access the memory bus. As shown in Chapter 3, "EZ-USB Memory", the EZ-USB is expandable to over 100K of external program and data memory.

Internal RAM

The internal RAM (Figure B-2) consists of:

- 128 bytes of registers and scratch pad memory accessible through direct or indirect addressing (addresses 00h–7Fh).
- A 128 register space for special function registers (SFRs) accessible through direct addressing (addresses 80h–FFh).
- Upper 128 bytes of scratch pad memory accessible through indirect addressing (addresses 80h–FFh).

Although the SFR space and the upper 128 bytes of RAM share the same address range, the actual address space is separate and is differentiated by the type of addressing. Direct addressing accesses the SFRs, and indirect addressing accesses the upper 128 bytes of RAM.

The lower 128 bytes are organized as shown in Figure B-2. The lower 32 bytes (0x00-0xIF) form four banks of eight registers (R0–R7). Two bits on the program status word (PSW) select which bank is in use. The next 16 bytes (0x20 - 0x2F) form a block of bit-addressable memory space at *bit addresses* 0h-7Fh. All of the bytes in the lower 128 bytes are accessible through direct or indirect addressing.

The SFRs occupy addresses 80h–FFh and are only accessible through direct addressing. Most SFRs are reserved for specific functions as described in the “Special Function Registers” on page B-15.

SFR addresses ending in 0h or 8h are bit-addressable.

Instruction Set

All 8051 instructions are binary code compatible and perform the same functions as they do with the industry standard 8051. The effects of these instructions on bits, flags, and other status functions is identical to the industry standard 8051. However, the timing of the instructions is different, both in terms of number of clock cycles per instruction cycle and timing within the instruction cycle.

Figure B-2 lists the 8051 instruction set and the number of instruction cycles required to complete each instruction. Table B-1 defines the symbols and mnemonics used in Table B-2

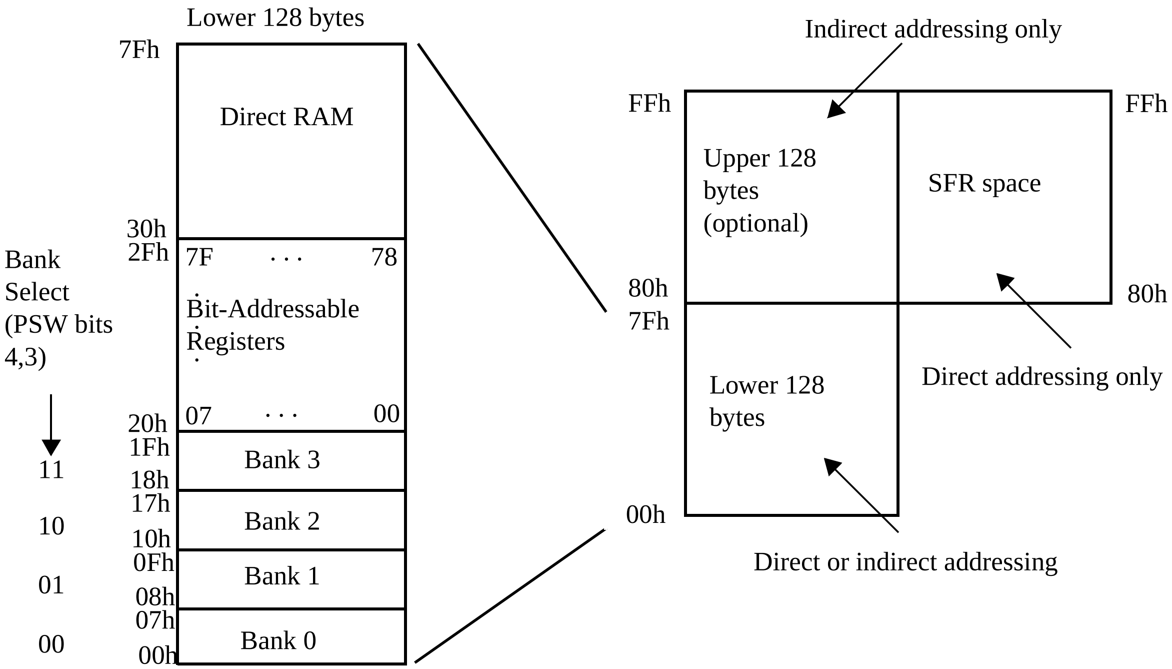

Figure B-2: Internal RAM Organization

Table B-1: Legend for Instruction Set Table

Symbol	**Function**
A	Accumulator
Rn	Register R7–R0
direct	Internal register address
@Ri	Internal register pointed to by R0 or R1 (except MOVX)
rel	Two’s complement offset byte
bit	Direct bit address
#data	8-bit constant
#data 16	16-bit constant
addr 16	16-bit destination address
addr 11	11-bit destination address

Table B-2: 8051 Instruction Set

Mnemonic	Description	Byte	Instr. Cycles	Hex Code
Arithmetic				
ADD A, Rn	Add register to A	1	1	28-2F
ADD A, direct	Add direct byte to A	2	2	25
ADD A, @Ri	Add data memory to A	1	1	26-27
ADDC A, #data	Add immediate to A	2	2	24
ADDC A, Rn	Add register to A with carry	1	1	38-3F
ADDC A, direct	Add direct byte to A with carry	2	2	35
ADDC A, @Ri	Add data memory to A with carry	1	1	36-37
ADDC A, #data	Add immediate to A with carry	2	2	34
SUBB A, Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A, direct	Subtract direct byte from A with borrow	2	2	95
SUBB A, @Ri	Subtract data memory from A with borrow	1	1	96-97
SUBB A, #data	Subtract immediate from A with borrow	2	2	94
INC A	increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC direct	Increment direct byte	2	2	05
INC @ Ri	Increment data memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement Register	1	1	18-1F
DEC direct	Decrement direct byte	2	2	15
DEC @Ri	Decrement data memory	1	1	16-17
INC DPTR	Increment data pointer	1	3	A3
MUL AB	Multiply A by B	1	5	A4
DIV AB	Divide A by B	1	5	84
DA A	Decimal adjust A	1	1	D4

Table B-2: 8051 Instruction Set

Mnemonic	**Description**	**Byte**	**Instr. Cycles**	**Hex Code**
Logical				
ANL, Rn	AND register to A	1	1	58-5F
ANL A, direct	AND direct byte to A	2	2	55
ANL A, @Ri	AND data memory to A	1	1	56-57
ANL A, #data	AND immediate to A	2	2	54
ANL direct, A	AND A to direct byte	2	2	52
ANL direct, #data	AND immediate data to direct byte	3	3	53
ORL A, Rn	OR register to A	1	1	48-4F
ORL A, direct	OR direct byte to A	2	2	45
ORL A, @Ri	OR data memory to A	1	1	46-47
ORL A, #data	OR immediate to A	2	2	44
ORL direct, A	OR A to direct byte	2	2	42
ORL direct, #data	OR immediate data to direct byte	3	3	43
XRL A, Rn	Exclusive-OR register to A	1	1	68-6F
XRL A, direct	Exclusive-OR direct byte to A	2	2	65
XRL A, @Ri	Exclusive-OR data memory to A	1	1	66-67
XRL A, #data	Exclusive-OR immediate to A	2	2	64
XRL direct, A	Exclusive-OR A to direct byte	2	2	62
XRL direct, #data	Exclusive-OR immediate data to direct byte	3	3	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap nibbles of a	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RRA	Rotate A right	1	1	03

Table B-2: 8051 Instruction Set

Mnemonic	**Description**	**Byte**	**Instr. Cycles**	**Hex Code**
RRC A	Rotate A right through carry	1	1	13
Data Transfer				
MOV A, Rn	Move register to A	1	1	E8-EF
MOV A, direct	Move direct byte to A	2	2	E5
MOV A, @Ri	Move data memory to A	1	1	E6-E7
MOV A, #data	Move immediate to A	2	2	74
MOV Rn, A	Move A to register	1	1	F8-FF
MOV Rn, direct	Move direct byte to register	2	2	A8-AF
MOV Rn, #data	Move immediate to register	2	2	78-7F
MOV direct, A	Move A to direct byte	2	2	F5
MOV direct, Rn	Move register to direct byte	2	2	88-8F
MOV direct, direct	Move direct byte to direct byte	3	3	85
MOV direct, @Ri	Move data memory to direct byte	2	2	86-87
MOV direct, #data	Move immediate to direct byte	3	3	75
MOV @Ri, A	MOV A to data memory	1	1	F6-F7
MOV @Ri, direct	Move direct byte to data memory	2	2	A6-A7
MOV @Ri, #data	Move immediate to data memory	2	2	76-77
MOV DPTR, #data	Move immediate to data pointer	3	3	90
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3	93
MOVC A, @A+PC	Move code byte relative PC to A	1	3	83
MOVX A, @Ri	Move external data (A8) to A (Note 1)	1	2-9*	E2-E3
MOVX A, @DPTR	Move external data (A16) to A	1	2-9*	E0
MOVX @Ri, A	Move A to external data (A8) (Note 1)	1	2-9*	F2-F3
MOVX @DPTR, A	Move A to external data (A16)	1	2-9*	F0

Table B-2: 8051 Instruction Set

Mnemonic	Description	Byte	Instr. Cycles	Hex Code
PUSH direct	Push direct byte onto stack	2	2	C0
POP direct	Pop direct byte from stack	2	2	D0
XCH A, Rn	Exchange A and register	1	1	C8-CF
XCH A, direct	Exchange A and direct byte	2	2	C5
XCH A, @Ri	Exchange A and data memory	1	1	C6-C7
XCHD A, @Ri	Exchange A and data memory nibble	1	1	D6-D7
** Number of cycles is user-selectable. See “Stretch Memory Cycles (Wait States)” on page B-13.*				
Boolean				
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	2	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	2	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	2	B2
ANL C, bit	AND direct bit to carry	2	2	82
ANL C, /bit	AND direct bit inverse to carry	2	2	B0
ORL C, bit	OR direct bit to carry	2	2	72
ORL C, /bit	OR direct bit inverse to carry	2	2	A0
MOV C, bit	Move direct bit to carry	2	2	A2
MOV bit, C	Move carry to direct bit	2	2	92
Branching				
ACALL addr 11	Absolute call to subroutine	2	3	11-F1
LCALL addr 16	Long call to subroutine	3	4	12
RET	Return from subroutine	1	4	22

Table B-2: 8051 Instruction Set

Mnemonic	Description	Byte	Instr. Cycles	Hex Code
RETI	Return from interrupt	1	4	32
AJMP addr 11	Absolute jump unconditional	2	3	01-E1
LJMP addr 16	Long jump unconditional	3	4	02
SJMP rel	Short jump (relative address)	2	3	80
JC rel	Jump on carry = 1	2	3	40
JNC rel	Jump on carry = 0	2	3	50
JB bit, rel	Jump on direct bit = 1	3	4	20
JNB bit, rel	Jump on direct bit = 0	3	4	30
JBC bit, rel	Jump on direct bit = 1 and clear	3	4	10
JMP @ A+DPTR	Jump indirect relative DPTR	1	3	73
JZ rel	Jump on accumulator = 0	2	3	60
JNZ rel	Jump on accumulator /= 0	2	3	70
CJNE A, direct, rel	Compare A, direct JNE relative	3	4	B5
CJNE A, #d, rel	Compare A, immediate JNE relative	3	4	B4
CJNE Rn, #d, rel	Compare reg, immediate JNE relative	3	4	B8-BF
CJNE @ Ri, #d, rel	Compare Ind, immediate JNE relative	3	4	B6-B7
DJNZ Rn, rel	Decrement register, JNZ relative	2	3	D8-DF
DJNZ direct, rel	Decrement direct byte, JNZ relative	3	4	D5
Miscellaneous				
NOP	No operation	1	1	00

There is an additional reserved opcode (A5) that performs the same function as NOP. All mnemonics are copyrighted. Intel Corporation 1980.

(Note 1) These instructions are not supported by EZ-USB chips since they do not implement ports P0 or P2.

Instruction Timing

Instruction cycles in the 8051 core are 4 clock cycles in length, as opposed to the 12 clock cycles per instruction cycle in the standard 8051. This translates to a 3X improvement in execution time for most instructions.

Some instructions require a different number of instruction cycles on the 8051 core than they do on the standard 8051. In the standard 8051, all instructions except for `MUL` and `DIV` take one or two instruction cycles to complete. In the 8051 core, instructions can take between one and five instruction cycles to complete.

For example, in the standard 8051, the instructions `MOVX A, @DPTR` and `MOV direct, direct` each take 2 instruction cycles (24 clock cycles) to execute. In the 8051 core, `MOVX A, @DPTR` takes two instruction cycles (8 clock cycles) and `MOV direct, direct` takes three instruction cycles (12 clock cycles). Both instructions execute faster on the 8051 core than they do on the standard 8051, but require different numbers of clock cycles.

For timing of real-time events, use the numbers of instruction cycles from Table B-1 to calculate the timing of software loops. The bytes column indicates the number of memory accesses (bytes) needed to execute the instruction. In most cases, the number of bytes is equal to the number of instruction cycles required to complete the instruction. However, as indicated, there are some instructions (for example, `DIV` and `MUL`) that require a greater number of instruction cycles than memory accesses.

By default, the 8051 core timer/counters run at 12 clock cycles per increment so that timer-based events have the same timing as with the standard 8051. The timers can also be configured to run at 4 clock cycles per increment to take advantage of the higher speed of the 8051 core.

