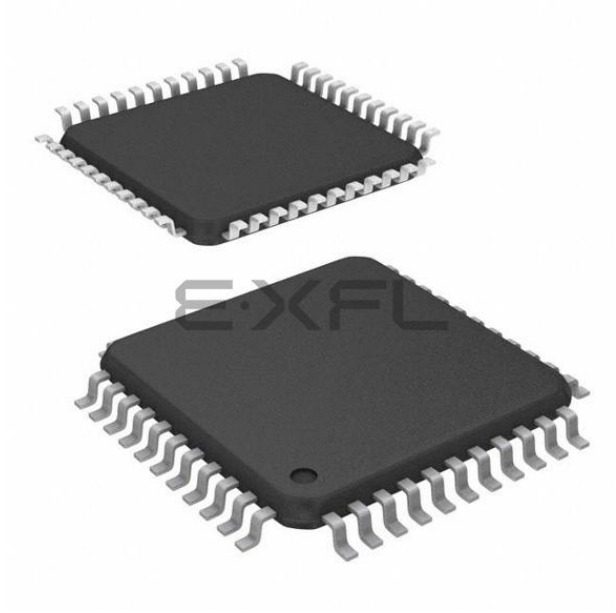


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	C251
Core Size	8/16-Bit
Speed	24MHz
Connectivity	EBI/EMI, I²C, Microwire, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-VQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at80251g2d-rltum

- **Typical Operating Current:11 mA at 3V**
- **Typical Power-down Current: 1 µA**
- **Temperature Ranges: Commercial (0°C to +70°C), Industrial (-40°C to +85°C), Automotive ((-40°C to +85°C) ROM only)**
- **Option: Extended Range (-55°C to +125°C)**
- **Packages: PDIL 40, PLCC 44 and VQFP 44**
- **Options: Known Good Dice and Ceramic Packages**

Description

The TSC80251G2D products are derivatives of the Atmel Microcontroller family based on the 8/16-bit C251 Architecture. This family of products is tailored to 8/16-bit microcontroller applications requiring an increased instruction throughput, a reduced operating frequency or a larger addressable memory space. The architecture can provide a significant code size reduction when compiling C programs while fully preserving the legacy of C51 assembly routines.

The TSC80251G2D derivatives are pin and software compatible with standard 80C51/Fx/Rx/Rx+ with extended on-chip data memory (1 Kbyte RAM) and up to 256 kilobytes of external code and data. Additionally, the TSC83251G2D and TSC87251G2D provide on-chip code memory: 32 kilobytes ROM and 32 kilobytes EPROM/OTPROM respectively.

They provide transparent enhancements to Intel's 8xC251Sx family with an additional Synchronous Serial Link Controller (SSLC supporting TWI, µWire and SPI protocols), a Keyboard interrupt interface, a dedicated Baud Rate Generator for UART, and Power Management features.

TSC80251G2D derivatives are optimized for speed and for low power consumption on a wide voltage range.

Note: 1. This Datasheet provides the technical description of the TSC80251G2D derivatives. For further information on the device usage, please request the TSC80251 Programmer's Guide and the TSC80251G1D Design Guide and errata sheet.

Typical Applications

- ISDN Terminals
- High-Speed Modems
- PABX (SOHO)
- Line Cards
- DVD ROM and Players
- Printers
- Plotters
- Scanners
- Banking Machines
- Barcode Readers
- Smart Cards Readers
- High-End Digital Monitors
- High-End Joysticks
- High-end TV's

Pin Description

Pinout

Figure 1. TSC80251G2D 40-pin DIP package

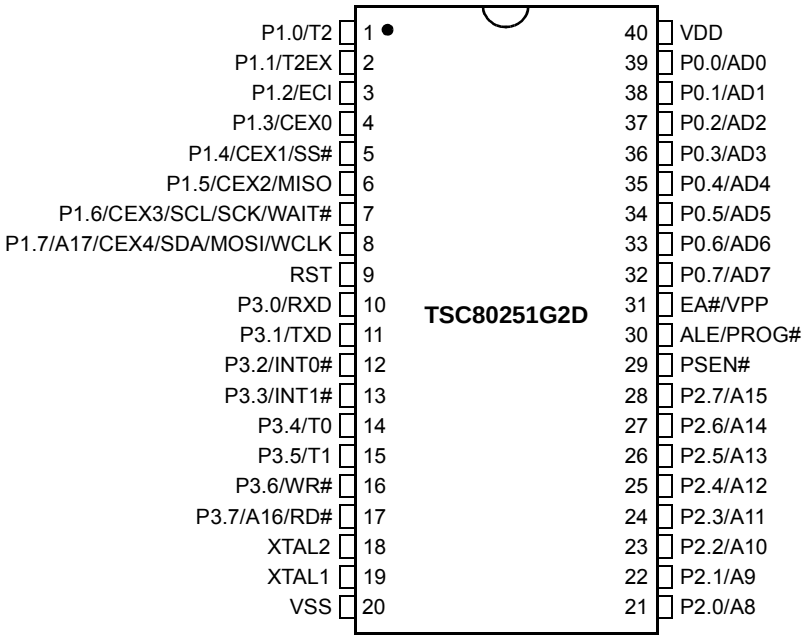


Figure 2. TSC80251G2D 44-pin PLCC Package

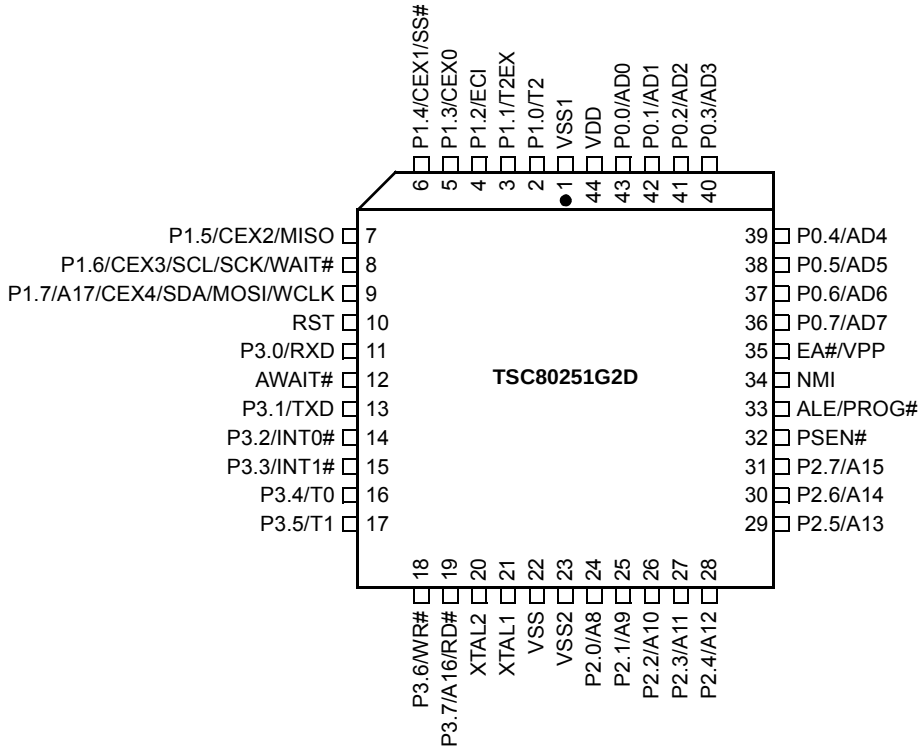


Table 2. Product Name Signal Description (Continued)

