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	0°C ~ 70°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/tsc80251g2d-24ce

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

Address Spaces

The TSC80251G2D derivatives implement four different address spaces:

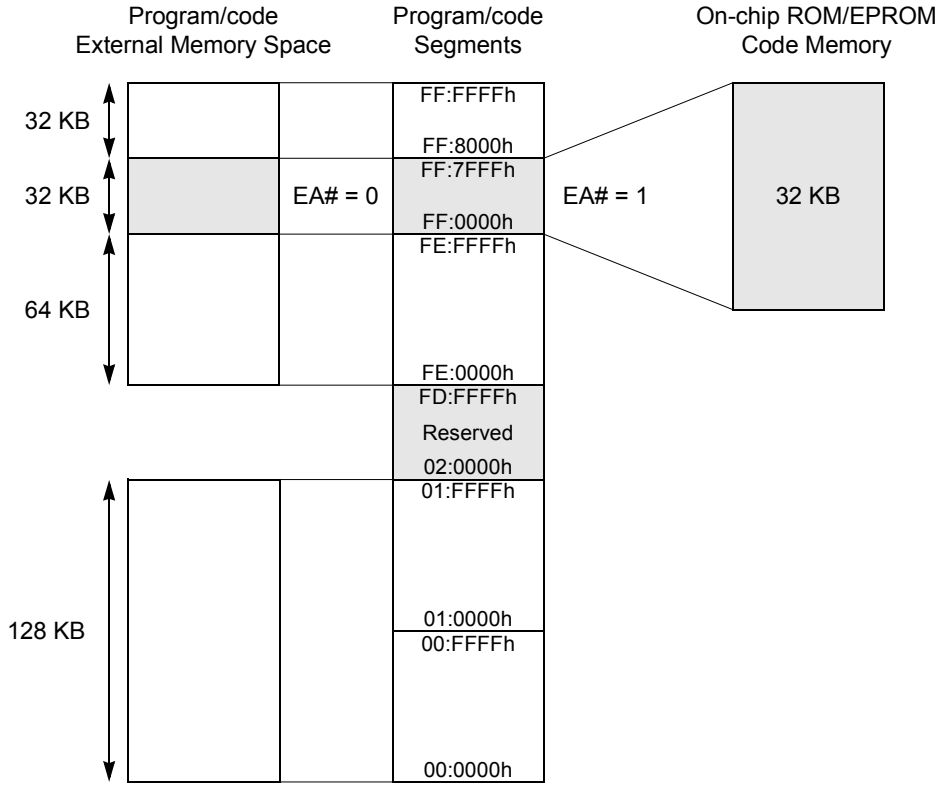
- On-chip ROM program/code memory (not present in ROMless devices)
- On-chip RAM data memory
- Special Function Registers (SFRs)
- Configuration array

Program/Code Memory

The TSC83251G2D and TSC87251G2D implement 32 KB of on-chip program/code memory. Figure 4 shows the split of the internal and external program/code memory spaces. If EA# is tied to a high level, the 32-Kbyte on-chip program memory is mapped in the lower part of segment FF: where the C251 core jumps after reset. The rest of the program/code memory space is mapped to the external memory. If EA# is tied to a low level, the internal program/code memory is not used and all the accesses are directed to the external memory.

The TSC83251G2D products provide the internal program/code memory in a masked ROM memory while the TSC87251G2D products provide it in an EPROM memory. For the TSC80251G2D products, there is no internal program/code memory and EA# must be tied to a low level.

Figure 4. Program/Code Memory Mapping



Note: Special care should be taken when the Program Counter (PC) increments:
If the program executes exclusively from on-chip code memory (not from external memory), beware of executing code from the upper eight bytes of the on-chip ROM (FF:7FF8h-FF:7FFFh). Because of its pipeline capability, the TSC80251G2D derivative may attempt to prefetch code from external memory (at an address above FF:7FFFh) and thereby disrupt I/O Ports 0 and 2. Fetching code constants from these 8 bytes does not affect Ports 0 and 2.
When PC reaches the end of segment FF:, it loops to the reset address FF:0000h (for

Table 11. Configuration Byte 0
UCONFIG0

7	6	5	4	3	2	1	0
-	WSA1#	WSA0#	XALE#	RD1	RD0	PAGE#	SRC

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** Set this bit when writing to UCONFIG0.
6	WSA1#	**Wait State A bits** Select the number of wait states for RD#, WR# and PSEN# signals for external memory accesses (all regions except 01:). WSA1# WSA0# Number of Wait States 0 0 3 0 1 2 1 0 1 1 1 0
5	WSA0#	
4	XALE#	**Extend ALE bit** Clear to extend the duration of the ALE pulse from T_{OSC} to $3 \cdot T_{OSC}$. Set to minimize the duration of the ALE pulse to $1 \cdot T_{OSC}$.
3	RD1	**Memory Signal Select bits** Specify a 18-bit, 17-bit or 16-bit external address bus and the usage of RD#, WR# and PSEN# signals (see Table 13).
2	RD0	
1	PAGE#	**Page Mode Select bit**(1) Clear to select the faster Page mode with A15:8/D7:0 on Port 2 and A7:0 on Port 0. Set to select the non-Page mode(2) with A15:8 on Port 2 and A7:0/D7:0 on Port 0.
0	SRC	**Source Mode/Binary Mode Select bit** Clear to select the binary mode. Set to select the source mode.