CPU Timing

As previously stated, an 8051 core instruction cycle consists of 4 *CLK24* cycles. Each *CLK24* cycle forms a CPU cycle. Therefore, an instruction cycle consists of 4 CPU cycles: C1, C2, C3, and C4, as illustrated in Figure B-3. Various events occur in each CPU cycle, depending on the type of instruction being executed. The labels C1, C2, C3, and C4 in timing descriptions refer to the 4 CPU cycles within a particular instruction cycle.

The execution for instruction *n* is performed during the fetch of instruction *n*+1. Data writes occur during fetch of instruction *n*+2. The level sensitive interrupts are sampled with the rising edge of *CLK24* at the end of C3.

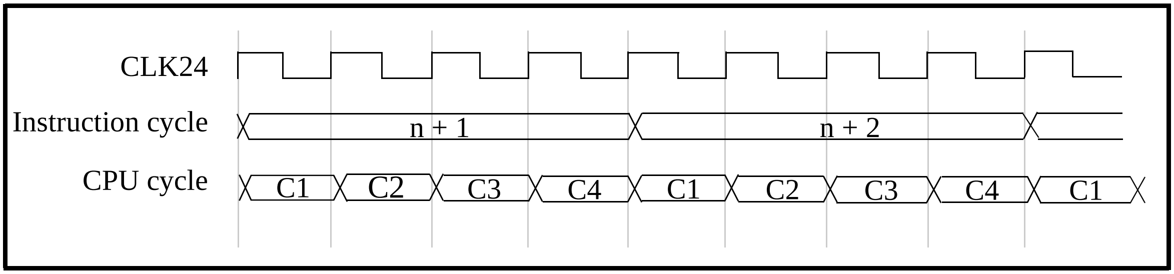

Figure B-3: CPU Timing for Single-Cycle Instruction

Stretch Memory Cycles (Wait States)

The stretch memory cycle feature enables application software to adjust the speed of data memory access. The 8051 core can execute the `MOVX` instruction in as few as 2 instruction cycles. However, it is sometimes desirable to stretch this value; for example to access slow memory or slow memory-mapped peripherals such as UARTs or LCDs.

The three LSBs of the Clock Control Register (at SFR location 8Eh) control the stretch value. You can use stretch values between zero and seven. A stretch value of zero adds zero instruction cycles, resulting in `MOVX` instructions executing in two instruction cycles. A stretch value of seven adds seven instruction cycles, resulting in `MOVX` instructions executing in nine instruction cycles. The stretch value can be changed dynamically under program control.

By default, the stretch value resets to one (three cycle `MOVX`). For full-speed data memory access, the software must set the stretch value to zero. The stretch value affects only data memory access (not program memory).

The stretch value affects the width of the read/write strobe and all related timing. Using a higher stretch value results in a wider read/write strobe, which allows the memory or peripheral more time to respond.

Table B-3 lists the data memory access speeds for stretch values zero through seven. MD2–0 are the three LSBs of the Clock Control Register (CKCON.2–0).

Table B-3: Data Memory Stretch Values

MD2	MD1	MD0	Memory Cycles	Read/Write Strobe Width (Clocks)	Strobe Width @ 24MHz
0	0	0	2	2	83.3 ns
0	0	1	3 (default)	4	166.7 ns
0	1	0	4	8	333.3 ns
0	1	1	5	12	500 ns
1	0	0	6	16	666.7 ns
1	0	1	7	20	833.3 ns
1	1	0	8	24	1000 ns
1	1	1	9	28	1166.7 ns

Dual Data Pointers

The 8051core employs dual data pointers to accelerate data memory block moves. The standard 8051 data pointer (DPTR) is a 16-bit value used to address external data RAM or peripherals. The 8051 maintains the standard data pointer as DPTR0 at SFR locations 82h (DPL0) and 83h (DPH0). It is not necessary to modify existing code to use DPTR0.

The 8051 core adds a second data pointer (DPTR1) at SFR locations 84h (DPL1) and 85h (DPH1). The SEL bit in the DPTR Select register, DPS (SFR 86h), selects the active pointer. When SEL = 0, instructions that use the DPTR will use DPL0 and DPH0. When SEL = 1, instructions that use the DPTR will use DPL1 and DPH1. SEL is the bit 0 of SFR location 86h. No other bits of SFR location 86h are used.

All DPTR-related instructions use the currently selected data pointer. To switch the active pointer, toggle the SEL bit. The fastest way to do so is to use the increment instruction (`INC DPS`). This requires only one instruction to switch from a source address to a destination address, saving application code from having to save source and destination addresses when doing a block move.

Using dual data pointers provides significantly increased efficiency when moving large blocks of data.

The SFR locations related to the dual data pointers are:

82h	DPL0	DPTR0 low byte
83h	DPH0	DPTR0 high byte
84h	DPL1	DPTR1 low byte
85h	DPH1	DPTR1 high byte
86h	DPS	DPTR Select (Bit 0)

Special Function Registers

The Special Function Registers (SFRs) control several of the features of the 8051. Most of the 8051 core SFRs are identical to the standard 8051 SFRs. However, there are additional SFRs that control features that are not available in the standard 8051.

Table B-4 lists the 8051 core SFRs and indicates which SFRs are not included in the standard 8051 SFR space.

In Table B-5, SFR bit positions that contain a 0 or a 1 cannot be written to and, when read, always return the value shown (0 or 1). SFR bit positions that contain “-” are available but not used. Table B-5 lists the reset values for the SFRs.

The following SFRs are related to CPU operation and program execution:

81h	SP	Stack Pointer
D0h	PSW	Program Status Word ()
E0h	ACC	Accumulator Register
F0h	B	B Register

Table B-6 list the functions of the bits in the PSW SFR. Detailed descriptions of the remaining SFRs appear with the associated hardware descriptions in Appendix C of this databook.

Table B-4: Special Function Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Addr
SP									81h
DPL0									82h
DPH0									83h
DPL1(1)									84h
DPH1(1)									85h
DPS(1)	0	0	0	0	0	0	0	SEL	86h
PCON	SMOD0	-	1	1	GF1	GF0	STOP	IDLE	87h
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88h
TMOD	GATE	C/T	M1	M0	GATE	$C/\overline{T}$	M1	M0	89h
TL0									8Ah
TL1									8Bh
TH0									8Ch
TH1									8Dh
CKCON(1)	-	-	T2M	T1M	T0M	MD2	MD1	MD0	8Eh
SPC_FNC(1)	0	0	0	0	0	0	0	WRS	8Fh
EXIF(1)	IE5	IE4	IE3	IE2	1	0	0	0	91h

Table B-4: Special Function Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Addr
MPAGE(1)									92h
SCON0	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	98h
SBUF0									99h
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	A8h
IP	1	PS1	PT2	PS0	PT1	PX1	PT0	PX0	B8h
SCON1(1)	SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	C0h
SBUF1(1)									C1h
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{\text{T2}}$	CP/$\overline{\text{RL2}}$	C8h
RCAP2L									CAh
RCAP2H									CBh
TL2									CCh
TH2									CDh
PSW	CY	AC	F0	RS1	RS0	OV	F1	P	D0h
EICON(1)	SMOD1	1	EPFI	_____	____	0	0	0	D8h
ACC									E0H
EIE(1)	1	1	1	EWDI	EX5	EX4	EX3	EX2	E8h
B									F0h
EIP(1)	1	1	1	PWDI	PX5	PX4	PX3	PX2	F8h

(1) Not part of standard 8051 architecture.

Table B-5: Special Function Register Reset Values

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Addr
SP	0	0	0	0	0	1	1	1	81h
DPL0	0	0	0	0	0	0	0	0	82h
DPH0	0	0	0	0	0	0	0	0	83h
DPL1(1)	0	0	0	0	0	0	0	0	84h
DPH1(1)	0	0	0	0	0	0	0	0	85h
DPS(1)	0	0	0	0	0	0	0	0	86h
PCON	0	0	1	1	0	0	0	0	87h
TCON	0	0	0	0	0	0	0	0	88h
TMOD	0	0	0	0	0	0	0	0	89h
TL0	0	0	0	0	0	0	0	0	8Ah
TL1	0	0	0	0	0	0	0	0	8Bh
TH0	0	0	0	0	0	0	0	0	8Ch
TH1	0	0	0	0	0	0	0	0	8Dh
CKCON(1)	0	0	0	0	0	0	0	1	8Eh
SPC_FNC(1)	0	0	0	0	0	0	0	0	8Fh
EXIF(1)	0	0	0	0	1	0	0	0	91h
MPAGE(1)	0	0	0	0	0	0	0	0	92h
SCON0	0	0	0	0	0	0	0	0	98h
SBUF0	0	0	0	0	0	0	0	0	99h
IE	0	0	0	0	0	0	0	0	A8h
IP	1	0	0	0	0	0	0	0	B8h
SCON1(1)	0	0	0	0	0	0	0	0	C0h
SBUF1(1)	0	0	0	0	0	0	0	0	C1h
T2CON	0	0	0	0	0	0	0	0	C8h
RCAP2L	0	0	0	0	0	0	0	0	CAh
RCAP2H	0	0	0	0	0	0	0	0	CBh
TL2	0	0	0	0	0	0	0	0	CCh
TH2	0	0	0	0	0	0	0	0	CDh

Table B-5: Special Function Register Reset Values

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Addr
PSW	0	0	0	0	0	0	0	0	D0h
EICON(1)	0	1	0	0	0	0	0	0	D8h
ACC	0	0	0	0	0	0	0	0	E0H
EIE(1)	1	1	1	0	0	0	0	0	E8h
B	0	0	0	0	0	0	0	0	F0h
EIP(1)	1	1	1	0	0	0	0	0	F8h
(1) Not part of standard 8051 architecture.									

Table B-6: PSW Register - SFR D0h

Bit	Function
PSW.7	CY - Carry flag. This is the **unsigned** carry bit. The CY flag is set when an arithmetic operation results in a carry from bit 7 to bit 8, and cleared otherwise. In other words, it acts as a virtual bit 8. The CY flag is cleared on multiplication and division.
PSW.6	AC - Auxiliary carry flag. Set to 1 when the last arithmetic operation resulted in a carry into (during addition) or borrow from (during subtraction) the high order nibble, otherwise cleared to 0 by all arithmetic operations.
PSW.5	F0 - User flag 0. Bit-addressable, general purpose flag for software control.
PSW.4	RS1 - Register bank select bit 1. used with RS0 to select a register bank in internal RAM.
PSW.3	RS0 - Register bank select bit 0, decoded as: **RS1 RS0 Banks Selected** 0 0 Register bank 0, addresses 00h-07h 0 1 Register bank 1, addresses 08h-0Fh 1 0 Register bank 2, addresses 10h-17h 1 1 Register bank 3, addresses 18h-1Fh
PSW.2	OV - Overflow flag. This is the **signed** carry bit. The OV flag is set when a positive sum exceeds 7fh, or a negative sum (in two’s compliment notation) exceeds 80h. On a multiply, if OV = 1, the result of the multiply is greater than FFh. On a divide, OV = 1 on a divide by 0.
PSW.1	F1 - User flag 1. Bit-addressable, general purpose flag for software control.
PSW.0	P - Parity flag. Set to 1 when the modulo-2 sum of the 8 bits in the accumulator is 1 (odd parity), cleared to 0 on even parity.

APPENDIX

C

8051 Hardware Description

This chapter provides technical data about the 8051 core hardware operation and timing. The topics are:

- Timers/Counters
- Serial Interface
- Interrupts
- Reset
- Power Saving Modes

Timers/ Counters

The 8051 core includes three timer/counters (Timer 0, Timer 1, and Timer 2). Each timer/counter can operate as either a timer with a clock rate based on the *CLK24* pin, or as an event counter clocked by the *T0* pin (Timer 0), *T1* pin (Timer 1), or the *T2* pin (Timer 2).

Each timer/counter consists of a 16-bit register that is accessible to software as two SFRs:

- Timer 0 - TL0 and TH0
- Timer 1 - TL1 and TH1
- Timer 2 - TL2 and TH2

803x/805x Compatibility

The implementation of the timers/counters is similar to that of the Dallas Semiconductor DS80C320. Table C-1 summarizes the differences in timer/counter implementation between the Intel 8051, the Dallas Semiconductor DS80C320, and the 8051 core.

Table C-1 : Timer/Counter Implementation Comparison

Feature	Intel 8051	Dallas DS80C320	8051
Number of timers	2	3	3
Timer 0/1 overflow available as output signals	not implemented	not implemented	T0OUT, T1OUT (one CLK24 pulse)
Timer 2 output enable	n/a	implemented	not implemented
Timer 2 downcount enable	n/a	implemented	not implemented
Timer 2 overflow available as output signal	n/a	implemented	T2OUT (one CLK24 pulse)

Timers 0 and 1

Timers 0 and 1 each operate in four modes, as controlled through the TMOD SFR (Table C-2) and the TCON SFR (Table C-3). The four modes are:

- 13-bit timer/counter (mode 0)
- 16-bit timer/counter (mode 1)
- 8-bit counter with auto-reload (mode 2)
- Two 8-bit counters (mode 3, Timer 0 only)

Mode 0

Mode 0 operation, illustrated in Figure C-1, is the same for Timer 0 and Timer 1. In mode 0, the timer is configured as a 13-bit counter that uses bits 0-4 of TL0 (or TL1) and all 8 bits of TH0 (or TH1). The timer enable bit (TR0/TR1) in the TCON SFR starts the timer. The $C/\overline{T}$ bit selects the timer/counter clock source, CLK24 or the T0/T1 pins.

The timer counts transitions from the selected source as long as the GATE bit is 0, or the GATE bit is 1 and the corresponding interrupt pin (INT0# or INT1#) is 1.