Signal Name	Type	Description	Alternate Function
NMI	I	**Non Maskable Interrupt** Holding this pin high for 24 oscillator periods triggers an interrupt. When using the Product Name as a pin-for-pin replacement for a 8xC51 product, NMI can be unconnected without loss of compatibility or power consumption increase (on-chip pull-down). Not available on DIP package.	–
P0.0:7	I/O	**Port 0** P0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high impedance inputs. To avoid any paraitic current consumption, Floating P0 inputs must be polarized to V_{DD} or V_{SS}.	AD7:0
P1.0:7	I/O	**Port 1** P1 is an 8-bit bidirectional I/O port with internal pull-ups. P1 provides interrupt capability for a keyboard interface.	–
P2.0:7	I/O	**Port 2** P2 is an 8-bit bidirectional I/O port with internal pull-ups.	A15:8
P3.0:7	I/O	**Port 3** P3 is an 8-bit bidirectional I/O port with internal pull-ups.	–
PROG#	I	**Programming Pulse input** The programming pulse is applied to this input for programming the on-chip EPROM/OTPROM.	–
PSEN#	O	**Program Store Enable/Read signal output** PSEN# is asserted for a memory address range that depends on bits RD0 and RD1 in UCONFIG0 byte (see).	–
RD#	O	**Read or 17th Address Bit (A16)** Read signal output to external data memory depending on the values of bits RD0 and RD1 in UCONFIG0 byte (see Table 13, Page 20).	P3.7
RST	I	**Reset input to the chip** Holding this pin high for 64 oscillator periods while the oscillator is running resets the device. The Port pins are driven to their reset conditions when a voltage greater than V_{IH1} is applied, whether or not the oscillator is running. This pin has an internal pull-down resistor which allows the device to be reset by connecting a capacitor between this pin and VDD. Asserting RST when the chip is in Idle mode or Power-Down mode returns the chip to normal operation.	–
RXD	I/O	**Receive Serial Data** RXD sends and receives data in serial I/O mode 0 and receives data in serial I/O modes 1, 2 and 3.	P3.0
SCL	I/O	**TWI Serial Clock** When TWI controller is in master mode, SCL outputs the serial clock to slave peripherals. When TWI controller is in slave mode, SCL receives clock from the master controller.	P1.6
SCK	I/O	**SPI Serial Clock** When SPI is in master mode, SCK outputs clock to the slave peripheral. When SPI is in slave mode, SCK receives clock from the master controller.	P1.6
SDA	I/O	**TWI Serial Data** SDA is the bidirectional TWI data line.	P1.7
SS#	I	**SPI Slave Select Input** When in Slave mode, SS# enables the slave mode.	P1.4

compatibility with the C51 Architecture). When PC increments beyond the end of segment FE:, it continues at the reset address FF:0000h (linearity). When PC increments beyond the end of segment 01:, it loops to the beginning of segment 00: (this prevents from its going into the reserved area).

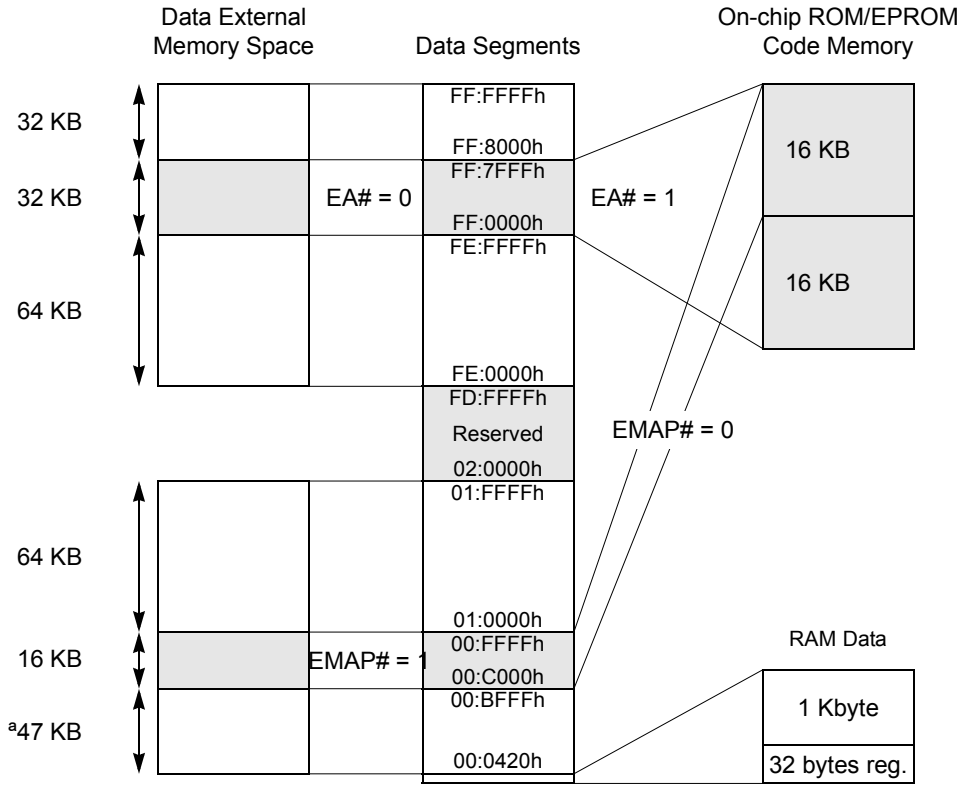
Data Memory

The TSC80251G2D derivatives implement 1 Kbyte of on-chip data RAM. Figure 5 shows the split of the internal and external data memory spaces. This memory is mapped in the data space just over the 32 bytes of registers area (see TSC80251 Programmers' Guide). Hence, the part of the on-chip RAM located from 20h to FFh is bit addressable. This on-chip RAM is not accessible through the program/code memory space.