Notes: 1. UCONFIG0 is fetched twice so it can be properly read both in Page or Non-Page modes. If P2.1 is cleared during the first data fetch, a Page mode configuration is used, otherwise the subsequent fetches are performed in Non-Page mode.
2. This selection provides compatibility with the standard 80C51 hardware which is multiplexing the address LSB and the data on Port 0.

Instruction Set Summary

This section contains tables that summarize the instruction set. For each instruction there is a short description, its length in bytes, and its execution time in states (one state time is equal to two system clock cycles). There are two concurrent processes limiting the effective instruction throughput:

- Instruction Fetch
- Instruction Execution

Table 20 to Table 32 assume code executing from on-chip memory, then the CPU is fetching 16-bit at a time and this is never limiting the execution speed.

If the code is fetched from external memory, a pre-fetch queue will store instructions ahead of execution to optimize the memory bandwidth usage when slower instructions are executed. However, the effective speed may be limited depending on the average size of instructions (for the considered section of the program flow). The maximum average instruction throughput is provided by Table 14 depending on the external memory configuration (from Page Mode to Non-Page Mode and the maximum number of wait states). If the average size of instructions is not an integer, the maximum effective throughput is found by pondering the number of states for the neighbor integer values.

Table 14. Minimum Number of States per Instruction for given Average Sizes

Average size of Instructions (bytes)	Page Mode (states)	Non-page Mode (states)				
		0 Wait State	1 Wait State	2 Wait States	3 Wait States	4 Wait States
1	1	2	3	4	5	6
2	2	4	6	8	10	12
3	3	6	9	12	15	18
4	4	8	12	16	20	24
5	5	10	15	20	25	30

If the average execution time of the considered instructions is larger than the number of states given by Table 14, this larger value will prevail as the limiting factor. Otherwise, the value from Table 14 must be taken. This is providing a fair estimation of the execution speed but only the actual code execution can provide the final value.

Notation for Instruction Operands

Table 15 to Table 19 provide notation for Instruction Operands.

Table 15. Notation for Direct Addressing

Direct Address	Description	C251	C51
dir8	A direct 8-bit address. This can be a memory address (00h-7Fh) or a SFR address (80h-FFh). It is a byte (default), word or double word depending on the other operand.	3	3
dir16	A 16-bit memory address (00:0000h-00:FFFFh) used in direct addressing.	3	–

Size and Execution Time for Instruction Families

Table 20. Summary of Add and Subtract Instructions

AddADD <dest>, <src>dest opnd ← dest opnd + src opnd SubtractSUB <dest>, <src>dest opnd ← dest opnd - src opnd Add with CarryADDC <dest>, <src>(A) ← (A) + src opnd + (CY) Subtract with BorrowSUBB <dest>, <src>(A) ← (A) - src opnd - (CY)						
Mnemonic	**<dest>, <src>(1)**	**Comments**	**Binary Mode**		**Source Mode**	
			Bytes	**States**	**Bytes**	**States**
ADD	A, Rn	Register to ACC	1	1	2	2
	A, dir8	Direct address to ACC	2	1(2)	2	1(2)
	A, at Ri	Indirect address to ACC	1	2	2	3
	A, #data	Immediate data to ACC	2	1	2	1
ADD/SUB	Rmd, Rms	Byte register to/from byte register	3	2	2	1
	WRjd, WRjs	Word register to/from word register	3	3	2	2
	DRkd, DRks	Dword register to/from dword register	3	5	2	4
	Rm, #data	Immediate 8-bit data to/from byte register	4	3	3	2
	WRj, #data16	Immediate 16-bit data to/from word register	5	4	4	3
	DRk, #0data16	16-bit unsigned immediate data to/from dword register	5	6	4	5
	Rm, dir8	Direct address (on-chip RAM or SFR) to/from byte register	4	3(2)	3	2(2)
	WRj, dir8	Direct address (on-chip RAM or SFR) to/from word register	4	4	3	3
	Rm, dir16	Direct address (64K) to/from byte register	5	3(3)	4	2(3)
	WRj, dir16	Direct address (64K) to/from word register	5	4(4)	4	3(4)
	Rm, at WRj	Indirect address (64K) to/from byte register	4	3(3)	3	2(3)
	Rm, at DRk	Indirect address (16M) to/from byte register	4	4(3)	3	3(3)
ADDC/SUBB	A, Rn	Register to/from ACC with carry	1	1	2	2
	A, dir8	Direct address (on-chip RAM or SFR) to/from ACC with carry	2	1(2)	2	1(2)
	A, at Ri	Indirect address to/from ACC with carry	1	2	2	3
	A, #data	Immediate data to/from ACC with carry	2	1	2	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 1 to the number of states. Add 2 if it addresses a Peripheral SFR.
3. If this instruction addresses external memory location, add N+2 to the number of states (N: number of wait states).

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).