When the 13-bit count increments from 1FFFh (all ones), the counter rolls over to all zeros, the TF0 (or TF1) bit is set in the TCON SFR, and the T0OUT (or T1OUT) pin goes high for one clock cycle.

The upper 3 bits of TL0 (or TL1) are indeterminate in mode 0 and must be masked when the software evaluates the register.

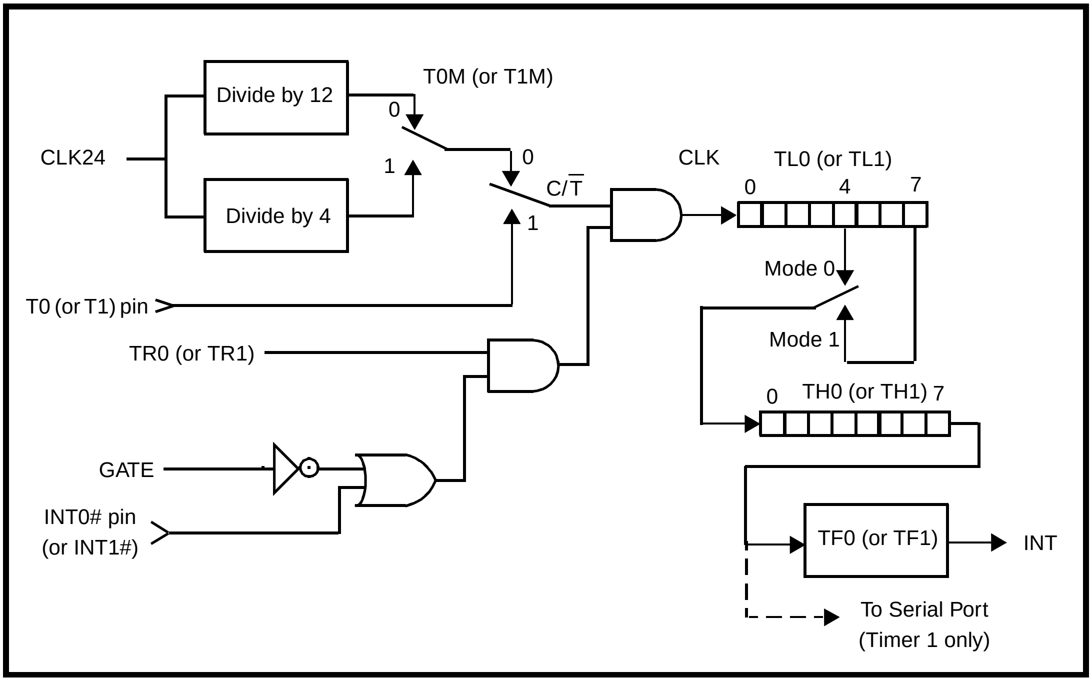

Figure C-1: Timer 0/1 - Modes 0 and 1

Mode 1

Mode 1 operation is the same for Timer 0 and Timer 1. In mode 1, the timer is configured as a 16-bit counter. As illustrated in Figure C-1, all 8 bits of the LSB register (TL0 or TL1) are used. The counter rolls over to all zeros when the count increments from FFFFh. Otherwise, mode 1 operation is the same as mode 0.

Table C-2: TMOD Register - SFR 89h

Bit	Function
TMOD.7	GATE - Timer 1 gate control. When GATE = 1, Timer 1 will clock only when INT1# = 1 and TR1 (TCON.6) = 1. When GATE = 0, Timer 1 will clock only when TR1 = 1, regardless of the state of INT1#.
TMOD.6	$C/\overline{T}$ - Counter/Timer select. When $C/\overline{T}$ = 0, Timer 1 is clocked by CLK24/4 or CLK24/12, depending on the state of T1M (CKCON.4). When $C/\overline{T}$ = 1, Timer 1 is clocked by the T1 pin.
TMOD.5	M1 - Timer 1 mode select bit 1.
TMOD.4	M0 - Timer 1 mode select bit 0, decoded as: **M1 M0 Mode** 0 0 Mode 0 : 13-bit counter 0 1 Mode 1 : 16-bit counter 1 0 Mode 2 : 8-bit counter with auto-reload 1 1 Mode 3 : Timer 1 stopped
TMOD.3	GATE - Timer 0 gate control, When GATE = 1, Timer 0 will clock only when INT0 = 1 and TR0 (TCON.4) = 1. When GATE = 0, Timer 0 will clock only when TR0 = 1, regardless of the state of INT0.
TMOD.2	$C/\overline{T}$ - Counter/Timer select. When $C/\overline{T}$ = 0, Timer 0 is clocked by CLK24/4 or CLK24/12, depending on the state of T0M (CKCON.3). When $C/\overline{T}$ = 1, Timer 0 is clocked by the T0 pin.
TMOD.1	M1 - Timer 0 mode select bit 1.
TMOD.0	M0 - Timer 0 mode select bit 0, decoded as: **M1 M0 Mode** 0 0 Mode 0 : 13-bit counter 0 1 Mode 1 : 16-bit counter 1 0 Mode 2 : 8-bit counter with auto-reload 1 1 Mode 3 : Two 8-bit counters

Table C-3: TCON Register - SRF 88h

Bit	Function
TCON.7	TF1 - Timer 1 overflow flag. Set to 1 when the Timer 1 count overflows and cleared when the processor vectors to the interrupt service routine.
TCON.6	TR1 - Timer 1 run control. Set to 1 to enable counting on Timer 1.
TCON.5	TF0 - Timer 0 overflow flag. Set to 1 when the Timer 0 count overflows and cleared when the processor vectors to the interrupt service routine.
TCON.4	TR0 - Timer 0 run control. Set to 1 to enable counting on Timer 0.
TCON.3	IE1 - Interrupt 1 edge detect. If external interrupt 1 is configured to be edge-sensitive (IT1 = 1), IE1 is set by hardware when a negative edge is detected on the INT1 pin and is automatically cleared when the CPU vectors to the corresponding interrupt service routine. In this case, IE1 can also be cleared by software. If external interrupt 1 is configured to be level-sensitive (IT1 = 0), IE1 is set when the INT1# pin is 0 and cleared when the INT1# pin is 1. In level-sensitive mode, software cannot write to IE1.
TCON.2	IT1 - Interrupt 1 type select. INT1 is detected on falling edge when IT1 = 1; INT1 is detected as a low level when IT1 = 0.
TCON.1	IE0 - Interrupt 0 edge detect. If external interrupt 0 is configured to be edge-sensitive (IT0 = 1), IE0 is set by hardware when a negative edge is detected on the INT0 pin and is automatically cleared when the CPU vectors to the corresponding interrupt service routine. In this case, IE0 can also be cleared by software. If external interrupt 0 is configured to be level-sensitive (IT0 = 0), IE0 is set when the INT0# pin is 0 and cleared when the INT0# pin is 1. In level-sensitive mode, software cannot write to IE0.
TCON.0	IT0 - Interrupt 0 type select. INT0 is detected on falling edge when IT0 = 1; INT0 is detected as a low level when IT0 = 0.

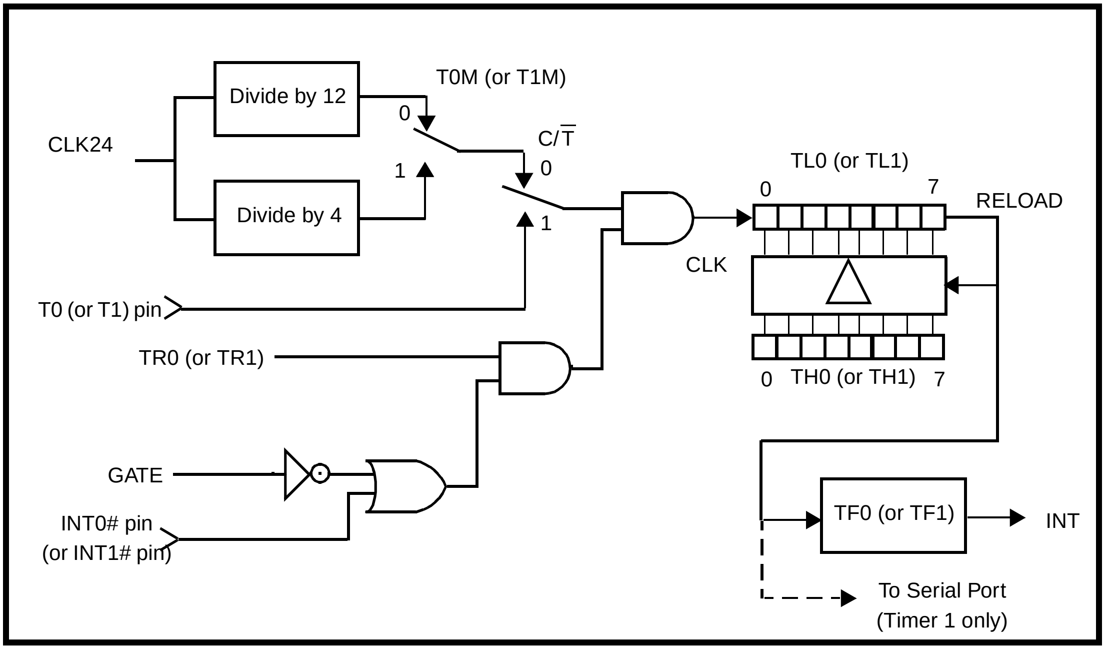

Figure C-2: Timer 0/1 - Mode 2

Mode 2

Mode 2 operation is the same for Timer 0 and Timer 1. In mode 2, the timer is configured as an 8-bit counter, with automatic reload of the start value. The LSB register (TL0 or TL1) is the counter and the MSB register (TH0 or TH1) stores the reload value.

As illustrated in Figure C-2, mode 2 counter control is the same as for mode 0 and mode 1. However, in mode 2, when TL*n* increments from FFh, the value stored in TH*n* is reloaded into TL*n*.

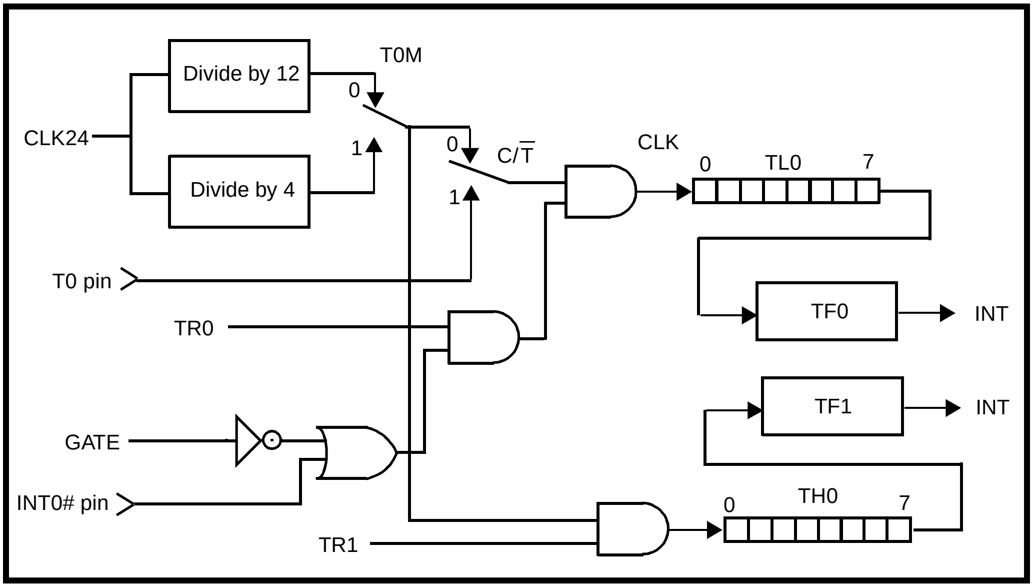

Figure C-3: Timer 0 - Mode 3

Mode 3

In mode 3, Timer 0 operates as two 8-bit counters and Timer 1 stops counting and holds its value.

As shown in Figure C-3, TL0 is configured as an 8-bit counter controlled by the normal Timer 0 control bits. TL0 can either count CLK24 cycles (divided by 4 or by 12) or high-to-low transitions on *T0,* as determined by the C/$\overline{T}$ bit. The GATE function can be used to give counter enable control to the INT0# pin.

TH0 functions as an independent 8-bit counter. However, TH0 can only count CLK24 cycles (divided by 4 or by 12). The Timer 1 control and flag bits (TR1 and TF1) are used as the control and flag bits for TH0.

When Timer 0 is in mode 3, Timer 1 has limited usage because Timer 0 uses the Timer 1 control bit (TR1) and interrupt flag (TF1). Timer 1 can still be used for baud rate generation and the Timer 1 count values are still available in the TL1 and TH1 registers.

Control of Timer 1 when Timer 0 is in mode 3 is through the Timer 1 mode bits. To turn Timer 1 on, set Timer 1 to mode 0, 1, or 2. To turn

Timer 1 off, set it to mode 3. The Timer 1 C/$\overline{\text{T}}$ bit and T1M bit are still available to Timer 1. Therefore, Timer 1 can count CLK24/4, CLK24/12, or high-to-low transitions on the T1 pin. The Timer 1 GATE function is also available when Timer 0 is in mode 3.

Timer Rate Control

The default timer clock scheme for the 8051 timers is 12 CLK24 cycles per increment, the same as in the standard 8051. However, in the 8051, the instruction cycle is 4 CLK24 cycles.

Using the default rate (12 clocks per timer increment) allows existing application code with real-time dependencies, such as baud rate, to operate properly. However, applications that require fast timing can set the timers to increment every 4 CLK24 cycles by setting bits in the Clock Control register (CKCON) at SFR location 8Eh (see Table C-4).