For faster computation with the on-chip ROM/EPROM code of the TSC83251G2D/TSC87251G2D, its upper 16 KB are also mapped in the upper part of the region 00: if the On-Chip Code Memory Map configuration bit is cleared (EMAP# bit in UCONFIG1 byte, see Figure). However, if EA# is tied to a low level, the TSC80251G2D derivative is running as a ROMless product and the code is actually fetched in the corresponding external memory (i.e. the upper 16 KB of the lower 32 KB of the segment FF:). If EMAP# bit is set, the on-chip ROM is not accessible through the region 00:.

All the accesses to the portion of the data space with no on-chip memory mapped onto are redirected to the external memory.

Figure 5. Data Memory Mapping



4135F–8051–11/06

Table 12. Configuration Byte 1
UCONFIG1

7	6	5	4	3	2	1	0
CSIZE	**-**	**-**	**INTR**	**WSB**	**WSB1#**	**WSB0#**	**EMAP#**

Bit Number	Bit Mnemonic	Description
7	CSIZE TSC87251G2D	**On-Chip Code Memory Size bit**(1) Clear to select 16 KB of on-chip code memory (TSC87251G1D product). Set to select 32 KB of on-chip code memory (TSC87251G2D product).
	TSC80251G2D TSC83251G2D	**Reserved** Set this bit when writing to UCONFIG1.
6	-	**Reserved** Set this bit when writing to UCONFIG1.
5	-	**Reserved** Set this bit when writing to UCONFIG1.
4	INTR	**Interrupt Mode bit**(2) Clear so that the interrupts push two bytes onto the stack (the two lower bytes of the PC register). Set so that the interrupts push four bytes onto the stack (the three bytes of the PC register and the PSW1 register).
3	WSB	**Wait State B bit**(3) Clear to generate one wait state for memory region 01:. Set for no wait states for memory region 01:.
2	WSB1#	**Wait State B bits** Select the number of wait states for RD#, WR# and PSEN# signals for external memory accesses (only region 01:). WSB1# WSB0# Number of Wait States 0 0 3 0 1 2 1 0 1 1 1 0
1	WSB0#	
0	EMAP#	**On-Chip Code Memory Map bit** Clear to map the upper 16 KB of on-chip code memory (at FF:4000h-FF:7FFFh) to the data space (at 00:C000h-00:FFFFh). Set not to map the upper 16 KB of on-chip code memory (at FF:4000h-FF:7FFFh) to the data space.

Notes:
1. The CSIZE is only available on EPROM/OTPROM products.
2. Two or four bytes are transparently popped according to INTR when using the RETI instruction. INTR must be set if interrupts are used with code executing outside region FF:.
3. Use only for Step A compatibility; set this bit when WSB1:0# are used.

Configuration Byte 1

Table 13. Address Ranges and Usage of RD#, WR# and PSEN# Signals

RD1	RD0	P1.7	P3.7/RD#	PSEN#	WR#	External Memory
0	0	A17	A16	Read signal for all external memory locations	Write signal for all external memory locations	256 KB
0	1	I/O pin	A16	Read signal for all external memory locations	Write signal for all external memory locations	128 KB
1	0	I/O pin	I/O pin	Read signal for all external memory locations	Write signal for all external memory locations	64 KB
1	1	I/O pin	Read signal for regions 00: and 01:	Read signal for regions FE: and FF:	Write signal for all external memory locations	2×64 KB(1)

Notes: 1. This selection provides compatibility with the standard 80C51 hardware which has separate external memory spaces for data and code.

4. If this instruction addresses external memory location, add 2(N+2) to the number of states (N: number of wait states).

Table 21. Summary of Increment and Decrement Instructions

IncrementINC <dest>dest opnd ← dest opnd + 1 IncrementINC <dest>, <src>dest opnd ← dest opnd + src opnd DecrementDEC <dest>dest opnd ← dest opnd - 1 DecrementDEC <dest>, <src>dest opnd ← dest opnd - src opnd						
Mnemonic	**<dest>, <src>(1)**	**Comments**	**Binary Mode**		**Source Mode**	
			Bytes	**States**	**Bytes**	**States**
INC DEC	A	ACC by 1	1	1	1	1
	Rn	Register by 1	1	1	2	2
	dir8	Direct address (on-chip RAM or SFR) by 1	2	2(2)	2	2(2)
	at Ri	Indirect address by 1	1	3	2	4
INC DEC	Rm, #short	Byte register by 1, 2, or 4	3	2	2	1
	WRj, #short	Word register by 1, 2, or 4	3	2	2	1
INC	DRk, #short	Double word register by 1, 2, or 4	3	4	2	3
DEC	DRk, #short	Double word register by 1, 2, or 4	3	5	2	4
INC	DPTR	Data pointer by 1	1	1	1	1

Notes: 1. A shaded cell denotes an instruction in the C51 Architecture.
2. If this instruction addresses an I/O Port (Px, x = 0-3), add 2 to the number of states. Add 3 if it addresses a Peripheral SFR.

Logical AND[1]ANL <dest>, <src>dest opnd ← dest opnd Λ src opnd
Logical OR[1]ORL <dest>, <src>dest opnd ← dest opnd ς src opnd
Logical Exclusive OR[1]XRL <dest>, <src>dest opnd ← dest opnd ∀ src opnd
Clear[1]CLR A(A) ← 0
Complement[1]CPL A(A) ← ∅ (A)
Rotate LeftRL $A(A)_{n+1} \leftarrow (A)_n$, n = 0..6
$(A)_0 \leftarrow (A)_7$
Rotate Left CarryRLC $A(A)_{n+1} \leftarrow (A)_n$, n = 0..6
$(CY) \leftarrow (A)_7$
$(A)_0 \leftarrow (CY)$
Rotate RightRR $A(A)_{n-1} \leftarrow (A)_n$, n = 7..1
$(A)_7 \leftarrow (A)_0$
Rotate Right CarryRRC $A(A)_{n-1} \leftarrow (A)_n$, n = 7..1
$(CY) \leftarrow (A)_0$
$(A)_7 \leftarrow (CY)$