Table 26. Summary of Move Instructions (2/3)

Move[1] MOV <dest>, <src> dest opnd ← src opnd						
			Binary Mode		Source Mode	
Mnemonic	<dest>, <src>[2]	Comments	Bytes	States	Bytes	States
MOV	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
	Rn, A	ACC to register	1	1	2	2
	Rn, dir8	Direct address (on-chip RAM or SFR) to register	2	1[3]	3	2[3]
	Rn, #data	Immediate data to register	2	1	3	2
	dir8, A	ACC to direct address (on-chip RAM or SFR)	2	2[3]	2	2[3]
	dir8, Rn	Register to direct address (on-chip RAM or SFR)	2	2[3]	3	3[3]
	dir8, dir8	Direct address to direct address (on-chip RAM or SFR)	3	3[4]	3	3[4]
	dir8, at Ri	Indirect address to direct address (on-chip RAM or SFR)	2	3[3]	3	4[3]
	dir8, #data	Immediate data to direct address (on-chip RAM or SFR)	3	3[3]	3	3[3]
	at Ri, A	ACC to indirect address	1	3	2	4
	at Ri, dir8	Direct address (on-chip RAM or SFR) to indirect address	2	3[3]	3	4[3]
	at Ri, #data	Immediate data to indirect address	2	3	3	4
	DPTR, #data16	Load Data Pointer with a 16-bit constant	3	2	3	2

Notes:
1. Instructions that move bits are in Table 27.
2. Move instructions from the C51 Architecture.
3. If this instruction addresses an I/O Port (Px, x = 0-3), add 1 to the number of states. Add 2 if it addresses a Peripheral SFR.
4. Apply note 3 for each dir8 operand.

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 27. Summary of Bit Instructions

Clear BitCLR <dest>dest opnd ← 0						
Set BitSETB <dest>dest opnd ← 1						
Complement BitCPL <dest>dest opnd ← ∅ bit						
AND Carry with BitANL CY, <src>(CY) ← (CY) ∧ src opnd						
AND Carry with Complement of BitANL CY, /<src>(CY) ← (CY) ∧ ∅ src opnd						
OR Carry with BitORL CY, <src>(CY) ← (CY) ∨ src opnd						
OR Carry with Complement of BitORL CY, /<src>(CY) ← (CY) ∨ ∅ src opnd						
Move Bit to CarryMOV CY, <src>(CY) ← src opnd						
Move Bit from CarryMOV <dest>, CYdest opnd ← (CY)						
Mnemonic	**<dest>, <src>(1)**	**Comments**	**Binary Mode**		**Source Mode**	
			Bytes	**States**	**Bytes**	**States**
CLR	CY	Clear carry	1	1	1	1
	bit51	Clear direct bit	2	2(3)	2	2(3)
	bit	Clear direct bit	4	4(3)	3	3(3)
SETB	CY	Set carry	1	1	1	1
	bit51	Set direct bit	2	2(3)	2	2(3)
	bit	Set direct bit	4	4(3)	3	3(3)
CPL	CY	Complement carry	1	1	1	1
	bit51	Complement direct bit	2	2(3)	2	2(3)
	bit	Complement direct bit	4	4(3)	3	3(3)
ANL	CY, bit51	And direct bit to carry	2	1(2)	2	1(2)
	CY, bit	And direct bit to carry	4	3(2)	3	2(2)
	CY, /bit51	And complemented direct bit to carry	2	1(2)	2	1(2)
	CY, /bit	And complemented direct bit to carry	4	3(2)	3	2(2)
ORL	CY, bit51	Or direct bit to carry	2	1(2)	2	1(2)
	CY, bit	Or direct bit to carry	4	3(2)	3	2(2)
	CY, /bit51	Or complemented direct bit to carry	2	1(2)	2	1(2)
	CY, /bit	Or complemented direct bit to carry	4	3(2)	3	2(2)
MOV	CY, bit51	Move direct bit to carry	2	1(2)	2	1(2)
	CY, bit	Move direct bit to carry	4	3(2)	3	2(2)
	bit51, CY	Move carry to direct bit	2	2(3)	2	2(3)
	bit, CY	Move carry to direct bit	4	4(3)	3	3(3)

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 1 to the number of states. Add 2 if it addresses a Peripheral SFR.
3. 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.

Table 30. Summary of Conditional Jump Instructions (2/2)

Jump if bitJB <src>, rel(PC) ← (PC) + size (instr);
IF [src opnd = 1] THEN (PC) ← (PC) + rel

Jump if not bitJNB <src>, rel(PC) ← (PC) + size (instr);
IF [src opnd = 0] THEN (PC) ← (PC) + rel

Jump if bit and clearJBC <dest>, rel(PC) ← (PC) + size (instr);
IF [dest opnd = 1] THEN
dest opnd ← 0
(PC) ← (PC) + rel

Jump if accumulator is zeroJZ rel(PC) ← (PC) + size (instr);
IF [(A) = 0] THEN (PC) ← (PC) + rel

Jump if accumulator is not zeroJNZ rel(PC) ← (PC) + size (instr);
IF [(A) ≠ 0] THEN (PC) ← (PC) + rel

Compare and jump if not equalCJNE <src1>, <src2>, rel(PC) ← (PC) + size (instr);
IF [src opnd1 < src opnd2] THEN (CY) ← 1
IF [src opnd1 ≥ src opnd2] THEN (CY) ← 0
IF [src opnd1 ≠ src opnd2] THEN (PC) ← (PC) + rel

Decrement and jump if not zeroDJNZ <dest>, rel(PC) ← (PC) + size (instr); dest opnd ← dest opnd -1;
IF [φ (Z)] THEN (PC) ← (PC) + rel