The CKCON bits that control the timer clock rates are:

CKCON Bit	Counter/Timer
5	Timer 2
4	Timer 1
3	Timer 0

When a CKCON register bit is set to 1, the associated counter increments at 4-CLK24 intervals. When a CKCON bit is cleared, the associated counter increments at 12-CLK24 intervals. The timer controls are independent of each other. The default setting for all three timers is 0 (12-CLK24 intervals). These bits have no effect in counter mode.

Table C-4: CKCON Register - SRF 8Eh

Bit	Function
CKCON.7,6	Reserved
CKCON.5	T2M - Timer 2 clock select. When T2M = 0, Timer 2 uses CLK24/12 (for compatibility with 80C32); when T2M = 1, Timer 2 uses CLK24/4. This bit has no effect when Timer 2 is configured for baud rate generation.
CKCON.4	T1M - Timer 1 clock select. When T1M = 0, Timer 1 uses CLK24/12 (for compatibility with 80C32); when T1M = 1, Timer 1 uses CLK24/4.
CKCON.3	T0M - Timer 0 clock select. When T0M = 0, Timer 0 uses CLK24/12 (for compatibility with 80C32); when T0M = 1, Timer 0 uses CLK24/4.
CKCON.2-0	MD2, MD1, MD0 - Control the number of cycles to be used for external MOVX instructions.

Timer 2

Timer 2 runs only in 16-bit mode and offers several capabilities not available with Timers 0 and 1. The modes available with Timer 2 are:

- 16-bit timer/counter
- 16-bit timer with capture
- 16-bit auto-reload timer/counter
- Baud rate generator

The SFRs associated with Timer 2 are:

- T2CON - SFR C8h (Table C-6)
- RCAP2L - SFR CAh - Used to capture the TL2 value when Timer 2 is configured for capture mode, or as the LSB of the 16-bit reload value when Timer 2 is configured for auto-reload mode.

- RCAP2H - SFR CBh - Used to capture the TH2 value when Timer 2 is configured for capture mode, or as the MSB of the 16-bit reload value when Timer 2 is configured for auto-reload mode.
- TL2 - SFR CCh - Lower 8 bits of the 16-bit count.
- TH2 - SFR CDh - Upper 8 bits of the 16-bit count.

Timer 2 Mode Control

Table C-5 summarizes how the SFR bits determine the Timer 2 mode.

Table C-5: Timer 2 Mode Control Summary

RCLK	**TCLK**	**CP/$\overline{\text{RL}}$2**	**TR2**	**Mode**
0	0	1	1	16-bit timer/counter with capture
0	0	0	1	16-bit timer/counter with auto-reload
1	X	X	1	Baud rate generator
X	1	X	1	Baud rate generator
X	X	X	0	Off
X = Don’t care.				

16-Bit Timer/Counter Mode

Figure C-4 illustrates how Timer 2 operates in timer/counter mode with the optional capture feature. The C/$\overline{\text{T2}}$ bit determines whether the 16-bit counter counts CLK24 cycles (divided by 4 or 12), or high-to-low transitions on the T2 pin. The TR2 bit enables the counter. When the count increments from FFFFh, the TF2 flag is set, and the T2OUT pin goes high for one CLK24 cycle.

Table C-6: T2CON Register - SFR C8h

Bit	**Function**
T2CON.7	TF2 - Timer 2 overflow flag. Hardware will set TF2 when the Timer 2 overflows from FFFFh. TF2 must be cleared to 0 by the software. TF2 will only be set to a 1 if RCLK and TCLK are both cleared to 0. Writing a 1 to TF2 forces a Timer 2 interrupt if enabled.
T2CON.6	EXF2 - Timer 2 external flag. Hardware will set EXF2 when a reload or capture is caused by a high-to-low transition on the T2EX pin, and EXEN2 is set. EXF2 must be cleared to 0 by the software. Writing a 1 to EXF2 forces a Timer 2 interrupt if enabled.
T2CON.5	RCLK - Receive clock flag. Determines whether Timer 1 or Timer 2 is used for Serial Port 0 timing of received data in serial mode 1 or 3. RCLK =1 selects Timer 2 overflow as the receive clock. RCLK =0 selects Timer 1 overflow as the receive clock.
T2CON.4	TCLK - Transmit clock flag. Determines whether Timer 1 or Timer 2 is used for Serial Port 0 timing of transmit data in serial mode 1 or 3. RCLK =1 selects Timer 2 overflow as the transmit clock. RCLK =0 selects Timer 1 overflow as the transmit clock.
T2CON.3	EXEN2 - Timer 2 external enable. EXEN2 = 1 enables capture or reload to occur as a result of a high-to-low transition on the T2EX pin, if Timer 2 is not generating baud rates for the serial port. EXEN2 = 0 causes Timer 2 to ignore all external events on the T2EX pin.
T2CON.2	TR2 - Timer 2 run control flag. TR2 = 1 starts Timer 2. TR2 = 0 stops Timer 2.
T2CON.1	$C/\overline{T2}$ - Counter/timer select. $C/\overline{T2}$ = 0 selects a timer function for Timer 2. $C/\overline{T2}$ = 1 selects a counter of falling transitions on the T2 pin. When used as a timer, Timer 2 runs at 4 clocks per tick or 12 clocks per tick as programmed by CKCON.5, in all modes except baud rate generator mode. When used in baud rate generator mode, Timer 2 runs at 2 clocks per tick, independent of the state of CKCON.5.

Table C-6: T2CON Register - SFR C8h

Bit	Function
T2CON.0	CP/$\overline{\text{RL2}}$ - Capture/reload flag. When CP/$\overline{\text{RL2}}$ = 1, Timer 2 captures occur on high-to-low transitions of the T2EX pin, if EXEN2 = 1. When CP/$\overline{\text{RL2}}$ = 0, auto-reloads occur when Timer 2 overflows or when high-to-low transitions occur on the T2EX pin, if EXEN2 = 1. If either RCLK or TCLK is set to 1, CP/$\overline{\text{RL2}}$ will not function and Timer 2 will operate in auto-reload mode following each overflow.

16-Bit Timer/Counter Mode with Capture

The Timer 2 capture mode (Figure C-4) is the same as the 16-bit timer/counter mode, with the addition of the capture registers and control signals.

The CP/$\overline{\text{RL2}}$ bit in the T2CON SFR enables the capture feature. When CP/$\overline{\text{RL2}}$ = 1, a high-to-low transition on the T2EX pin when EXEN2 = 1 causes the Timer 2 value to be loaded into the capture registers RCAP2L and RCAP2H.

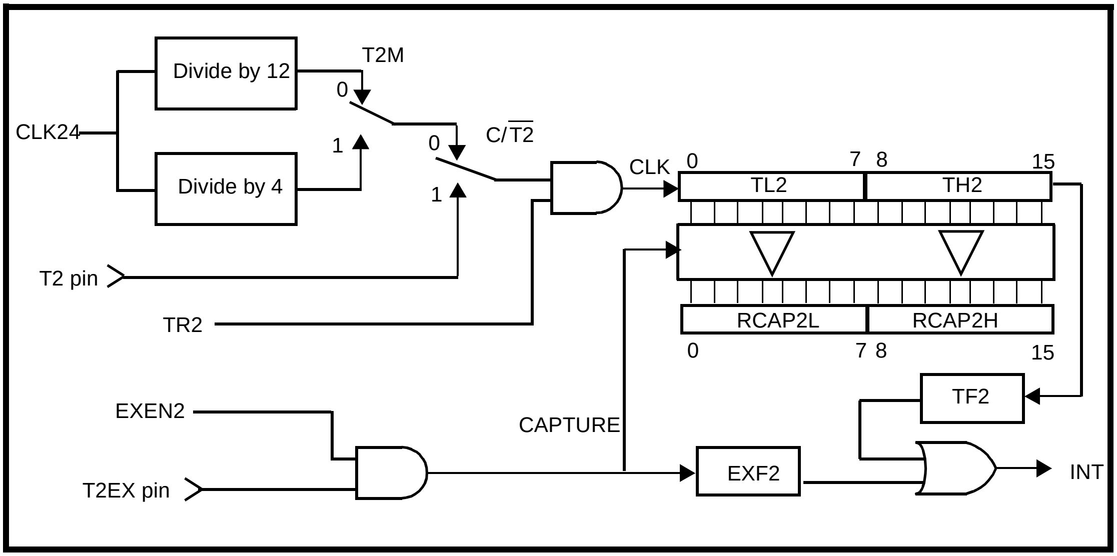

Figure C-4: Timer 2 - Timer/Counter with Capture

16-Bit Timer/Counter Mode with Auto-Reload

When CP/$\overline{\text{RL2}}$ = 0, Timer 2 is configured for the auto-reload mode illustrated in Figure C-5. Control of counter input is the same as for the other 16-bit counter modes. When the count increments from FFFFh, Timer 2 sets the TF2 flag and the starting value is reloaded into TL2 and TH2. The software must preload the starting value into the RCAP2L and RCAP2H registers.

When Timer 2 is in auto-reload mode, a reload can be forced by a high-to-low transition on the T2EX pin, if enabled by EXEN2 = 1.

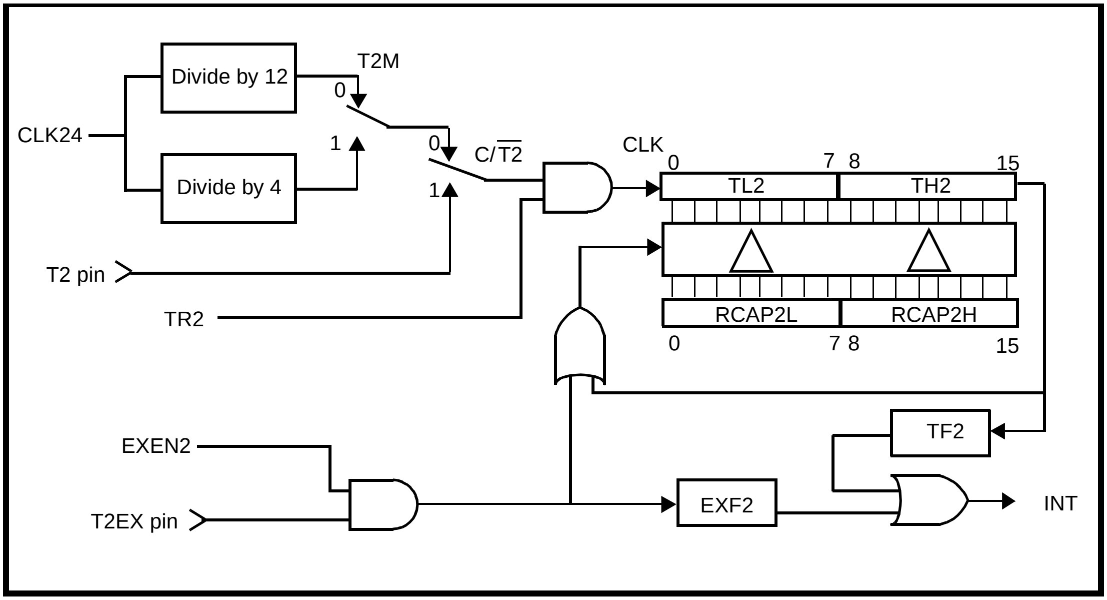

Figure C-5: Timer 2 - Timer/Counter with Auto Reload

Baud Rate Generator Mode

Setting either RCLK or TCLK to 1 configures Timer 2 to generate baud rates for Serial Port 0 in serial mode 1 or 3. In baud rate generator mode, Timer 2 functions in auto-reload mode. However, instead of setting the TF2 flag, the counter overflow is used to generate a shift clock for the serial port function. As in normal auto-reload mode, the overflow also causes the preloaded start value in the RCAP2L and RCAP2H registers to be reloaded into the TL2 and TH2 registers.

When either TCLK = 1 or RCLK = 1, Timer 2 is forced into auto-reload operation, regardless of the state of the CP/$\overline{\text{RL2}}$ bit.

When operating as a baud rate generator, Timer 2 does not set the TF2 bit. In this mode, a Timer 2 interrupt can only be generated by a high-to-low transition on the T2EX pin setting the EXF2 bit, and only if enabled by EXEN2 = 1.

The counter time base in baud rate generator mode is CLK24/2. To use an external clock source, set C/$\overline{\text{T2}}$ to 1 and apply the desired clock source to the T2 pin.