Mnemonic	<dest>, <src>[1]	Comments	Binary Mode		Source Mode	
			Bytes	States	Bytes	States
ANL ORL XRL	A, Rn	register to ACC	1	1	2	2
	A, dir8	Direct address (on-chip RAM or SFR) to ACC	2	1[3]	2	1[3]
	A, at Ri	Indirect address to ACC	1	2	2	3
	A, #data	Immediate data to ACC	2	1	2	1
	dir8, A	ACC to direct address	2	2[4]	2	2[4]
	dir8, #data	Immediate 8-bit data to direct address	3	3[4]	3	3[4]
	Rmd, Rms	Byte register to byte register	3	2	2	1
	WRjd, WRjs	Word register to word register	3	3	2	2
	Rm, #data	Immediate 8-bit data to byte register	4	3	3	2
	WRj, #data16	Immediate 16-bit data to word register	5	4	4	3
	Rm, dir8	Direct address (on-chip RAM or SFR) to byte register	4	3[3]	3	2[3]
	WRj, dir8	Direct address (on-chip RAM or SFR) to word register	4	4	3	3
	Rm, dir16	Direct address (64K) to byte register	5	3[5]	4	2[5]
	WRj, dir16	Direct address (64K) to word register	5	4[6]	4	3[6]
	Rm, at WRj	Indirect address (64K) to byte register	4	3[5]	3	2[5]
	Rm, at DRk	Indirect address (16M) to byte register	4	4[5]	3	3[5]
CLR	A	Clear ACC	1	1	1	1
CPL	A	Complement ACC	1	1	1	1
RL	A	Rotate ACC left	1	1	1	1
RLC	A	Rotate ACC left through CY	1	1	1	1
RR	A	Rotate ACC right	1	1	1	1
RRC	A	Rotate ACC right through CY	1	1	1	1

Table 25. Summary of Move Instructions (1/3)

Move to High wordMOVH <dest>, <src>dest $opnd_{31:16}$ ← src opnd Move with Sign extensionMOVS <dest>, <src>dest opnd ← src opnd with sign extend Move with Zero extensionMOVZ <dest>, <src>dest opnd ← src opnd with zero extend Move CodeMOVC A, <src>(A) ← src opnd Move eXtendedMOVX <dest>, <src>dest opnd ← src opnd						
Mnemonic	**<dest>, <src>(2)**	**Comments**	**Binary Mode**		**Source Mode**	
			Bytes	**States**	**Bytes**	**States**
MOVH	DRk, #data16	16-bit immediate data into upper word of dword register	5	3	4	2
MOVS	WRj, Rm	Byte register to word register with sign extension	3	2	2	1
MOVZ	WRj, Rm	Byte register to word register with zeros extension	3	2	2	1
MOVC	A, at A +DPTR	Code byte relative to DPTR to ACC	1	6(3)	1	6(3)
	A, at A +PC	Code byte relative to PC to ACC	1	6(3)	1	6(3)
MOVX	A, at Ri	Extended memory (8-bit address) to ACC(2)	1	4	1	5
	A, at DPTR	Extended memory (16-bit address) to ACC(2)	1	3(4)	1	3(4)
	at Ri, A	ACC to extended memory (8-bit address)(2)	1	4	1	4
	at DPTR, A	ACC to extended memory (16-bit address)(2)	1	4(3)	1	4(3)

Notes:
1. A shaded cell denotes an instruction in the C51 Architecture.
2. Extended memory addressed is in the region specified by DPXL (reset value = 01h).
3. If this instruction addresses external memory location, add N+1 to the number of states (N: number of wait states).
4. If this instruction addresses external memory location, add N+2 to the number of states (N: number of wait states).

Move[1] MOV <dest>, <src> dest opnd ← src opnd						
			Binary Mode		Source Mode	
Mnemonic	<dest>, <src>[1]	Comments	Bytes	States	Bytes	States
MOV	Rmd, Rms	Byte register to byte register	3	2	2	1
MOV	WRjd, WRjs	Word register to word register	3	2	2	1
MOV	DRkd, DRks	Dword register to dword register	3	3	2	2
MOV	Rm, #data	Immediate 8-bit data to byte register	4	3	3	2
MOV	WRj, #data16	Immediate 16-bit data to word register	5	3	4	2
MOV	DRk, #0data16	zero-ext 16bit immediate data to dword register	5	5	4	4
MOV	DRk, #1data16	one-ext 16bit immediate data to dword register	5	5	4	4
MOV	Rm, dir8	Direct address (on-chip RAM or SFR) to byte register	4	3[3]	3	2[3]
MOV	WRj, dir8	Direct address (on-chip RAM or SFR) to word register	4	4	3	3
MOV	DRk, dir8	Direct address (on-chip RAM or SFR) to dword register	4	6	3	5
MOV	Rm, dir16	Direct address (64K) to byte register	5	3[4]	4	2[4]
MOV	WRj, dir16	Direct address (64K) to word register	5	4[5]	4	3[5]
MOV	DRk, dir16	Direct address (64K) to dword register	5	6[6]	4	5[6]
MOV	Rm, at WRj	Indirect address (64K) to byte register	4	3[4]	3	2[4]
MOV	Rm, at DRk	Indirect address (16M) to byte register	4	4[4]	3	3[4]
MOV	WRjd, at WRjs	Indirect address (64K) to word register	4	4[5]	3	3[5]
MOV	WRj, at DRk	Indirect address (16M) to word register	4	5[5]	3	4[5]
MOV	dir8, Rm	Byte register to direct address (on-chip RAM or SFR)	4	4[3]	3	3[3]
MOV	dir8, WRj	Word register to direct address (on-chip RAM or SFR)	4	5	3	4
MOV	dir8, DRk	Dword register to direct address (on-chip RAM or SFR)	4	7	3	6
MOV	dir16, Rm	Byte register to direct address (64K)	5	4[4]	4	3[4]
MOV	dir16, WRj	Word register to direct address (64K)	5	5[5]	4	4[5]
MOV	dir16, DRk	Dword register to direct address (64K)	5	7[6]	4	6[6]
MOV	at WRj, Rm	Byte register to indirect address (64K)	4	4[4]	3	3[4]
MOV	at DRk, Rm	Byte register to indirect address (16M)	4	5[4]	3	4[4]
MOV	at WRjd, WRjs	Word register to indirect address (64K)	4	5[5]	3	4[5]
MOV	at DRk, WRj	Word register to indirect address (16M)	4	6[5]	3	5[5]
MOV	Rm, at WRj +dis16	Indirect with 16-bit displacement (64K) to byte register	5	6[4]	4	5[4]
MOV	WRj, at WRj +dis16	Indirect with 16-bit displacement (64K) to word register	5	7[5]	4	6[5]
MOV	Rm, at DRk +dis24	Indirect with 16-bit displacement (16M) to byte register	5	7[4]	4	6[4]

Table 29. Summary of Conditional Jump Instructions (1/2)