Mnemonic	<dest>, <src>[1]	Comments	Binary Mode[2]		Source Mode[2]	
			Bytes	States	Bytes	States
JB	bit51, rel	Jump if direct bit is set	3	2/5[3][6]	3	2/5[3][6]
	bit, rel	Jump if direct bit of 8-bit address location is set	5	4/7[3][6]	4	3/6[3][6]
JNB	bit51, rel	Jump if direct bit is not set	3	2/5[3][6]	3	2/5[3][6]
	bit, rel	Jump if direct bit of 8-bit address location is not set	5	4/7[3][6]	4	3/6[3]
JBC	bit51, rel	Jump if direct bit is set & clear bit	3	4/7[5][6]	3	4/7[5][6]
	bit, rel	Jump if direct bit of 8-bit address location is set and clear	5	7/10[5][6]	4	6/9[5][6]
JZ	rel	Jump if ACC is zero	2	2/5[6]	2	2/5[6]
JNZ	rel	Jump if ACC is not zero	2	2/5[6]	2	2/5[6]
CJNE	A, dir8, rel	Compare direct address to ACC and jump if not equal	3	2/5[3][6]	3	2/5[3][6]
	A, #data, rel	Compare immediate to ACC and jump if not equal	3	2/5[6]	3	2/5[6]
	Rn, #data, rel	Compare immediate to register and jump if not equal	3	2/5[6]	4	3/6[6]
	at Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	3/6[6]	4	4/7[6]
DJNZ	Rn, rel	Decrement register and jump if not zero	2	2/5[6]	3	3/6[6]
	dir8, rel	Decrement direct address and jump if not zero	3	3/6[4][6]	3	3/6[4][6]

Notes:
1. A shaded cell denotes an instruction in the C51 Architecture.
2. States are given as jump not-taken/taken.
3. If this instruction addresses an I/O Port (Px, x = 0-3), add 1 to the number of states. Add 2 if it addresses a Peripheral SFR.
4. If this instruction addresses an I/O Port (Px, x = 0-3), add 2 to the number of states.

To preserve the secrecy of the encryption key byte sequence, the Encryption Array can not be verified.

Notes:
1. When a MOVC instruction is executed, the content of the ROM is not encrypted. In order to fully protect the user program code, the lock bit level 1 (see Table 33) must always be set when encryption is used.
2. If the encryption feature is implemented, the portion of the on-chip code memory that does not contain program code should be filled with “random” byte values to prevent the encryption key sequence from being revealed.

Signature Bytes

The TSC80251G2D derivatives contain factory-programmed Signature Bytes. These bytes are located in non-volatile memory outside the memory address space at 30h, 31h, 60h and 61h. To read the Signature Bytes, perform the procedure described in section Verify Algorithm, using the verify signature mode (see Table 37). Signature byte values are listed in Table 35.

Table 35. Signature Bytes (Electronic ID)

		Signature Address	Signature Data
Vendor	Atmel	30h	58h
Architecture	C251	31h	40h
Memory	32 kilobytes EPROM or OTPROM	60h	F7h
	32 kilobytes MaskROM or ROMless		77h
Revision	TSC80251G2D derivative	61h	FDh

Programming Algorithm

Figure 6 shows the hardware setup needed to program the TSC87251G2D EPROM/OTPROM areas:

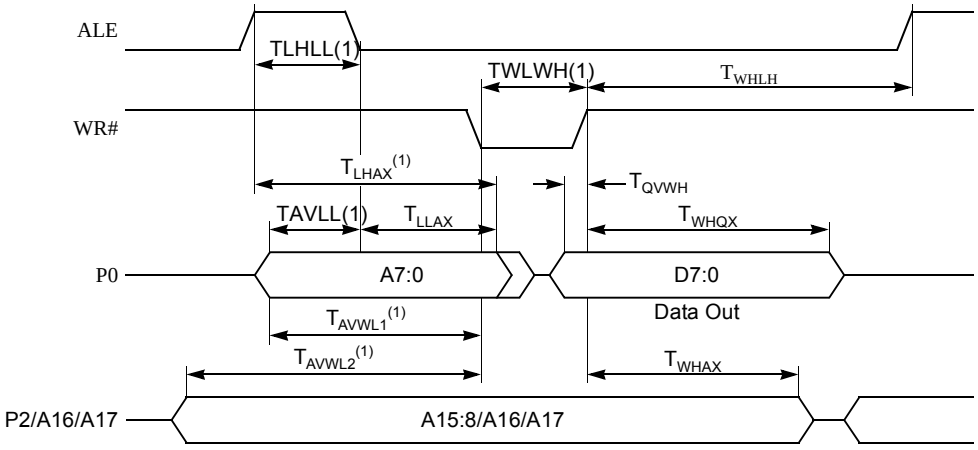
- The chip has to be put under reset and maintained in this state until completion of the programming sequence.
- PSEN# and the other control signals (ALE and Port 0) have to be set to a high level.
- Then PSEN# has to be to forced to a low level after two clock cycles or more and it has to be maintained in this state until the completion of the programming sequence (see below).
- The voltage on the EA# pin must be set to V_{DD}.
- The programming mode is selected according to the code applied on Port 0 (see Table 36). It has to be applied until the completion of this programming operation.
- The programming address is applied on Ports 1 and 3 which are respectively the Most Significant Byte (MSB) and the Least Significant Byte (LSB) of the address.
- The programming data are applied on Port 2.
- The EPROM Programming is done by raising the voltage on the EA# pin to V_{PP}, then by generating a low level pulse on ALE/PROG# pin.
- The voltage on the EA# pin must be lowered to V_{DD} before completing the programming operation.
- It is possible to alternate programming and verifying operation (See Paragraph Verify Algorithm). Please make sure the voltage on the EA# pin has actually been lowered to V_{DD} before performing the verifying operation.