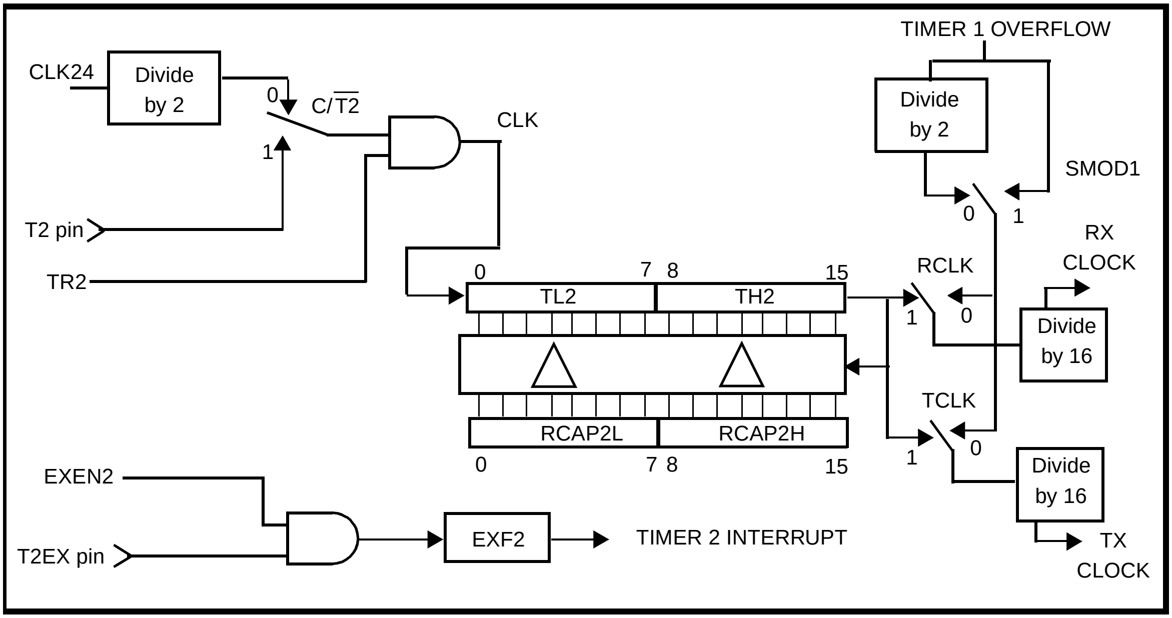

Figure C-6: Timer 2 - Baud Rate Generator Mode

Serial Interface

The 8051 core provides two serial ports. Serial Port 0 is identical in operation to the standard 8051 serial port. Serial Port 1 is identical to Serial Port 0, except that Timer 2 cannot be used as the baud rate generator for Serial Port 1.

Each serial port can operate in synchronous or asynchronous mode. In synchronous mode, 8051 generates the serial clock and the serial port operates in half-duplex mode. In asynchronous mode, the serial port operates in full-duplex mode. In all modes, 8051 buffers received data in a holding register, enabling the UART to receive an incoming byte before the software has read the previous value.

Each serial port can operate in one of four modes, as outlined in Table C-7.

Table C-7: Serial Port Modes

Mode	Sync/ Async	Baud Clock	Data Bits	Start/Stop	9th Bit Function
0	Sync	CLK24/4 or CLK24/12	8	None	None
1	Async	Timer 1 or Timer 2[1]	8	1 start, 1 stop	None
2	Async	CLK24/32 or CLK24/64	9	1 start, 1 stop	0, 1, parity
3	Async	Timer 1 or Timer 2[1]	9	1 start, 1 stop	0, 1, parity
(1) Timer 2 available for Serial Port 0 only.					

The SFRs associated with the serial ports are:

- SCON0 - SFR 98h - Serial Port 0 control (Table C-8).
- SBUF0 - SFR 99h - Serial Port 0 buffer.
- SCON1 - SFR C0h - Serial Port 1 control (Table C-9).
- SBUF1 - SFR C1h - Serial Port 1 buffer.

803x/805x Compatibility

The implementation of the serial interface is similar to that of the Intel 8052.

Mode 0

Serial mode 0 provides synchronous, half-duplex serial communication. For Serial Port 0, serial data output occurs on the RXD0OUT pin, serial data is received on the RXD0 pin, and the TXD0 pin provides the shift clock for both transmit and receive. For Serial Port 1, the corresponding pins are RXD1OUT, RXD1, and TXD1.

The serial mode 0 baud rate is either CLK24/12 or CLK24/4, depending on the state of the SM2_0 bit (or SM2_1 for Serial Port 1). When SM2_0 = 0, the baud rate is CLK24/12, when SM2_0 = 1, the baud rate is CLK24/4.

Mode 0 operation is identical to the standard 8051. Data transmission begins when an instruction writes to the SBUF0 (or SBUF1) SFR. The UART shifts the data, LSB first, at the selected baud rate, until the 8-bit value has been shifted out.

Mode 0 data reception begins when the REN_0 (or REN_1) bit is set and the RI_0 (or RI_1) bit is cleared in the corresponding SCON SFR. The shift clock is activated and the UART shifts data in on each rising edge of the shift clock until 8 bits have been received. One machine cycle after the 8th bit is shifted in, the RI_0 (or RI_1) bit is set and reception stops until the software clears the RI bit.

Figure C-7 through Figure C-10 illustrate Serial Port Mode 0 transmit and receive timing for both low-speed (CLK24/12) and high-speed (CLK24/4) operation.

Table C-8: SCON0 Register - SFR 98h

Bit	Function
SCON0.7	SM0_0 - Serial Port 0 mode bit 0.
SCON0.6	SM1_0 - Serial Port 0 mode bit 1, decoded as: **SM0_0 SM1_0 Mode** 0 0 0 0 1 1 1 0 2 1 1 3
SCON0.5	SM2_0 - Multiprocessor communication enable. In modes 2 and 3, this bit enables the multiprocessor communication feature. If SM2_0 = 1 in mode 2 or 3, then RI_0 will not be activated if the received 9th bit is 0. If SM2_0=1 in mode 1, then RI_0 will only be activated if a valid stop is received. In mode 0, SM2_0 establishes the baud rate: when SM2_0=0, the baud rate is CLK24/12; when SM2_0=1, the baud rate is CLK24/4.
SCON0.4	REN_0 - Receive enable. When REN_0=1, reception is enabled.
SCON0.3	TB8_0 - Defines the state of the 9th data bit transmitted in modes 2 and 3.
SCON0.2	RB8_0 - In modes 2 and 3, RB8_0 indicates the state of the 9th bit received. In mode 1, RB8_0 indicates the state of the received stop bit. In mode 0, RB8_0 is not used.
SCON0.1	TI_0 - Transmit interrupt flag. indicates that the transmit data word has been shifted out. In mode 0, TI_0 is set at the end of the 8th data bit. In all other modes, TI_0 is set when the stop bit is placed on the TXD0 pin. **TI_0 must be cleared by firmware.**
SCON0.0	RI_0 - Receive interrupt flag. Indicates that serial data word has been received. In mode 0, RI_0 is set at the end of the 8th data bit. In mode 1, RI_0 is set after the last sample of the incoming stop bit, subject to the state of SM2_0. In modes 2 and 3, RI_0 is set at the end of the last sample of RB8_0. **RI_0 must be cleared by firmware.**

Table C-9: SCON1 Register - SFR C0h

Bit	Function
SCON1.7	SM0_1 - Serial Port 1 mode bit 0.
SCON1.6	SM1_1 - Serial Port 1 mode bit 1, decoded as: **SM0_1** **SM1_1** **Mode** 0 0 0 0 1 1 1 0 2 1 1 3
SCON1.5	SM2_1 - Multiprocessor communication enable. In modes 2 and 3, this bit enables the multiprocessor communication feature. If SM2_1 = 1 in mode 2 or 3, then RI_1 will not be activated if the received 9th bit is 0. If SM2_1=1 in mode 1, then RI_1 will only be activated if a valid stop is received. In mode 0, SM2_1 establishes the baud rate: when SM2_1=0, the baud rate is CLK24/12; when SM2_1=1, the baud rate is CLK24/4.
SCON1.4	REN_1 - Receive enable. When REN_1=1, reception is enabled.
SCON1.3	TB8_1 - Defines the state of the 9th data bit transmitted in modes 2 and 3.
SCON1.2	RB8_1 - In modes 2 and 3, RB8_0 indicates the state of the 9th bit received. In mode 1, RB8_1 indicates the state of the received stop bit. In mode 0, RB8_1 is not used.
SCON1.1	TI_1 - Transmit interrupt flag. indicates that the transmit data word has been shifted out. In mode 0, TI_1 is set at the end of the 8th data bit. In all other modes, TI_1 is set when the stop bit is placed on the TXD0 pin. TI_1 must be cleared by the software.
SCON1.0	RI_1 - Receive interrupt flag. Indicates that serial data word has been received. In mode 0, RI_1 is set at the end of the 8th data bit. In mode 1, RI_1 is set after the last sample of the incoming stop bit, subject to the state of SM2_1. In modes 2 and 3, RI_1 is set at the end of the last sample of RB8_1. RI_1 must be cleared by the software.