Jump conditional on statusJcc rel(PC) ← (PC) + size (instr); IF [cc] THEN (PC) ← (PC) + rel						
			Binary Mode		Source Mode	
Mnemonic	**<dest>, <src>[1]**	**Comments**	**Bytes**	**States**	**Bytes**	**States**
JC	rel	Jump if carry	2	1/4[3]	2	1/4[3]
JNC	rel	Jump if not carry	2	1/4[3]	2	1/4[3]
JE	rel	Jump if equal	3	2/5[3]	2	1/4[3]
JNE	rel	Jump if not equal	3	2/5[3]	2	1/4[3]
JG	rel	Jump if greater than	3	2/5[3]	2	1/4[3]
JLE	rel	Jump if less than, or equal	3	2/5[3]	2	1/4[3]
JSL	rel	Jump if less than (signed)	3	2/5[3]	2	1/4[3]
JSLE	rel	Jump if less than, or equal (signed)	3	2/5[3]	2	1/4[3]
JSG	rel	Jump if greater than (signed)	3	2/5[3]	2	1/4[3]
JSGE	rel	Jump if greater than or equal (signed)	3	2/5[3]	2	1/4[3]

Notes:
1. A shaded cell denotes an instruction in the C51 Architecture.
2. States are given as jump not-taken/taken.
3. In internal execution only, add 1 to the number of states of the ‘jump taken’ if the destination address is internal and odd.

Add 3 if it addresses a Peripheral SFR.

5. If this instruction addresses an I/O Port (Px, x = 0-3), add 3 to the number of states. Add 5 if it addresses a Peripheral SFR.
6. In internal execution only, add 1 to the number of states of the 'jump taken' if the destination address is internal and odd.

Table 31. Summary of Unconditional Jump Instructions

Absolute jumpAJMP <src>(PC) ← (PC) +2; $(PC)_{10:0}$ ← src opnd						
Extended jumpEJMP <src>(PC) ← (PC) + size (instr); $(PC)_{23:0}$ ← src opnd						
Long jumpLJMP <src>(PC) ← (PC) + size (instr); $(PC)_{15:0}$ ← src opnd						
Short jumpSJMP rel(PC) ← (PC) +2; (PC) ← (PC) +rel						
Jump indirectJMP at A +DPTR$(PC)_{23:16}$ ← FFh; $(PC)_{15:0}$ ← (A) + (DPTR)						
No operationNOP(PC) ← (PC) +1						
Mnemonic	**<dest>, <src>(1)**	**Comments**	**Binary Mode**		**Source Mode**	
			Bytes	**States**	**Bytes**	**States**
AJMP	addr11	Absolute jump	2	3(2)(3)	2	3(2)(3)
EJMP	addr24	Extended jump	5	6(2)(4)	4	5(2)(4)
	at DRk	Extended jump (indirect)	3	7(2)(4)	2	6(2)(4)
LJMP	at WRj	Long jump (indirect)	3	6(2)(4)	2	5(2)(4)
	addr16	Long jump (direct address)	3	5(2)(4)	3	5(2)(4)
SJMP	rel	Short jump (relative address)	2	4(2)(4)	2	4(2)(4)
JMP	at A +DPTR	Jump indirect relative to the DPTR	1	5(2)(4)	1	5(2)(4)
NOP		No operation (Jump never)	1	1	1	1

Notes:
1. A shaded cell denotes an instruction in the C51 Architecture.
2. In internal execution only, add 1 to the number of states if the destination address is internal and odd.
3. Add 2 to the number of states if the destination address is external.
4. Add 3 to the number of states if the destination address is external.

- PSEN# and the other control signals have to be released to complete a sequence of programming operations or a sequence of programming and verifying operations.

Figure 6. Setup for Programming

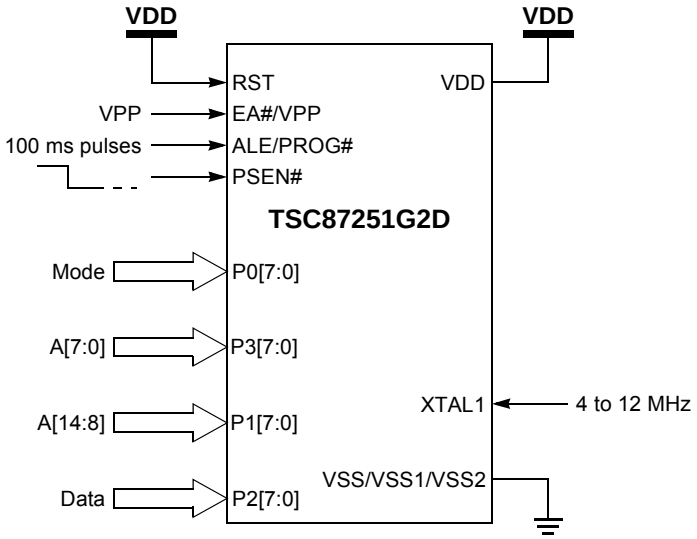


Table 36. Programming Modes

ROM Area[1]	RST	EA#/VPP	PSEN #	ALE/PROG#[2]	P0	P2	P1(MSB) P3(LSB)
On-chip Code Memory	1	V_{PP}	0	1 Pulse	68h	Data	16-bit Address 0000h-7FFFh (32 kilobytes)
Configuration Bytes	1	V_{PP}	0	1 Pulse	69h	Data	CONFIG0: FFF8h CONFIG1: FFF9h
Lock Bits	1	V_{PP}	0	1 Pulse	6Bh	X	LB0: 0001h LB1: 0002h LB2: 0003h
Encryption Array	1	V_{PP}	0	1 Pulse	6Ch	Data	0000h-007Fh

Notes: 1. Signature Bytes are not user-programmable.
2. The ALE/PROG# pulse waveform is shown in Figure 23 page 59.

Verify Algorithm

Figure 7 shows the hardware setup needed to verify the TSC87251G2D EPROM/OTPROM or TSC83251G2D ROM areas:

- The chip has to be put under reset and maintained in this state until the completion of the verifying sequence.
- PSEN# and the other control signals (ALE and Port 0) have to be set to a high level.
- Then PSEN# has to be to forced to a low level after two clock cycles or more and it has to be maintained in this state until the completion of the verifying sequence (see below).
- The voltage on the EA# pin must be set to V_{DD} and ALE must be set to a high level.
- The Verifying Mode is selected according to the code applied on Port 0. It has to be applied until the completion of this verifying operation.
- The verifying address is applied on Ports 1 and 3 which are respectively the MSB and the LSB of the address.

- Then device is driving the data on Port 2.
- It is possible to alternate programming and verification operation (see Paragraph Programming Algorithm). Please make sure the voltage on the EA# pin has actually been lowered to V_{DD} before performing the verifying operation.
- PSEN# and the other control signals have to be released to complete a sequence of verifying operations or a sequence of programming and verifying operations.