Table 40. Bus Cycles AC Timings; V_{DD} = 2.7 to 5.5 V, T_A = -40 to 85°C

Symbol	Parameter	12 MHz		16 MHz		Unit
		Min	Max	Min	Max	
T_{OSC}	$1/F_{OSC}$	83		62		ns
T_{LHLL}	ALE Pulse Width	72		52		ns(2)
T_{AVLL}	Address Valid to ALE Low	71		51		ns(2)
T_{LLAX}	Address hold after ALE Low	14		6		ns
T_{RLRH}(1)	RD#/PSEN# Pulse Width	163		121		ns(3)
T_{WLWH}	WR# Pulse Width	165		124		ns(3)
T_{LLRL}(1)	ALE Low to RD#/PSEN# Low	17		11		ns
T_{LHAX}	ALE High to Address Hold	90		57		ns(2)
T_{RLDV}(1)	RD#/PSEN# Low to Valid Data		133		92	ns(3)
T_{RHDX}(1)	Data Hold After RD#/PSEN# High	0		0		ns
T_{RHAX}(1)	Address Hold After RD#/PSEN# High	0		0		ns
T_{RLAZ}(1)	RD#/PSEN# Low to Address Float		0		0	ns
T_{RHDZ1}	Instruction Float After RD#/PSEN# High		59		48	ns
T_{RHDZ2}	Data Float After RD#/PSEN# High		225		175	ns
T_{RHLH1}	RD#/PSEN# high to ALE High (Instruction)	60		47		ns
T_{RHLH2}	RD#/PSEN# high to ALE High (Data)	226		172		ns
T_{WHLH}	WR# High to ALE High	226		172		ns
T_{AVDV1}	Address (P0) Valid to Valid Data In		289		160	ns(2)(3)
T_{AVDV2}	Address (P2) Valid to Valid Data In		296		211	ns(2)(3)
T_{AVDV3}	Address (P0) Valid to Valid Instruction In		144		98	ns(3)
T_{AXDX}	Data Hold after Address Hold	0		0		ns
T_{AVRL}(1)	Address Valid to RD# Low	111		64		ns(2)
T_{AVWL1}	Address (P0) Valid to WR# Low	111		64		ns(2)
T_{AVWL2}	Address (P2) Valid to WR# Low	158		116		ns(2)
T_{WHQX}	Data Hold after WR# High	82		66		ns
T_{QVWH}	Data Valid to WR# High	135		103		ns(3)
T_{WHAX}	WR# High to Address Hold	168		125		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).

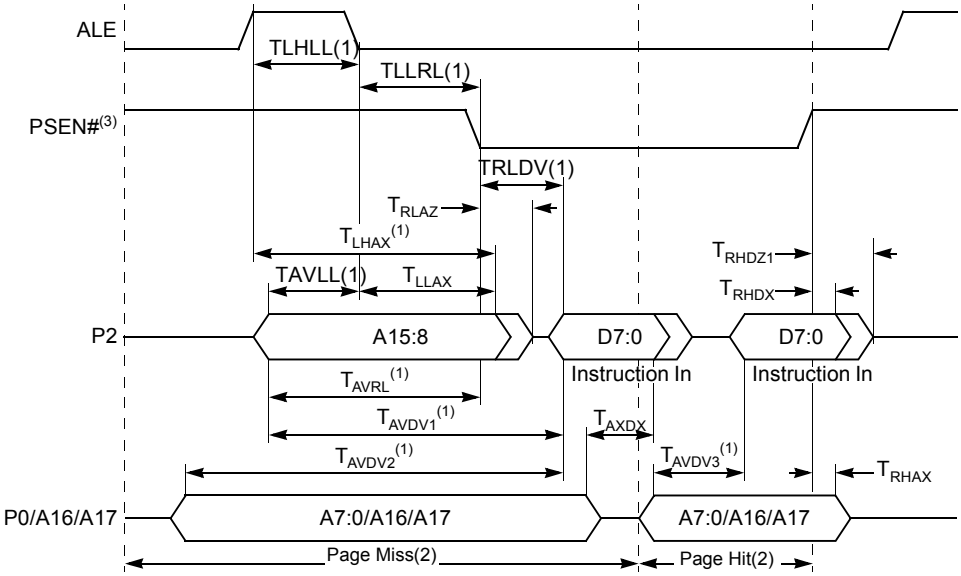
Figure 10. External Bus Cycle: Data Write (Non-Page Mode)



Note: 1. The value of this parameter depends on wait states. See Table 39 and Table 40.