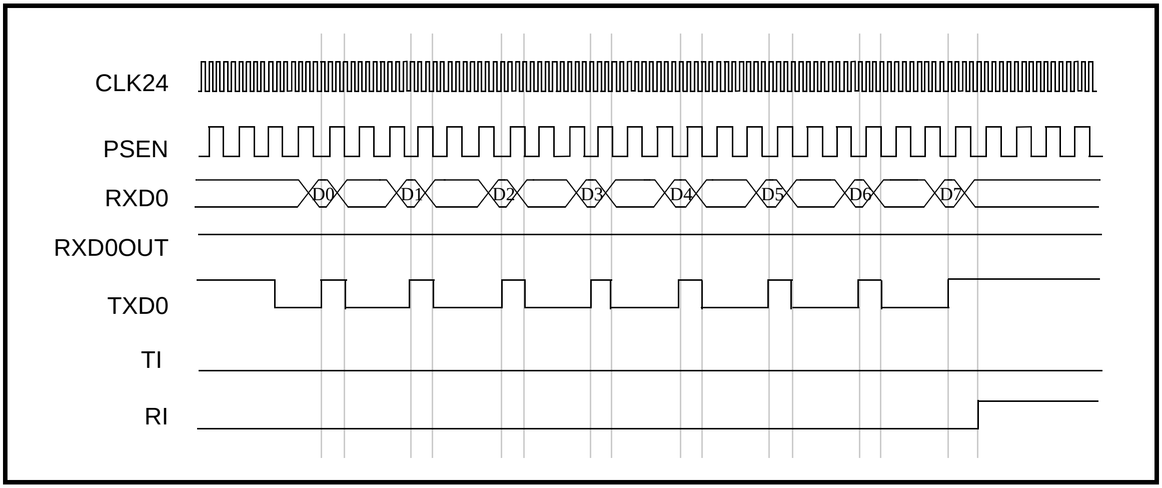

Figure C-7: Serial Port Mode 0 Receive Timing - Low Speed Operation

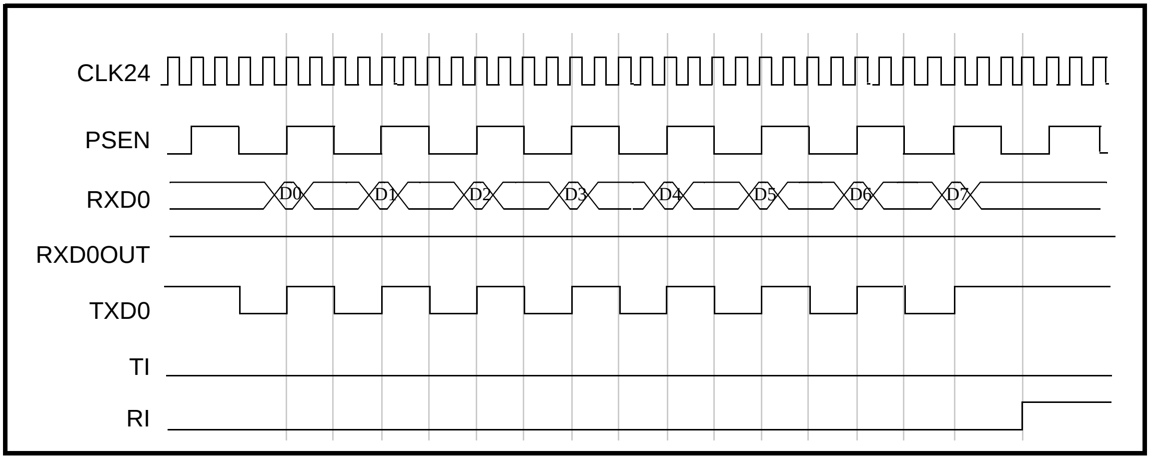

Figure C-8: Serial Port Mode 0 Receive Timing - High Speed Operation

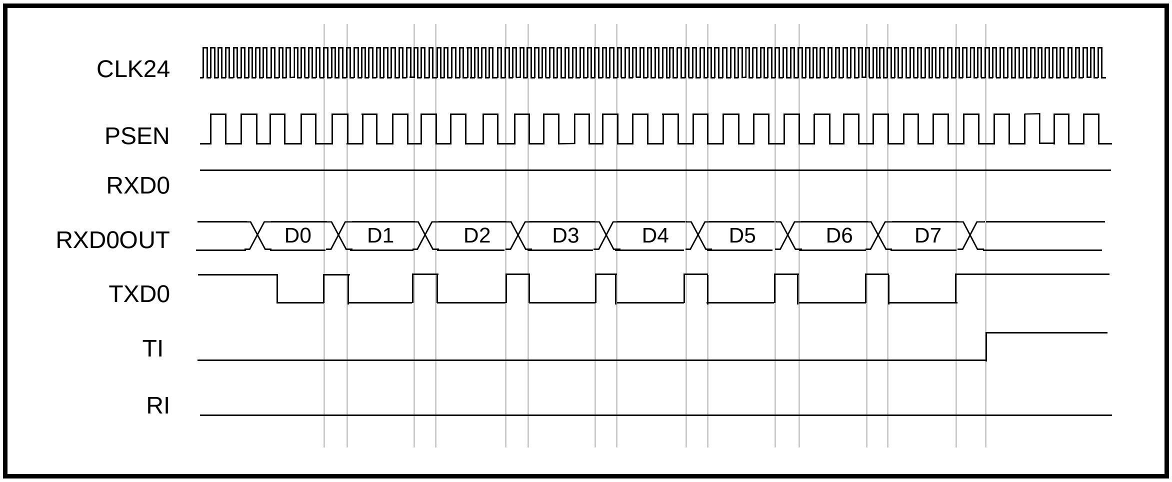

Figure C-9: Serial Port Mode 0 Transmit Timing - Low Speed Operation

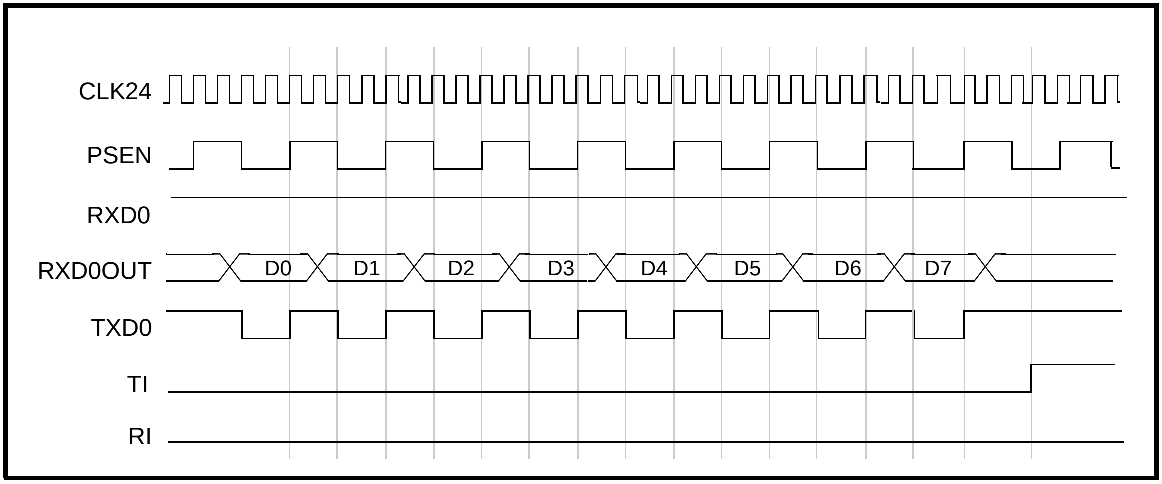

Figure C-10: Serial Port Mode 0 Transmit Timing - High Speed Operation

Mode 1

Mode 1 provides standard asynchronous, full-duplex communication, using a total of 10 bits: 1 start bit, 8 data bits, and 1 stop bit. For receive operations, the stop bit is stored in RB8_0 (or RB8_1). Data bits are received and transmitted LSB first.

Mode 1 Baud Rate

The mode 1 baud rate is a function of timer overflow. Serial Port 0 can use either Timer 1 or Timer 2 to generate baud rates. Serial Port 1 can only use Timer 1. The two serial ports can run at the same baud rate if they both use Timer 1, or different baud rates if Serial Port 0 uses Timer 2 and Serial Port 1 uses Timer 1.

Each time the timer increments from its maximum count (FFh for Timer 1 or FFFFh for Timer 2), a clock is sent to the baud rate circuit. The clock is then divided by 16 to generate the baud rate.

When using Timer 1, the SMOD0 (or SMOD1) bit selects whether or not to divide the Timer 1 rollover rate by 2. Therefore, when using Timer 1, the baud rate is determined by the equation:

$$\text{Baud Rate} = \frac{2^{\text{SMODx}}}{32} \times \text{Timer 1 Overflow}$$

SMOD0 is SFR bit PCON.7; SMOD1 is SFR bit EICON.7.

When using Timer 2, the baud rate is determined by the equation:

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow}}{16}$$

To use Timer 1 as the baud rate generator, it is best to use Timer 1 mode 2 (8-bit counter with auto-reload), although any counter mode can be used. The Timer 1 reload is stored in the TH1 register, which makes the complete formula for Timer 1:

$$\text{Baud Rate} = \frac{2^{\text{SMODx}}}{32} \times \frac{\text{CLK24}}{12 \times (256 - \text{TH1})}$$

The 12 in the denominator in the above equation can be changed to 4 by setting the T1M bit in the CKCON SFR. To derive the required TH1 value from a known baud rate (when TM1 = 0), use the equation:

$$\text{TH1} = 256 - \frac{2^{\text{SMODx}} \times \text{CLK24}}{384 \times \text{Baud Rate}}$$

You can also achieve very low serial port baud rates from Timer 1 by enabling the Timer 1 interrupt, configuring Timer 1 to mode 1, and using the Timer 1 interrupt to initiate a 16-bit software reload. Table C-10lists sample reload values for a variety of common serial port baud rates.

Note that more accurate baud rates are achieved by using Timer 2 as the baud rate generator (next section).

Table C-10: Timer 1 Reload Values for Common Serial Port Mode 1 Baud Rates

<table>
<tr><th>Nominal Rate</th><th>24 MHz Divisor</th><th>Reload Value</th><th>Actual Rate</th><th>Error</th></tr>
<tr><td>57600</td><td>6</td><td>FA</td><td>62500</td><td>8.5%</td></tr>
<tr><td>38400</td><td>10</td><td>F6</td><td>37500</td><td>-2.3%</td></tr>
<tr><td>28800</td><td>13</td><td>F3</td><td>28846</td><td>+0.16%</td></tr>
<tr><td>19200</td><td>20</td><td>EC</td><td>18750</td><td>-2.3%</td></tr>
<tr><td>9600</td><td>39</td><td>D9</td><td>9615</td><td>+0.16%</td></tr>
<tr><td>4800</td><td>78</td><td>B2</td><td>4807</td><td>+0.15%</td></tr>
<tr><td>2400</td><td>156</td><td>64</td><td>2403</td><td>+.13%</td></tr>
<tr><td colspan="5">Settings: SMOD =1, C/$\overline{\text{T}}$=0, Timer1 mode=2, TIM=1

Note: Using rates that are off by 2.3% or more will not work in all systems.</td></tr>
</table>

To use Timer 2 as the baud rate generator, configure Timer 2 in auto-reload mode and set the TCLK and/or RCLK bits in the T2CON SFR. TCLK selects Timer 2 as the baud rate generator for the transmitter; RCLK selects Timer 2 as the baud rate generator for the receiver. The 16-bit reload value for Timer 2 is stored in the RCAP2L and RCA2H SFRs, which makes the equation for the Timer 2 baud rate:

$$\text{Baud Rate} = \frac{\text{CLK24}}{32 \times (65536 - \text{RCAP2H,RCAP2L})}$$

where RCAP2H,RCAP2L is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned number.

The 32 in the denominator is the result of CLK24 being divided by 2 and the Timer 2 overflow being divided by 16. Setting TCLK or RCLK to 1 automatically causes CLK24 to be divided by 2, as shown in Figure C-6, instead of the 4 or 12 determined by the T2M bit in the CKCON SFR.

To derive the required RCAP2H and RCAP2L values from a known baud rate, use the equation:

$$\text{RCAP2H,RCAP2L} = 65536 - \frac{\text{CLK24}}{32 \times \text{Baud Rate}}$$

When either RCLK or TCLK is set, the TF2 flag will not be set on a Timer 2 roll over, and the T2EX reload trigger is disabled.

Table C-11: Timer 2 Reload Values for Common Serial port Mode 1 Baud Rates

Nominal Rate	C/T2	RCLK/ TCLK	Divisor	Reload Val	Actual Rate	Error
57600	0	1	13	FFF3	57692.31	0.16%
38400	0	1	20	FFEC	37500	-2.34%
28800	0	1	26	FFE6	28846.15	0.16%
19200	0	1	39	FFD9	19230.77	0.16%
9600	0	1	78	FFB2	9615.385	0.16%
4800	0	1	156	FF64	4807.692	0.16%
2400	0	1	312	FEC8	2403.846	0.16%
Note: using rates that are off by 2.3% or more will not work in all systems.						

Mode 1 Transmit

Figure C-11 illustrates the mode 1 transmit timing. In mode 1, the UART begins transmitting after the first roll over of the divide-by-16 counter after the software writes to the SBUF0 (or SBUF1) register. The UART transmits data on the TXD0 (or TXD1) pin in the following order: start bit, 8 data bits (LSB first), stop bit. The TI_0 (or TI_1) bit is set 2 CLK24 cycles after the stop bit is transmitted.

Mode 1 Receive

Figure C-12 illustrates the mode 1 receive timing. Reception begins at the falling edge of a start bit received on the RXD0 (or RXD1) pin, when enabled by the REN_0 (or REN_1) bit. For this purpose, the RXD0 (or RXD1) pin is sampled 16 times per bit for any baud rate. When a falling edge of a start bit is detected, the divide-by-16 counter used to generate the receive clock is reset to align the counter roll over to the bit boundaries.

If the above conditions are met, the serial port then writes the received byte to the SBUF0 (or SBUF1) register, loads the 9th received bit into RB8_0 (or RB8_1), and sets the RI_0 (or RI_1) bit. If the above conditions are not met, the received data is lost, the SBUF register and RB8 bit are not loaded, and the RI bit is not set. After the middle of the stop bit time, the serial port waits for another high-to-low transition on the RXD0 or RXD1 pin.

At the middle of the stop bit time, the serial port checks for the following conditions:

- RI_0 (or RI_1) = 0, and
- If SM2_0 (or SM2_1) = 1, the state of the stop bit is 1.
  (If SM2_0 (or SM2_1) = 0, the state of the stop bit doesn’t matter.)

If the above conditions are met, the serial port then writes the received byte to the SBUF0 (or SBUF1) register, loads the stop bit into RB8_0 (or RB8_1), and sets the RI_0 (or RI_1) bit. If the above conditions are not met, the received data is lost, the SBUF register and RB8 bit are not loaded, and the RI bit is not set.

After the middle of the stop bit time, the serial port waits for another high-to-low transition on the (RXD0 or RXD1) pin.

Mode 1 operation is identical to that of the standard 8051 when Timers 1 and 2 use CLK24/12 (the default).

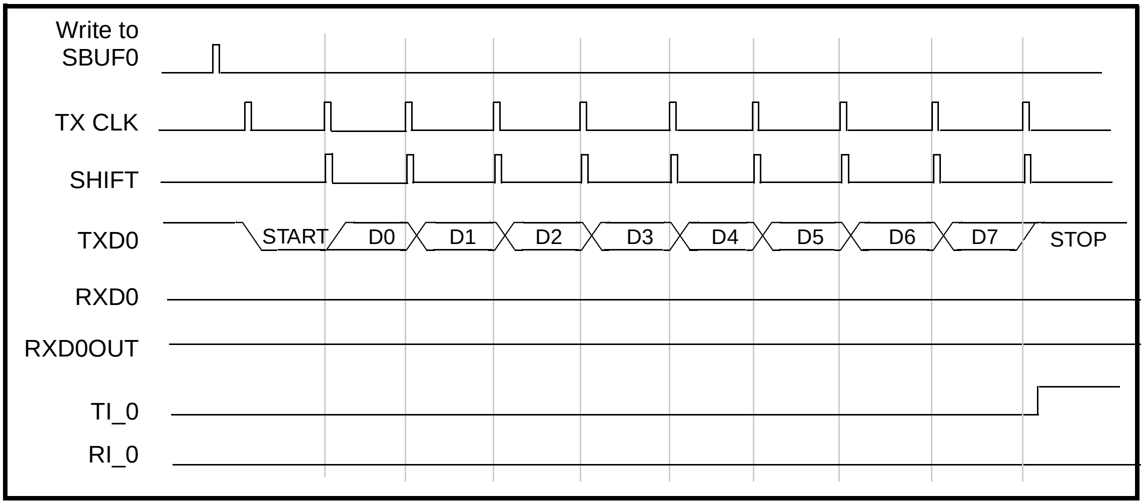

Figure C-11: Serial Port 0 Mode 1 Transmit Timing

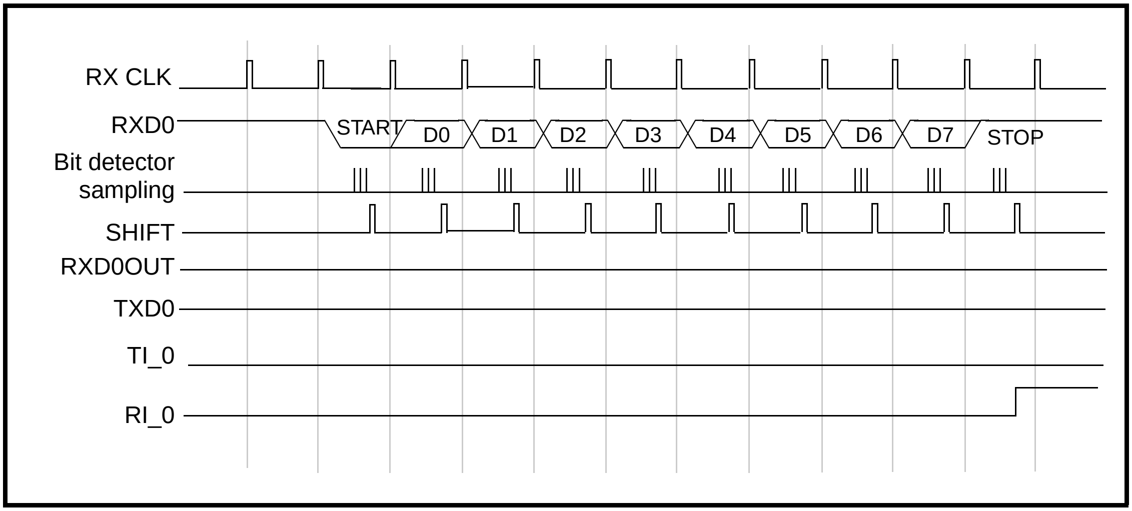

Figure C-12: Serial Port 0 Mode 1 Receive Timing

Mode 2

Mode 2 provides asynchronous, full-duplex communication, using a total of 11 bits: 1 start bit, 8 data bits, a programmable 9th bit, and 1 stop bit. The data bits are transmitted and received LSB first. For transmission, the 9th bit is determined by the value in TB8_0 (or TB8_1). To use the 9th bit as a parity bit, move the value of the P bit (SFR PSW.0) to TB8_0 (or TB8_1).