Table 37. Verifying Modes

ROM Area[1]	RST	EA#/VPP	PSEN#	ALE/PROG#	P0	P2	P1(MSB) P3(LSB)
On-chip code memory	1	1	0	1	28h	Data	16-bit Address 0000h-7FFFh (32 kilobytes)
Configuration Bytes	1	1	0	1	29h	Data	CONFIG0: FFF8h CONFIG1: FFF9h
Lock Bits	1	1	0	1	2Bh	Data	0000h
Signature Bytes	1	1	0	1	29h	Data	0030h, 0031h, 0060h, 0061h

Notes: 1. To preserve the secrecy of on-chip code memory when encrypted, the Encryption Array can not be verified.

Figure 7. Setup for Verifying

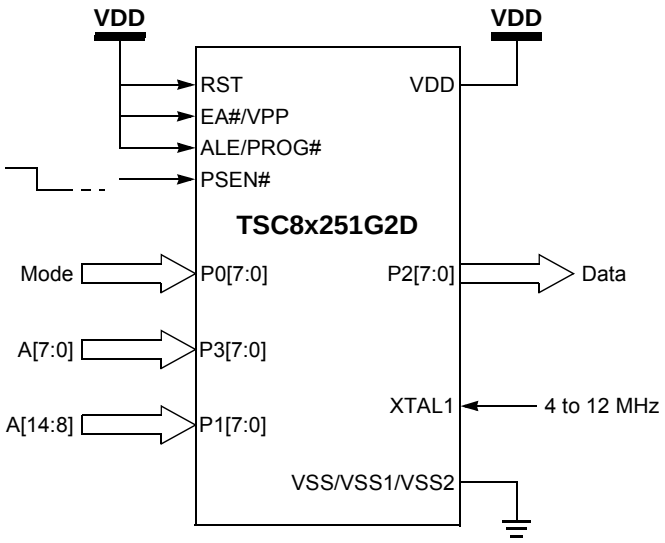


Table 39. Bus Cycles AC Timings; V_{DD} = 4.5 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	12 MHz		16 MHz		24 MHz		Unit
		Min	Max	Min	Max	Min	Max	
T_{OSC}	$1/F_{OSC}$	83		62		41		ns
T_{LHLL}	ALE Pulse Width	78		58		38		ns(2)
T_{AVLL}	Address Valid to ALE Low	78		58		37		ns(2)
T_{LLAX}	Address hold after ALE Low	19		11		3		ns
T_{RLRH}(1)	RD#/PSEN# Pulse Width	162		121		78		ns(3)
T_{WLWH}	WR# Pulse Width	165		124		81		ns(3)
T_{LLRL}(1)	ALE Low to RD#/PSEN# Low	22		14		6		ns
T_{LHAX}	ALE High to Address Hold	99		70		40		ns(2)
T_{RLDV}(1)	RD#/PSEN# Low to Valid Data		146		104		61	ns(3)
T_{RHDX}(1)	Data Hold After RD#/PSEN# High	0		0		0		ns
T_{RHAX}(1)	Address Hold After RD#/PSEN# High	0		0		0		ns
T_{RLAZ}(1)	RD#/PSEN# Low to Address Float		0		0		0	ns
T_{RHDZ1}	Instruction Float After RD#/PSEN# High		45		40		30	ns
T_{RHDZ2}	Data Float After RD#/PSEN# High		215		165		115	ns
T_{RHLH1}	RD#/PSEN# high to ALE High (Instruction)	49		43		31		ns
T_{RHLH2}	RD#/PSEN# high to ALE High (Data)	215		169		115		ns
T_{WHLH}	WR# High to ALE High	215		169		115		ns
T_{AVDV1}	Address (P0) Valid to Valid Data In		250		175		105	ns(2)(3)
T_{AVDV2}	Address (P2) Valid to Valid Data In		306		223		140	ns(2)(3)
T_{AVDV3}	Address (P0) Valid to Valid Instruction In		150		109		68	ns(3)
T_{AXDX}	Data Hold after Address Hold	0		0		0		ns
T_{AVRL}(1)	Address Valid to RD# Low	100		70		40		ns(2)
T_{AVWL1}	Address (P0) Valid to WR# Low	100		70		40		ns(2)
T_{AVWL2}	Address (P2) Valid to WR# Low	158		115		74		ns(2)
T_{WHQX}	Data Hold after WR# High	90		69		32		ns
T_{QVWH}	Data Valid to WR# High	133		102		72		ns(3)
T_{WHAX}	WR# High to Address Hold	167		125		84		ns

Notes:
1. Specification for PSEN# are identical to those for RD#.
2. If a wait state is added by extending ALE, add $2 \cdot T_{OSC.}$
3. If wait states are added by extending RD#/PSEN#/WR#, add $2N \cdot T_{OSC}$ (N = 1..3).

AC Characteristics - SSLC: TWI Interface

Timings

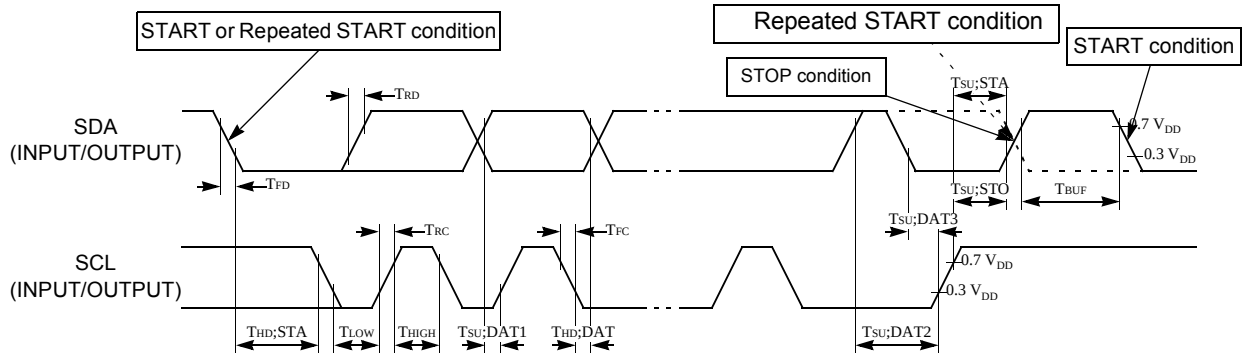
Table 47. TWI Interface AC Timing; V_{DD} = 2.7 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	INPUT Min	INPUT Max	OUTPUT Min	OUTPUT Max
T_{HD}; STA	Start condition hold time	14·T_{CLCL}(4)		4.0 µs(1)	
T_{LOW}	SCL low time	16·T_{CLCL}(4)		4.7 µs(1)	
T_{HIGH}	SCL high time	14·T_{CLCL}(4)		4.0 µs(1)	
T_{RC}	SCL rise time	1 µs		-(2)	
T_{FC}	SCL fall time	0.3 µs		0.3 µs(3)	
T_{SU}; DAT1	Data set-up time	250 ns		20·T_{CLCL}(4)- T_{RD}	
T_{SU}; DAT2	SDA set-up time (before repeated START condition)	250 ns		1 µs(1)	
T_{SU}; DAT3	SDA set-up time (before STOP condition)	250 ns		8·T_{CLCL}(4)	
T_{HD}; DAT	Data hold time	0 ns		8·T_{CLCL}(4) - T_{FC}	
T_{SU}; STA	Repeated START set-up time	14·T_{CLCL}(4)		4.7 µs(1)	
T_{SU}; STO	STOP condition set-up time	14·T_{CLCL}(4)		4.0 µs(1)	
T_{BUF}	Bus free time	14·T_{CLCL}(4)		4.7 µs(1)	
T_{RD}	SDA rise time	1 µs		-(2)	
T_{FD}	SDA fall time	0.3 µs		0.3 µs(3)	