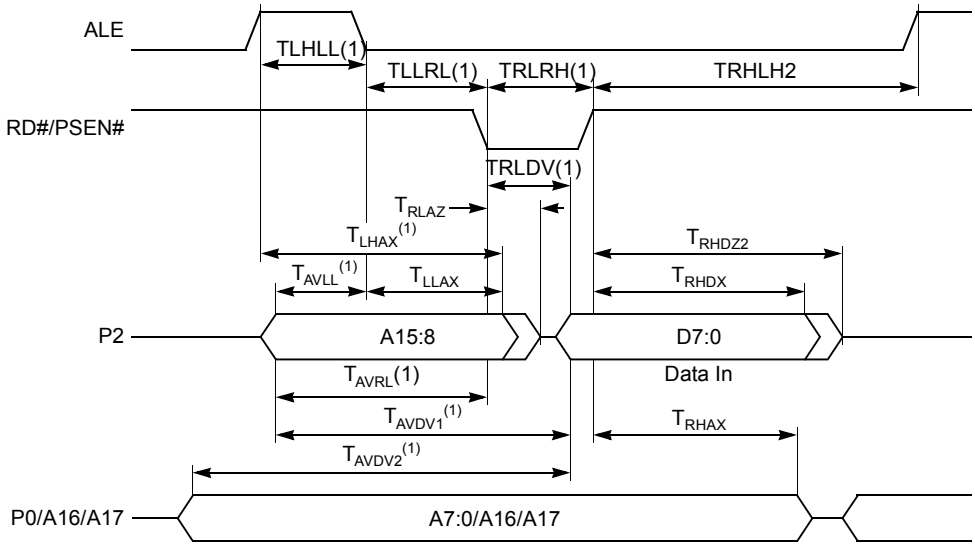
Waveforms in Page Mode

Figure 11. External Bus Cycle: Code Fetch (Page Mode)



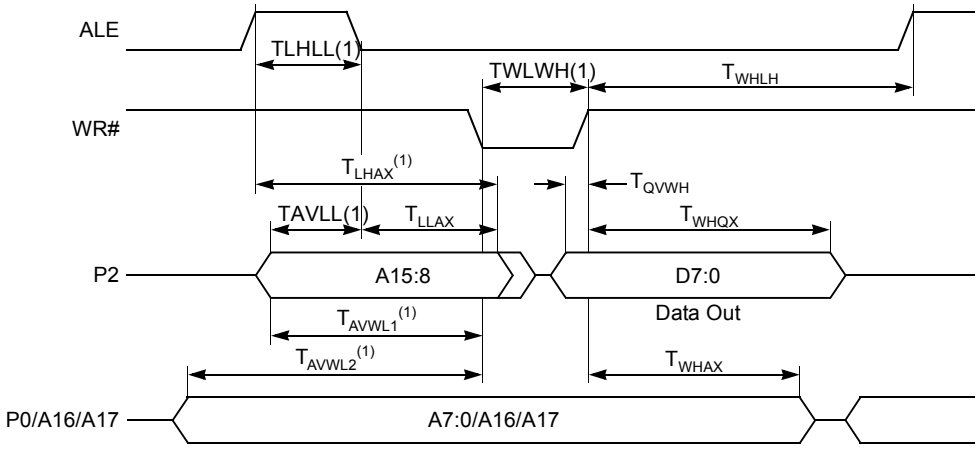
Note: 1. The value of this parameter depends on wait states. See Table 39 and Table 40.
2. A page hit (i.e., a code fetch to the same 256-byte “page” as the previous code fetch) requires one state ($2 \cdot T_{OSC}$);
a page miss requires two states ($4 \cdot T_{OSC}$).
3. During a sequence of page hits, PSEN# remains low until the end of the last page-hit cycle.

Figure 12. External Bus Cycle: Data Read (Page Mode)



Note: 1. The value of this parameter depends on wait states. See Table 39 and Table 40.

Figure 13. External Bus Cycle: Data Write (Page Mode)



Note: 1. The value of this parameter depends on wait states. See Table 39 and Table 40.

AC Characteristics - Real-Time Synchronous Wait State

Definition of Symbols

Table 41. Real-Time Synchronous Wait Timing Symbol Definitions

Signals	
C	WCLK
R	RD#/PSEN#
W	WR#
Y	WAIT#

Conditions	
L	Low
V	Valid
X	No Longer Valid

Timings

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

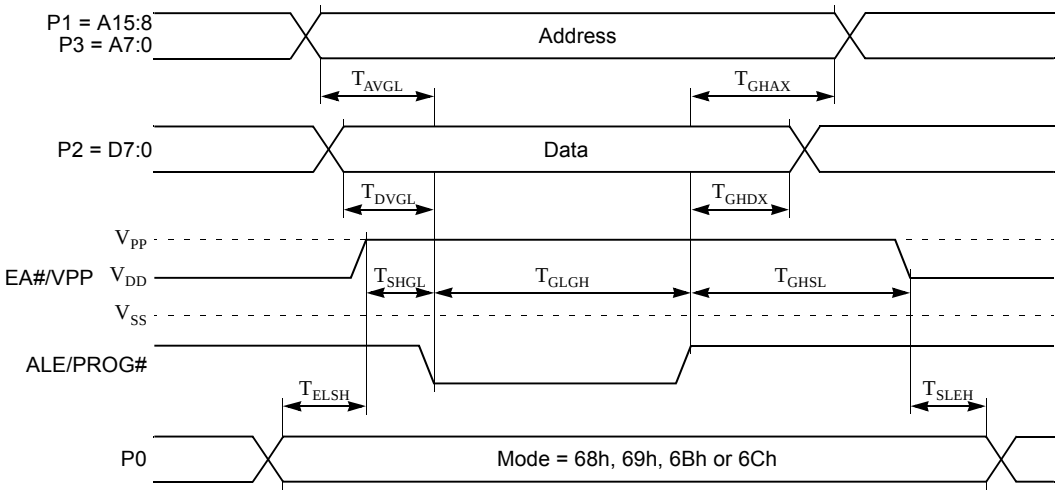
Symbol	Parameter	Min	Max	Unit
T_{OSC}	XTAL1 Period	83.5	250	ns
T_{AVGL}	Address Setup to PROG# low	48		T_{OSC}
T_{GHAX}	Address Hold after PROG# low	48		T_{OSC}
T_{DVGL}	Data Setup to PROG# low	48		T_{OSC}
T_{GHDX}	Data Hold after PROG#	48		T_{OSC}
T_{ELSH}	ENABLE High to V_{PP}	48		T_{OSC}
T_{SHGL}	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