The mode 2 baud rate is either CLK24/32 or CLK24/64, as determined by the SMOD0 (or SMOD1) bit. The formula for the mode 2 baud rate is:

$$\text{Baud Rate} = \frac{2^{\text{SMODx}} \times \text{CLK24}}{64}$$

Mode 2 operation is identical to the standard 8051.

Mode 2 Transmit

Figure C-13 illustrates the mode 2 transmit timing. Transmission begins after the first roll over of the divide-by-16 counter following a software write to SBUF0 (or SBUF1). The UART shifts data out on the TXD0 (or TXD1) pin in the following order: start bit, data bits (LSB first), 9th bit, stop bit. The TI_0 (or TI_1) bit is set when the stop bit is placed on the TXD0 (or TXD1) pin.

Mode 2 Receive

Figure C-14 illustrates the mode 2 receive timing. Reception begins at the falling edge of a start bit received on the RXD0 (or RXD1) pin, when enabled by the REN_0 (or REN_1) bit. For this purpose, the RXD0 (or RXD1) pin is sampled 16 times per bit for any baud rate. When a falling edge of a start bit is detected, the divide-by-16 counter used to generate the receive clock is reset to align the counter roll over to the bit boundaries.

For noise rejection, the serial port establishes the content of each received bit by a majority decision of 3 consecutive samples in the middle of each bit time. This is especially true for the start bit. If the falling edge on the RXD0 (or RXD1) pin is not verified by a majority decision of 3 consecutive samples (low), then the serial port stops reception and waits for another falling edge on the RXD0 (or RXD1) pin.

At the middle of the stop bit time, the serial port checks for the following conditions:

- RI_0 (or RI_1) = 0, and
- If SM2_0 (or SM2_1) = 1, the state of the stop bit is 1.
  (If SM2_0 (or SM2_1) = 0, the state of the stop bit doesn't matter.)

If the above conditions are met, the serial port then writes the received byte to the SBUF0 (or SBUF1) register, loads the stop bit into RB8_0 (or RB8_1), and sets the RI_0 (or RI_1) bit. If the above conditions are not met, the received data is lost, the SBUF register and RB8 bit are not loaded, and the RI bit is not set. After the middle of the stop bit time, the serial port waits for another high-to-low transition on the RXD0 *(*or RXD1) pin.

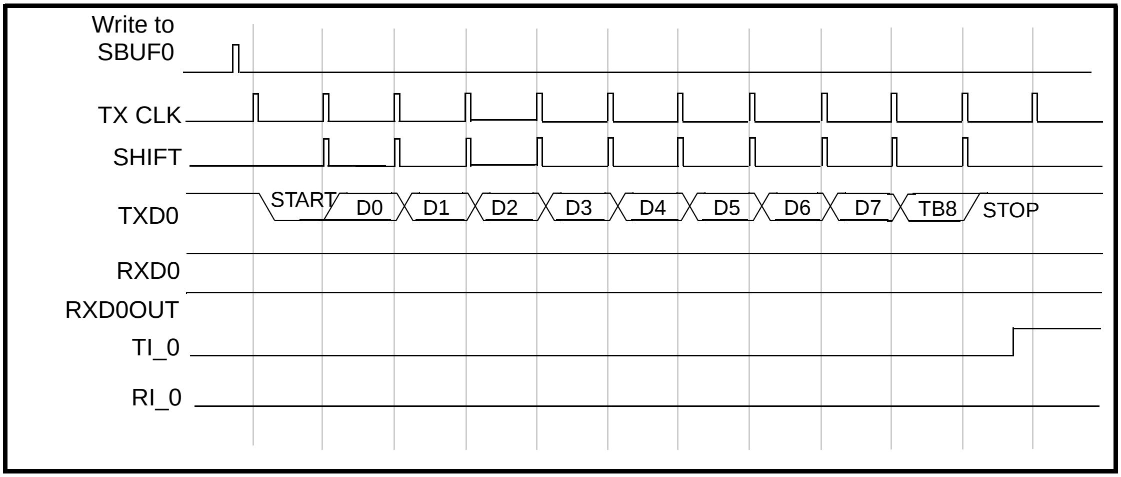

Figure C-13: Serial Port 0 Mode 2 Transmit Timing

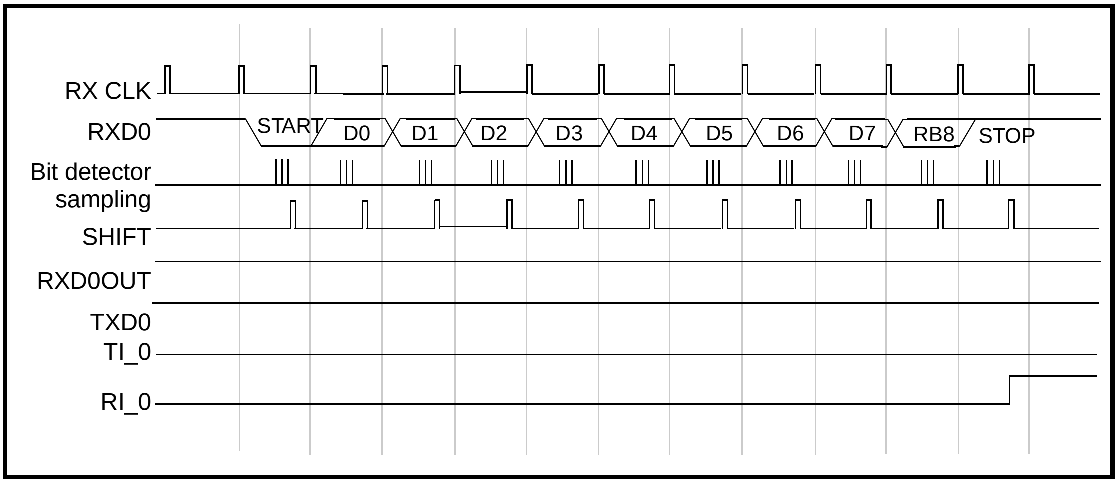

Figure C-14: Serial Port 0 Mode 2 Receive Timing

Mode 3

Mode 3 provides asynchronous, full-duplex communication, using a total of 11 bits: 1 start bit, 8 data bits, a programmable 9th bit, and 1 stop bit. The data bits are transmitted and received LSB first.

The mode 3 transmit and operations are identical to mode 2. The mode 3 baud rate generation is identical to mode 1. That is, mode 3 is a combination of mode 2 protocol and mode 1 baud rate. Figure C-15 illustrates the mode 3 transmit timing. Figure C-16 illustrates the mode 3 receive timing.

Mode 3 operation is identical to that of the standard 8051 when Timers 1 and 2 use CLK24/12 (the default).

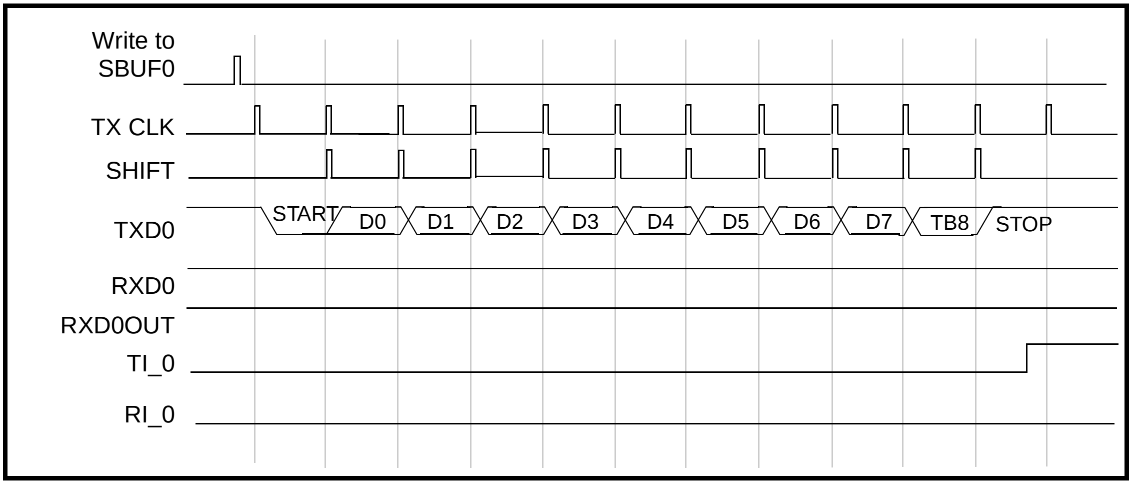

Figure C-15: Serial Port 0 Mode 3 Transmit Timing

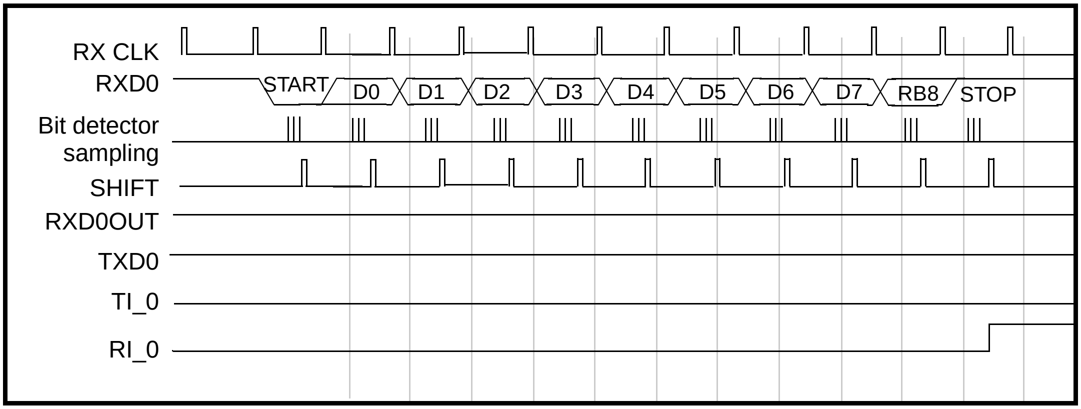

Figure C-16: Serial Port 0 Mode 3 Receive Timing

Multiprocessor Communications

The multiprocessor communication feature is enabled in modes 2 and 3 when the SM2 bit is set in the SCON SFR for a serial port (SM2_0 for Serial Port 0, SM2_1 for Serial Port 1). In multiprocessor communication mode, the 9th bit received is stored in RB8_0 (or RB8_1) and, after the stop bit is received, the serial port interrupt is activated only if RB8_0 (or RB8_1) = 1.

A typical use for the multiprocessor communication feature is when a master wants to send a block of data to one of several slaves. The master first transmits an address byte that identifies the target slave. When transmitting an address byte, the master sets the 9th bit to 1; for data bytes, the 9th bit is 0.

With SM2_0 (or SM2_1) = 1, no slave will be interrupted by a data byte. However, an address byte interrupts all slaves so that each slave can examine the received address byte to determine whether that slave is being addressed. Address decoding must be done by software during the interrupt service routine. The addressed slave clears its SM2_0 (or SM2_1) bit and prepares to receive the data bytes. The slaves that are not being addressed leave the SM2_0 (or SM2_1) bit set and ignore the incoming data bytes.

Interrupt SFRs

The following SFRs are associated with interrupt control:

- IE - SFR A8h (Table C-12)
- IP - SFR B8h (Table C-13)
- EXIF - SFR 91h (Table C-14)
- EICON - SFR D8h (Table C-15)
- EIE - SFR E8h (Table C-16)
- EIP - SFR F8h (Table C-17)

The IE and IP SFRs provide interrupt enable and priority control for the standard interrupt unit, as with the standard 8051. Additionally, these SFRs provide control bits for the Serial Port 1 interrupt. These bits (ES1 and PS1) are available only when the extended interrupt unit is implemented (ext_intr=1). Otherwise, they are read as 0.

Bits ES0, ES1, ET2, PS0, PS1, and PT2 are present, but not used, when the corresponding module is not implemented.

The EXIF, EICON, EIE and EIP registers provide flags, enable control, and priority control for the optional extended interrupt unit.

Table C-12: IE Register - SFR A8h

Bit	Function
IE.7	EA - Global interrupt enable. Controls masking of all interrupts except USB wakeup (resume). EA = 0 disables all interrupts except USB wakeup. When EA = 1, interrupts are enabled or masked by their individual enable bits.
IE.6	ES1 - Enable Serial Port 1 interrupt. ES1 = 0 disables Serial port 1 interrupts (TI_1 and RI_1). ES1 = 1 enables interrupts generated by the TI_1 or TI_1 flag.
IE.5	ET2 - Enable Timer 2 interrupt. ET2 = 0 disables Timer 2 interrupt (TF2). ET2=1 enables interrupts generated by the TF2 or EXF2 flag.
IE.4	ES0 - Enable Serial Port 0 interrupt. ES0 = 0 disables Serial Port 0 interrupts (TI_0 and RI_0). ES0=1 enables interrupts generated by the TI_0 or RI_0 flag.
IE.3	ET1 - Enable Timer 1 interrupt. ET1 = 0 disables Timer 1 interrupt (TF1). ET1=1 enables interrupts generated by the TF1 flag.
IE.2	EX1 - Enable external interrupt 1. EX1 = 0 disables external interrupt 1 (INT1). EX1=1 enables interrupts generated by the INT1# pin.
IE.1	ET0 - Enable Timer 0 interrupt. ET0 = 0 disables Timer 0 interrupt (TF0). ET0=1 enables interrupts generated by the TF0 flag.
IE.0	EX0 - Enable external interrupt 0. EX0 = 0 disables external interrupt 0 (INT0). EX0=1 enables interrupts generated by the INT0# pin.