Notes:
1. At 100 kbit/s. At other bit-rates this value is inversely proportional to the bit-rate of 100 kbit/s.
2. Determined by the external bus-line capacitance and the external bus-line pull-up resistor, this must be < 1 µs.
3. Spikes on the SDA and SCL lines with a duration of less than 3·T_{CLCL} will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400 pF.
4. T_{CLCL} = T_{OSC} = one oscillator clock period.

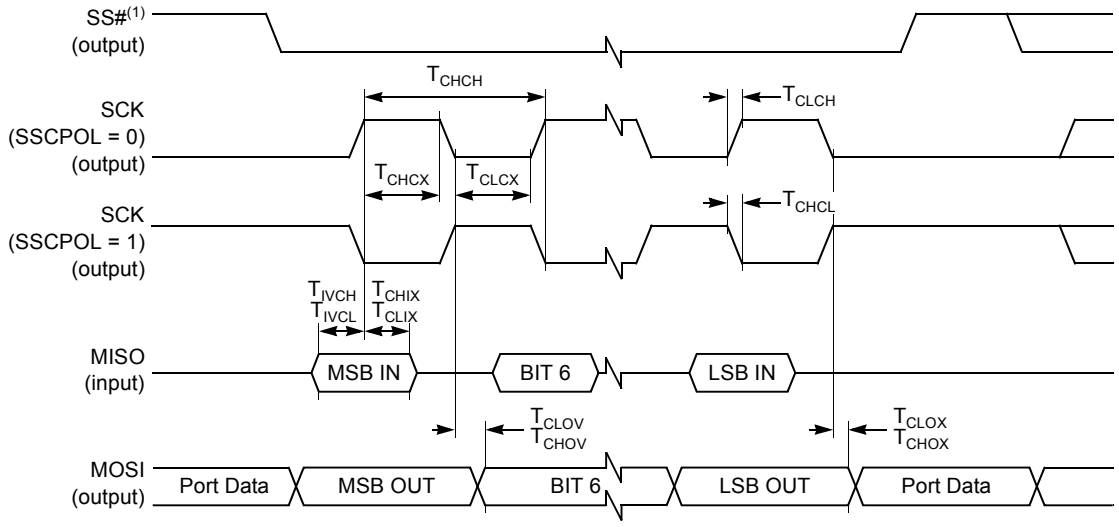
Waveforms

Figure 18. TWI Waveforms



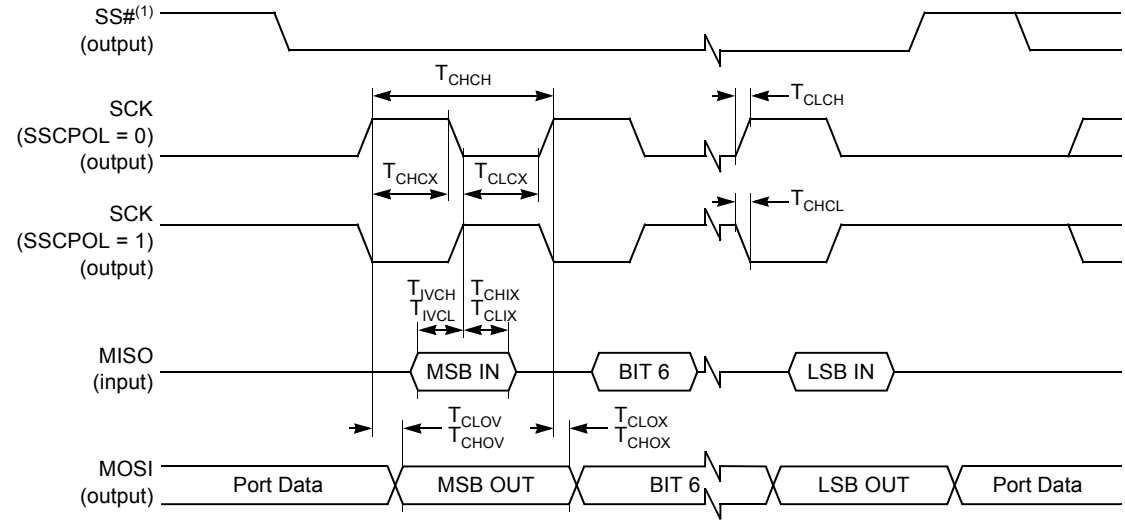
Waveforms

Figure 19. SPI Master Waveforms (SSCPHA = 0)



Note: 1. SS# handled by software.

Figure 20. SPI Master Waveforms (SSCPHA = 1)



Note: 1. Not Defined but normally MSB of character just received.