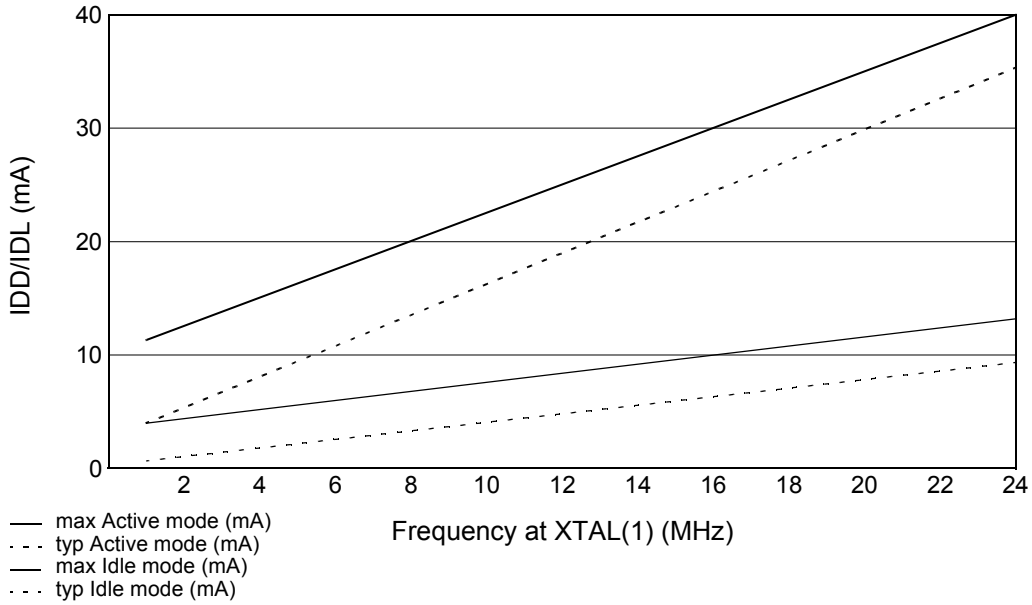
Ports 1-3 15 mA

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: $F_{OSC} = F_{XTAL}$.

Low Voltage Versions - Commercial & Industrial

Table 56. DC Characteristics; V_{DD} = 2.7 to 5.5 V, T_A = -40 to +85°C

Symbol	Parameter	Min	Typical[4]	Max	Units	Test Conditions
V_{IL}	Input Low Voltage (except EA#, SCL, SDA)	-0.5		$0.2 \cdot V_{DD}$ - 0.1	V	
V_{IL1}[5]	Input Low Voltage (SCL, SDA)	-0.5		$0.3 \cdot V_{DD}$	V	
V_{IL2}	Input Low Voltage (EA#)	0		$0.2 \cdot V_{DD}$ - 0.3	V	
V_{IH}	Input high Voltage (except XTAL1, RST, SCL, SDA)	$0.2 \cdot V_{DD}$ + 0.9		V_{DD} + 0.5	V	
V_{IH1}[5]	Input high Voltage (XTAL1, RST, SCL, SDA)	$0.7 \cdot V_{DD}$		V_{DD} + 0.5	V	
V_{OL}	Output Low Voltage (Ports 1, 2, 3)			0.45	V	I_{OL} = 0.8 mA[1][2]
V_{OL1}	Output Low Voltage (Ports 0, ALE, PSEN#, Port 2 in Page Mode during External Address)			0.45	V	I_{OL} = 1.6 mA[1][2]
V_{OH}	Output high Voltage (Ports 1, 2, 3, ALE, PSEN#)	$0.9 \cdot V_{DD}$			V	I_{OH} = -10 µA[3]
V_{OH1}	Output high Voltage (Port 0, Port 2 in Page Mode during External Address)	$0.9 \cdot V_{DD}$			V	I_{OH} = -40 µA
V_{RET}	V_{DD} data retention limit			1.8	V	
I_{IL0}	Logical 0 Input Current (Ports 1, 2, 3 - AWAIT#)			- 50	µA	V_{IN} = 0.45 V
I_{IL1}	Logical 1 Input Current (NMI)			+ 50	µA	$V_{IN} = V_{DD}$
I_{LI}	Input Leakage Current (Port 0)			± 10	µA	0.45 V < V_{IN} < V_{DD}
I_{TL}	Logical 1-to-0 Transition Current (Ports 1, 2, 3)			- 650	µA	V_{IN} = 2.0 V
R_{RST}	RST Pull-Down Resistor	40	110	225	kΩ	
C_{IO}	Pin Capacitance		10		pF	T_A = 25°C
I_{DD}	Operating Current		4 8 9 11	8 11 12 14	mA	5 MHz, V_{DD} < 3.6 V 10 MHz, V_{DD} < 3.6 V 12 MHz, V_{DD} < 3.6 V 16 MHz, V_{DD} < 3.6 V
I_{DL}	Idle Mode Current		0.5 1.5 2 3	1 4 5 7	mA	5 MHz, V_{DD} < 3.6 V 10 MHz, V_{DD} < 3.6 V 12 MHz, V_{DD} < 3.6 V 16 MHz, V_{DD} < 3.6 V
I_{PD}	Power-Down Current		1	10	µA	V_{RET} < V_{DD} < 3.6 V