Table C-13: IP Register - SFR B8h

Bit	Function
IP.7	Reserved. Read as 1.
IP.6	PS1 - Serial Port 1 interrupt priority control. PS1=0 sets Serial Port 1 interrupt (TI_1 or RI_1) to low priority. PS1=1 sets Serial port 1 interrupt to high priority.
IP.5	PT2 - Timer 2 interrupt priority control. PT2=0 sets Timer 2 interrupt (TF2) to low priority. PT2=1 sets Timer 2 interrupt to high priority.
IP.4	PS0 - Serial Port 0 interrupt priority control. PS0=0 sets Serial Port 0 interrupt (TI_0 or RI_0) to low priority. PS0=1 sets Serial Port 0 interrupt to high priority.
IP.3	PT2 - Timer 1 interrupt priority control. PT1 = 0 sets Timer 1 interrupt (TF1) to low priority. PT1=1 sets Timer 1 interrupt to high priority.
IP.2	PX1 - External interrupt 1 priority control. PX 1= 0 sets external interrupt 1 (INT1) to low priority. PT1 = 1 sets external interrupt 1 to high priority.
IP.1	PT0 - Timer 0 interrupt priority control. PT0 = 0 sets Timer 0 interrupt (TF0) to low priority. PT0=1 sets Timer 0 interrupt to high priority.
IP.0	PX0 - External interrupt 0 priority control. PX0 = 0 sets external interrupt 0 (INT0) to low priority. PX0=1 sets external interrupt 0 to high priority.

Table C-14: EXIF Register - SFR 91h

Bit	Function
EXIF.7	IE5 - External interrupt 5 flag. IE 5= 1 indicates a falling edge was detected at the INT5# pin. IE5 must be cleared by software. Setting IE5 in software generates an interrupt, if enabled.
EXIF.6	IE4 - External interrupt 4 flag. IE4 indicates a rising edge was detected at the INT4 pin. IE4 must be cleared by software. Setting IE4 in software generates an interrupt, if enabled.
EXIF.5	I2CINT - External interrupt 3 flag. The "INT3" interrupt is internally connected to the EZ-USB I^2C controller and renamed "I2CINT". I2CINT = 1 indicates an I^2C interrupt. I2CINT must be cleared by software. Setting I2CINT in software generates an interrupt, if enabled.
EXIF.4	USBINT - External interrupt 2 flag. The "INT2" interrupt is internally connected to the EZ-USB interrupt and renamed "USBINT". USBINT = 1 indicates an USB interrupt. USBINT must be cleared by software. Setting USBINT in software generates an interrupt, if enabled.
EXIF.3	Reserved. Read as 1.
EXIF.2-0	Reserved. Read as 0.

Table C-15: EICON Register - SFR D8h

Bit	Function
EICON.7	SMOD1 - Serial Port 1 baud rate doubler enable. When SMOD1 = 1 the baud rate for Serial Port is doubled.
EICON.6	Reserved. Read as 1.
EICON.5	ERESI - Enable resume interrupt. ERESI = 0 disables resume interrupt (RESI). ERESI = 1 enables interrupts generated by the resume event.
EICON.4	RESI - Wakeup interrupt flag. EICON.4 = 1 indicates a negative transition was detected at the WAKEUP# pin, or that USB has activity resumed from the suspended state. EICON.4 = 1 must be cleared by software before exiting the interrupt service routine, otherwise the interrupt occurs again. Setting EICON.4=1 in software generates a wakeup interrupt, if enabled.
EICON.3	INT6 - External interrupt 6. When INT6 = 1, the INT6 pin has detected a low to high transition. INT6 will remain active until cleared by writing a 0 to this bit. Setting this bit in software generates an INT6 interrupt in enabled.
EICON.2-0	Reserved. Read as 0.

Table C-16: EIE Register - SFR E8h

Bit	Function
EIE.7-5	Reserved. Read as 1.
EIE.4	EX6 - Enable external interrupt 6. EX6 = 0 disables external interrupt 6 (INT6). EX6 = 1 enables interrupts generated by the INT6 pin.
EIE.3	EX5 - Enable external interrupt 5. EX5 = 0 disables external interrupt 5 (INT5). EX5 = 1 enables interrupts generated by the INT5# pin.
EIE.2	EX4 - Enable external interrupt 4. EX4 = 0 disables external interrupt 4 (INT4). EX4 = 1 enables interrupts generated by the INT4 pin.
EIE.1	EI2C - Enable external interrupt 3. EI2C = 0 disables external interrupt 3 (INT3). EI2C = 1 enables interrupts generated by the I^{2}C interface.
EIE.0	EUSB - Enable USB interrupt. EUSB = 0 disables USB interrupts. EUSB = 1 enables interrupts generated by the USB Interface.

Table C-17: EIP Register - SFR F8h

Bit	Function
EIP.7-5	Reserved. Read as 1.
EIP.4	PX6 - External interrupt 6 priority control. PX6 = 0 sets external interrupt 6 (INT6) to low priority. PX6 = 1 sets external interrupt 6 to high priority.
EIP.3	PX5 - External interrupt 5 priority control. PX5 = 0 sets external interrupt 5 (INT5#) to low priority. PX5=1 sets external interrupt 5 to high priority.
EIP.2	PX4 - External interrupt 4 priority control. PX4 = 0 sets external interrupt 4 (INT4) to low priority. PX4=1 sets external interrupt 4 to high priority.
EIP.1	PI2C - External interrupt 3 priority control. PI2C = 0 sets I^2C interrupt to low priority. PI2C=1 sets I^2C interrupt to high priority.
EIP.0	PUSB - External interrupt 2 priority control. PUSB = 0 sets USB interrupt to low priority. PUSB=1 sets USB interrupt to high priority.

Interrupt Processing

When an enabled interrupt occurs, the 8051 core vectors to the address of the interrupt service routine (ISR) associated with that interrupt, as listed in Table C-18. The 8051 core executes the ISR to completion unless another interrupt of higher priority occurs. Each ISR ends with a `RETI` (return from interrupt) instruction. After executing the `RETI`, the CPU returns to the next instruction that would have been executed if the interrupt had not occurred.

An ISR can only be interrupted by a higher priority interrupt. That is, an ISR for a low-level interrupt can only be interrupted by high-level interrupt. An ISR for a high-level interrupt can only be interrupted by the resume interrupt.

The 8051 core always completes the instruction in progress before servicing an interrupt. If the instruction in progress is `RETI`, or a write access to any of the IP, IE, EIP, or EIE SFRs, the 8051 core completes one additional instruction before servicing the interrupt.

Interrupt Masking

The EA bit in the IE SFR (IE.7) is a global enable for all interrupts except the USB wakeup (resume) interrupt. When EA = 1, each interrupt is enabled or masked by its individual enable bit. When EA = 0, all interrupts are masked, except the USB wakeup interrupt.

Table C-19 provides a summary of interrupt sources, flags, enables, and priorities.

Table C-18: Interrupt Natural Vectors and Priorities

Interrupt	**Description**	**Natural Priority**	**Interrupt Vector**
RESUME	USB Wakeup (resume) interrupt	0	33h
INT0	External interrupt 0	1	03h
TF0	Timer 0 interrupt	2	0Bh
INT1	External interrupt 1	3	13h
TF1	Timer 1 interrupt	4	1Bh
TI_0 or RI_0	Serial port 0 interrupt	5	23h
TF2 or EXF2	Timer 2 interrupt	6	2Bh
TI_1 or RI_1	Serial port 1 interrupt	7	3Bh
INT2	USB interrupt	8	43h
INT3	I^2C interrupt	9	4Bh
INT4	External interrupt 4	4	53h
INT5	External interrupt 5	11	5Bh
INT6	External interrupt 6	12	63H

Interrupt Priorities

There are two stages of interrupt priority assignment, interrupt level and natural priority. The interrupt level (highest, high, or low) takes precedence over natural priority. The USB wakeup interrupt, if enabled, always has highest priority and is the only interrupt that can have highest priority. All other interrupts can be assigned either high or low priority.

In addition to an assigned priority level (high or low), each interrupt also has a natural priority, as listed in Table C-18. Simultaneous interrupts with the same priority level (for example, both high) are resolved according to their natural priority. For example, if INT0 and INT2 are both programmed as high priority, INT0 takes precedence due to its higher natural priority.

Once an interrupt is being serviced, only an interrupt of higher priority level can interrupt the service routine of the interrupt currently being serviced.

Table C-19: Interrupt Flags, Enables, and Priority Control

Interrupt	**Description**	**Flag**	**Enable**	**Priority Control**
RESUME	Resume interrupt	EICON.4	EICON.5	N/A
INT0	External interrupt 0	TCON.1	IE.0	IP.0
TF0	Timer 0 interrupt	TCON.5	IE.1	IP.1
INT1	External interrupt 1	TCON.3	IE.2	IP.2
TF1	Timer 1 interrupt	TCON.7	IE.3	IP.3
TI_0 or RI_0	Serial port 0 transmit or receive	SCON0.0 (RI.0), SCON0.1 (Ti_0)	IE.4	IP.4
TF2 or EXF2	Timer 2 interrupt	T2CON.7 (TF2), T2CON.6 (EXF2)	IE.5	IP.5
TI_1 or RI_1	Serial port 1 transmit or receive	SCON1.0 (RI_1), SCON1.1 (TI_1)	IE.6	IP.6
USB	USB interrupt	EXIF.4	EIE.0	EIP.0
I^{2}C	I^{2}C interrupt	EXIT.5	EIE.1	EIP.1
INT4	External interrupt 4	EXIF.6	EIE.2	EIP.2
INT5	External interrupt 5	EXIF.7	EIE.3	EIP.3
INT6	External INT 6	EICON.3	EIE.4	EIP.4

Interrupt Sampling

The internal timers and serial ports generate interrupts by setting their respective SFR interrupt flag bits. External interrupts are sampled once per instruction cycle.

INT0 and INT1 are both active low and can be programmed to be either edge-sensitive or level-sensitive, through the IT0 and IT1 bits in the TCON SFR. For example, when IT0 = 0, INT0 is level-sensitive and the 8051 core sets the IE0 flag when the INT0# pin is sampled low. When IT0 = 1, INT0 is edge-sensitive and the 8051 sets the IE0 flag when the INT0# pin is sampled high then low on consecutive samples.

The remaining five interrupts (INT 4-6, USB & I^{2}C Interrupts) are edge-sensitive only. INT6 and INT4 are active high and INT5 is active low.

To ensure that edge-sensitive interrupts are detected, the corresponding ports should be held high for 4 CLK24 cycles and then low for 4 CLK24 cycles. Level-sensitive interrupts are not latched and must remain active until serviced.

Interrupt Latency

Interrupt response time depends on the current state of the 8051. The fastest response time is 5 instruction cycles: 1 to detect the interrupt, and 4 to perform the `LCALL` to the ISR.

The maximum latency (13 instruction cycles) occurs when the 8051 is currently executing a `RETI` instruction followed by a `MUL` or `DIV` instruction. The 13 instruction cycles in this case are: 1 to detect the interrupt, 3 to complete the `RETI`, 5 to execute the `DIV` or `MUL`, and 4 to execute the `LCALL` to the ISR. For the maximum latency case, the response time is 13 x 4 = 52 CLK24 cycles.

Single-Step Operation

The 8051 interrupt structure provides a way to perform single-step program execution. When exiting an ISR with an `RETI` instruction, the 8051 will always execute at least one instruction of the task program. Therefore, once an ISR is entered, it cannot be re-entered until at least one program instruction is executed.

To perform single-step execution, program one of the external interrupts (for example,INT0) to be level-sensitive and write an ISR for that interrupt the terminates as follows:

```
JNB  TCON.1,$   ; wait for high on INT0# pin
JB   TCON.1,$   ; wait for low on INT0# pin
RETI            ; return for ISR
```

The CPU enters the ISR when the INT0# pin goes low, then waits for a pulse on INT0#. Each time INT0# is pulsed, the CPU exits the ISR, executes one program instruction, then re-enters the ISR.

Reset

The 8051 RESET pin is internally connected to an EZ-USB register bit that is controllable through the USB host. See Chapter 10, "EZ-USB Resets" for details.

Power Saving Modes

Idle Mode

An instruction that sets the IDLE bit (PCON.0) causes the 8051 to enter idle mode when that instruction completes. In idle mode, CPU processing is suspended, and internal registers maintain their current data. When the 8051 core is in idle, the EZ-USB core enters suspend mode and shuts down the 24 MHz oscillator. See Chapter 11, "EZ-USB Power Management" for a full description of the Suspend/Resume process.

Table C-20: PCON Register - SFR 87h

Bit	Function
PCON.7	SMOD0 - Serial Port 0 baud rate double enable. When SMOD0 = 1, the baud rate for Serial Port 0 is doubled.
PCON.6-4	Reserved.
PCON.3	GF1 - General purpose flag 1. Bit-addressable, general purpose flag for software control.
PCON.2	GF0 - General purpose flag 0. Bit-addressable, general purpose flag for software control.
PCON.1	This bit should always be set to 0.
PCON.0	IDLE - Idle mode select. Setting the IDLE bit places the 8051 in idle mode.