BL35P02 DATASHEET

8-bit OTP MCU

V1.0 (2010-4-6)

Shanghai Belling Co., Ltd.

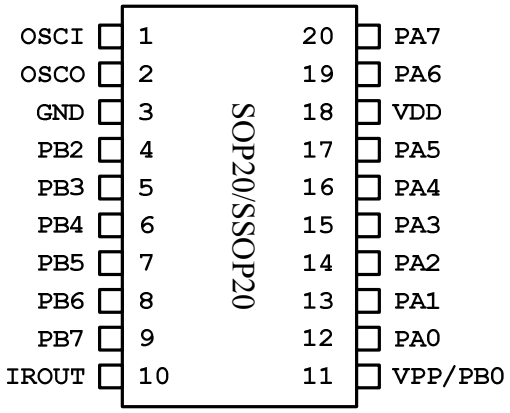
1. General Description

BL35P02 is a single-chip 8-bit micro-controller. This device integrates a HC05 8-bit CPU core, RAM, ROM, timer, programmable input/output pins and carrier synthesizer. When in standby status, system will stop oscillator and remain low power consumption. The BL35P02 is suitable for infrared remote control transmitter application.

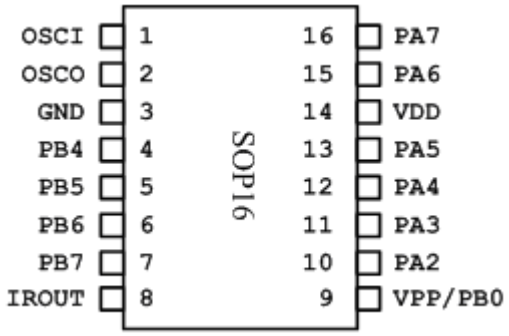
2. Features

- 8-bit CISC core（compatible with Motorola HC05）
- 14 CMOS Bi-directional I/O pins and 1 CMOS input pin
- One 8-bit timer
- 9 keyboard interruption
- One infrared remote output, 8 kinds of carrier wave selected (1/3 duty)
- Crystal/Ceramic oscillator(325K-8MHz)
- Low power (Standby current less than 1uA@3V)
- 32*8 bits RAM (including stack)
- 2K*8 bits OTP ROM
- OTP data encrypted
- Operation Voltage:2.0-5.5V
- Package: SOP20(300mil)/SSOP20(200mil)/SOP16(150mil)