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 1-315 mA

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

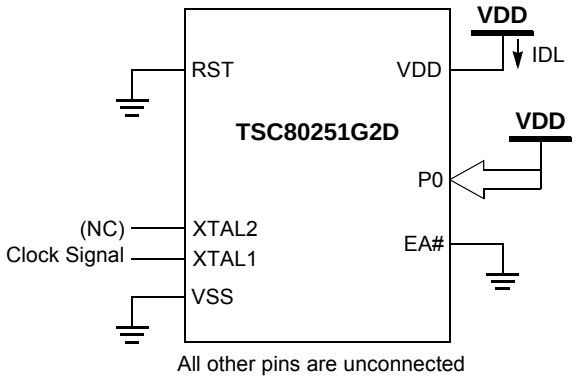
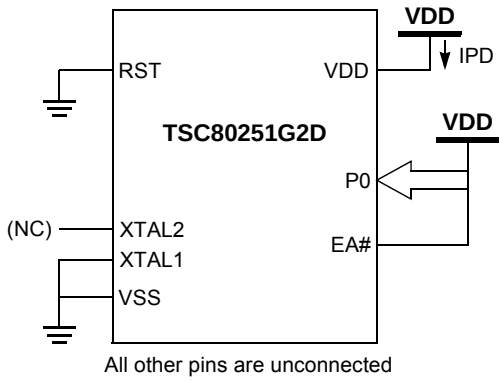


Figure 32. I_{PD} Test Condition, Power-Down Mode



VQFP 44 (10x10) - Mechanical Outline

Figure 37. Shrink Quad Flat Pack (Plastic)

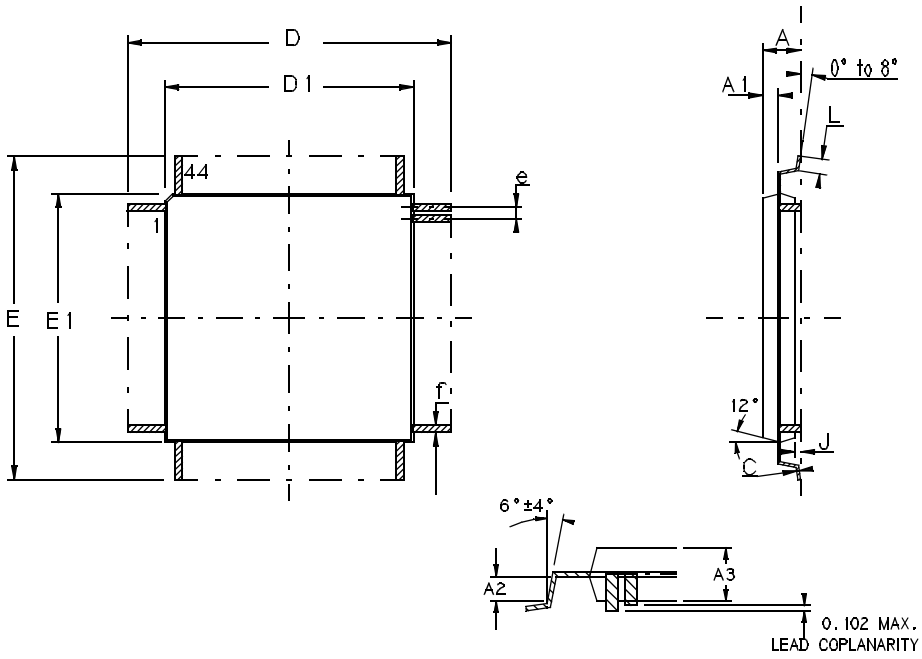


Table 61. VQFP Package Size

	MM		Inch	
	Min	**Max**	**Min**	**Max**
A	-	1.60	-	.063
A1	0.64 REF		.025 REF	
A2	0.64 REF		.025REF	
A3	1.35	1.45	.053	.057
D	11.90	12.10	.468	.476
D1	9.90	10.10	.390	.398
E	11.90	12.10	.468	.476
E1	9.90	10.10	.390	.398
J	0.05	-	.002	6
L	0.45	0.75	.018	.030
e	0.80 BSC		.0315 BSC	
f	0.35 BSC		.014 BSC	

Part Number[1]	ROM	Description
Low Voltage Versions 2.7 to 5.5 V		
TSC251G2Dxxx-L16CB	32K MaskROM	16 MHz, Commercial 0° to 70°C, PLCC 44
TSC251G2Dxxx-L16CE	32K MaskROM	16 MHz, Commercial 0° to 70°C, VQFP 44
AT251G2Dxxx-SLSUL	32K MaskROM	16 MHz, Industrial & Green, PLCC 44
AT251G2Dxxx-RLTUL	32K MaskROM	16 MHz, Industrial & Green, VQFP 44

Note: 1. xxx: means ROM code, is Cxxx in case of encrypted code.