Timings

Table 51. EPROM Programming AC timings; V_{DD} = 4.5 to 5.5 V, T_A = 0 to 40°C

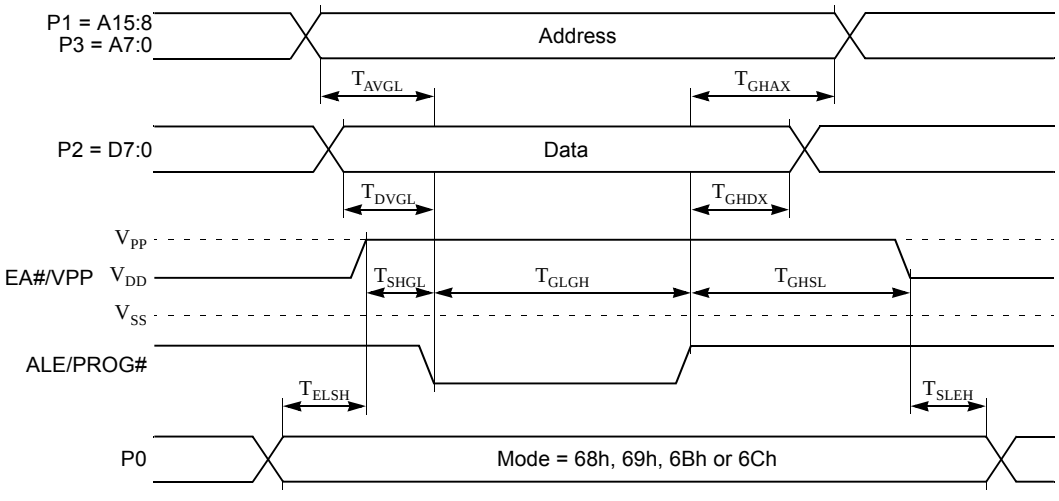
Symbol	Parameter	Min	Max	Unit
T_{OSC}	XTAL1 Period	83.5	250	ns
T_{AVGL}	Address Setup to PROG# low	48		T_{OSC}
T_{GHAX}	Address Hold after PROG# low	48		T_{OSC}
T_{DVGL}	Data Setup to PROG# low	48		T_{OSC}
T_{GHDX}	Data Hold after PROG#	48		T_{OSC}
T_{ELSH}	ENABLE High to V_{PP}	48		T_{OSC}
T_{SHGL}	V_{PP} Setup to PROG# low	10		µs
T_{GHSL}	V_{PP} Hold after PROG#	10		µs
T_{SLEH}	ENABLE Hold after V_{PP}	0		ns
T_{GLGH}	PROG# Width	90	110	µs

Table 52. EPROM Verifying AC timings; V_{DD} = 4.5 to 5.5 V, V_{DD} = 2.7 to 5.5 V, T_A = 0 to 40°C

Symbol	Parameter	Min	Max	Unit
T_{OSC}	XTAL1 Period	83.5	250	ns
T_{AVQV}	Address to Data Valid		48	T_{OSC}
T_{AXQX}	Address to Data Invalid	0		ns
T_{ELQV}	ENABLE low to Data Valid	0	48	T_{OSC}
T_{EHQZ}	Data Float after ENABLE	0	48	T_{OSC}

Waveforms

Figure 23. EPROM Programming Waveforms



Notes: 1. Under steady-state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum IOL per port pin: 10 mA

Maximum IOL per 8-bit port:Port 0 26 mA

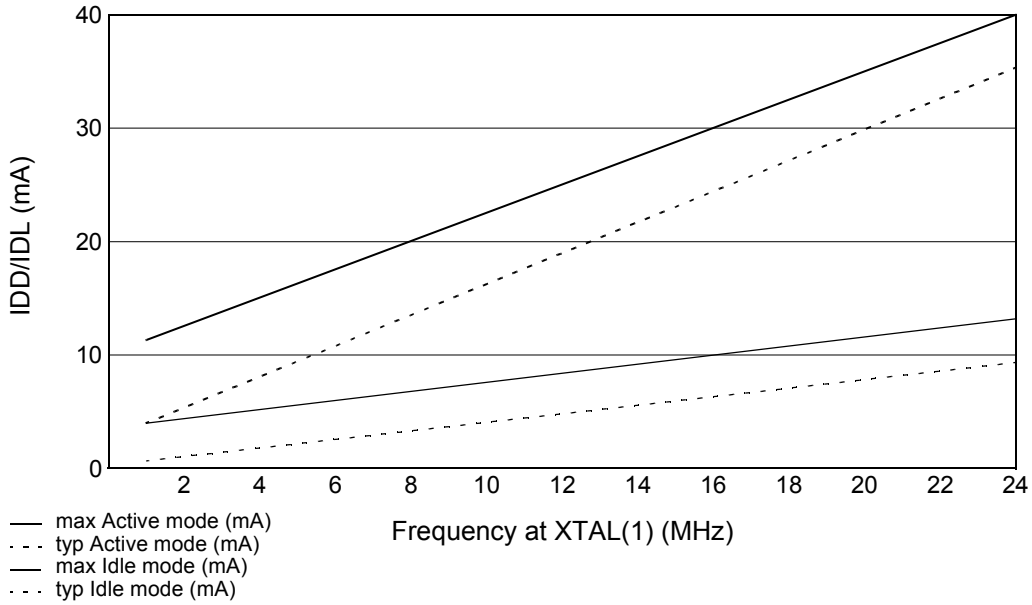
Ports 1-3 15 mA

Maximum Total IOL for all: Output Pins 71 mA

If IOL exceeds the test conditions, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses above 0.4 V on the low-level outputs of ALE and Ports 1, 2, and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins change from high to low. In applications where capacitive loading exceeds 100 pF, the noise pulses on these signals may exceed 0.8 V. It may be desirable to qualify ALE or other signals with a Schmitt Trigger or CMOS-level input logic.
3. Capacitive loading on Ports 0 and 2 causes the V_{OH} on ALE and PSEN# to drop below the specification when the address lines are stabilizing.
4. Typical values are obtained using V_{DD} = 5 V and T_A = 25°C. They are not tested and there is not guarantee on these values.
5. The input threshold voltage of SCL and SDA meets the TWI specification, so an input voltage below $0.3 \cdot V_{DD}$ will be recognized as a logic 0 while an input voltage above $0.7 \cdot V_{DD}$ will be recognized as a logic 1.

Figure 28. I_{DD}/I_{DL} Versus Frequency; V_{DD} = 4.5 to 5.5 V



Note: 1. The clock prescaler is not used: $F_{OSC} = F_{XTAL}$.

Figure 31. I_{DL} Test Condition, Idle Mode

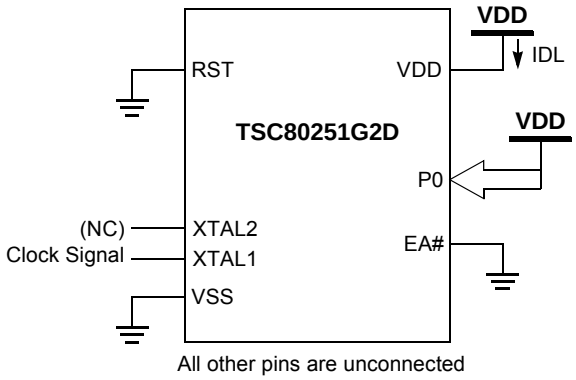


Figure 32. I_{PD} Test Condition, Power-Down Mode