3. Pin Configuration



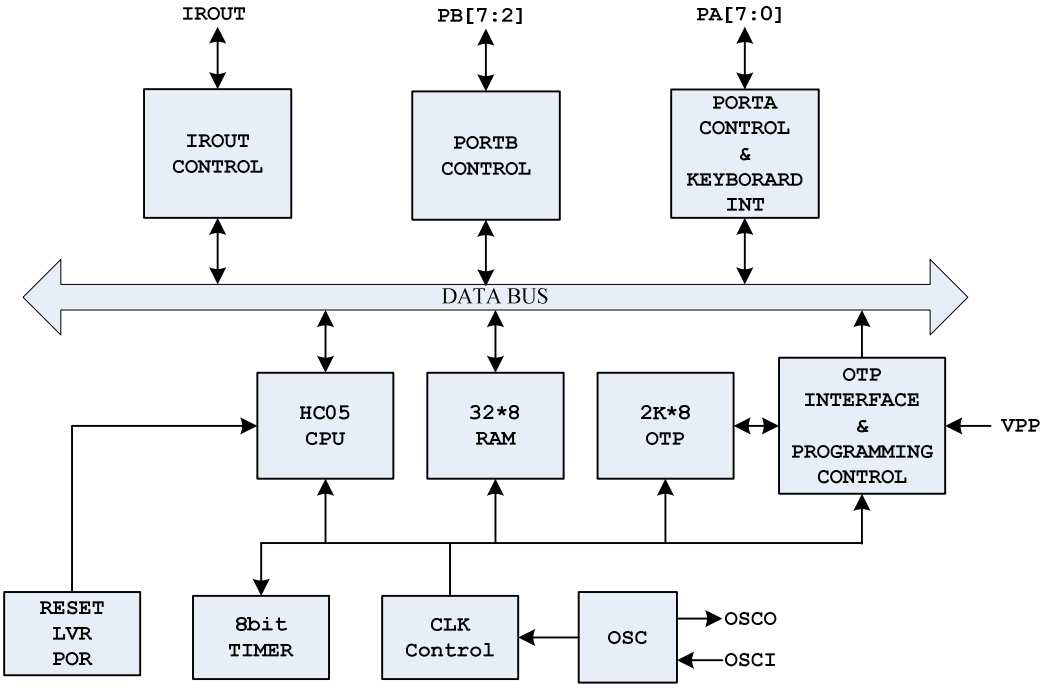
SOP20（300mil）
SSOP20（200mil）



SOP16（150mil）

Name	In/Out	Description
OSCI	INPUT	The OSCI and OSCO pins are the connections for the on-chip oscillator
OSCO	OUTPUT	
GND	SOURCE	Ground
VDD	SOURCE	Power
IROUT	OUTPUT	Infrared remote output
VPP/PB0	INPUT	High voltage power supply as OTP programming ; normal input, keyboard interrupt input port
PB2-PB7	I/O	Bit-programmable I/O port for Schmitt trigger input or push-pull output. Pull-up resistors are assignable by software.
PA0-PA7	I/O	Bit-programmable I/O port for Schmitt trigger input or push-pull output. Pull-up resistors are assignable by software. Keyboard interrupt input port

4. Block Diagram



5. Electrical Characteristics

5.1 Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Operating Voltage	Vdd	-0.3~6.5	V
Input Voltage	VIN	Vss-0.3～Vdd+0.3	V
Operating Ambient Temperature	TA	-20～85	℃
Storage Temerature	Tstg	-65～150	℃

5.2 DC Electrical Characteristics (VDD=3.0V, GND=0V,T=25℃, unless otherwise specified)

Parameter	Symbol	PIN	Condition	Min.	Typ.	Max.	Unit
Operating Voltage	VDD			2.0	3.0	5.5	V
Output High Voltage driving current	I_{oh}	PA7~PA0 PB7~PB2 IROUT	V_{oh}=2.7V	3	5		mA
Output Low Voltage Sink current	I_{ol1}	PA7~PA0 PB7~PB2	V_{ol}=0.3V	10	14		mA
	I_{ol2}	IROUT	V_{ol}=0.3V	20	22		mA
Input High Voltage	V_{ih}	PA7~PA0 PB7~PB2 PB0		0.7Vdd		Vdd	V
Input Low Voltage	V_{il}	PA7~PA0 PB7~PB2 PB0		0		0.2Vdd	V
LVR Voltage	V_{LVR}		0-40℃	1.15	1.40	1.65	V
STOP Current	I_{st}	VDD	STOP Mode		0.1	1	uA
Pull-up resistor	R_p	PA7~PA0 PB7~PB2		10	25	50	Kohm

5.3 AC Electrical Characteristics (VDD=3.0V, GND=0V,T=25℃)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Oscillator Frequency	F_{osc}	325K		8M	Hz
Oscillator Start time	T_{oxov}			20	ms

6. Function Description

6.1 Instructions

See Section 7 for details on the instructions.

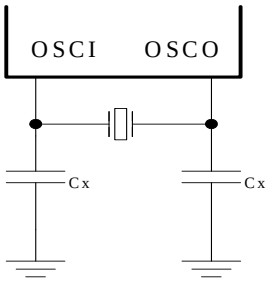
6.2 Address Spaces

$0000-$000F：Control registers
$0010-$00DF：Reserved
$00E0-$00FF：RAM
$0100-$17FF：Reserved
$1800-$1FFF：OTP ROM

6.3 Crystal Oscillator

6.3.1 High frequency Oscillator

Simplified external crystal/ceramic oscillator circuits are shown in Figure 6.3.1.1. An external crystal or ceramic oscillation source provides 355KHz~8MHz. The load capacitor Cx which values used in the oscillator circuit design should include all stray capacitance is necessary, but frequency is more than 3.5MHz, Cx can be removed. The crystal and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion.



Oscillator

Figure 6.3.1.1

Frequency	Value of Cx
8MHz	0/15p
4MHz	0/15p/30p
3.64MHz	0/15p/30p

6.3.2 455KHz Oscillator

Using 455KHz crystal/ceramic oscillation is shown Figure 6.3.2.1，generally OSCI/OSCO must be connected external 200pF capacitor. Otherwise OSCO can be connected 2KΩ to 5KΩ resistor that can be used by carbon film resistor.

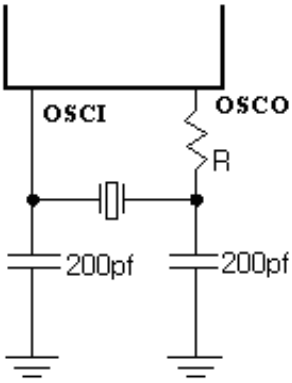


Figure 6.3.2.1

6.4 I/O Ports

The MCU provides 14 Bi-directional I/O pins (PA7-PA0, PB7-PB2) and 1 input pin (PB0). The individual bits in these ports are programmable as either inputs or outputs under software control by the Data Direction Registers (DDRx). All port pins each has an associated 25KΩ pull-up resistor, which can be connected/disconnected under software control. Each Port pin is controlled by the corresponding bits in a Data Direction Register and a Data Register as shown in Figure 6.4.1,

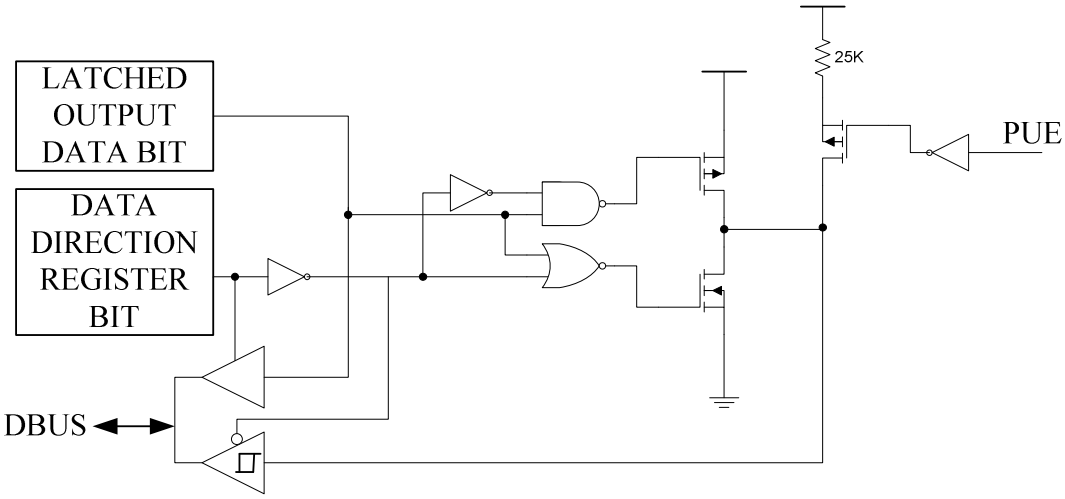


Figure 6.4.1

The functions of the I/O pins are summarized as follows:

Read/Write	DDRx	Function
Write	0	The I/O pins is in input mode. Data is written into the output data latch
Write	1	Data is written into the output data latch and output to the I/O pin.
Read	0	The state of the I/O pin is read.
Read	1	The I/O pin is in an output mode. The output data latch is read.

Port A is configured for use as keyboard interrupts when the KBIE bit is set in the Miscellaneous Control Register (MCR). Individual keyboard interrupt port pins are also maskable by setting corresponding bits in the Keyboard Interrupt Mask Register (KBIM). When the KBEx bit is set, the corresponding Port A pin will the configured as an input pin, regardless of the DDR setting, and a 25KΩ pull-up resistor is connected to the pin. See Section 6.7.1 for details on the keyboard interrupts.

When Port B is used input port, it has an associated 25KΩ pull-up resistor, which can be connected/disconnected under software control. As Port B is used output port, it has not an associated 25 KΩ pull-up resistor. PB2's pull-up resistor is controlled by PBP2 of MCR, PB3's pull-up resistor is controlled by PBP3 of MCR. PB4~PB7's pull-up resistors are controlled by PBP of MCR.

When OTP is programming, PB0 is used high voltage input, normally it is used input port that has no pull-up resistor, and configured for use as a keyboard interrupt when the KBEB0 is set in DDRB. See Section 6.7.1 for details on the keyboard interrupts.

6.5 Timer

The BL35P02 timer block diagram is shown in Figure 6.5.1. The timer contains a single 8-bit software programmable count-down counter with a 7-bit software selectable prescaler. The counter may be preset under software control and decrements towards zero. When the counter decrements to zero, the timer interrupt flag

(TIF bit in Timer Control Register, TCR) is set. Once timer interrupt flag is set, an interrupt is generated to the CPU only if the TIM bit in the TCR and 1-bit in the CCR are cleared. When an interrupt is recognized, after completion of the current instruction, the processor proceeds to store the appropriate registers on the stack and then fetches the timer interrupt vector from locations $1FF6 and $1FF7. See Section 6.7.2 for details on the timer interrupts.

The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle and do not change during the read. The timer interrupt flag remains set until cleared by the software. If a write occurs before the timer interrupt is served, the interrupt is lost. The timer interrupt flag may also be sued as a scanned status bit in a non-interrupt mode of operation.

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0,1,2(PR0,PR1,PR2) of TCR are programmed to choose the appropriate prescaler output which is used as the 8-bit counter clock input. The processor cannot write into or read from the prescaler; however, its contents can be cleared to all zeros by writing to the PRER bit in the TCR. This will allow for truncation-free counting.

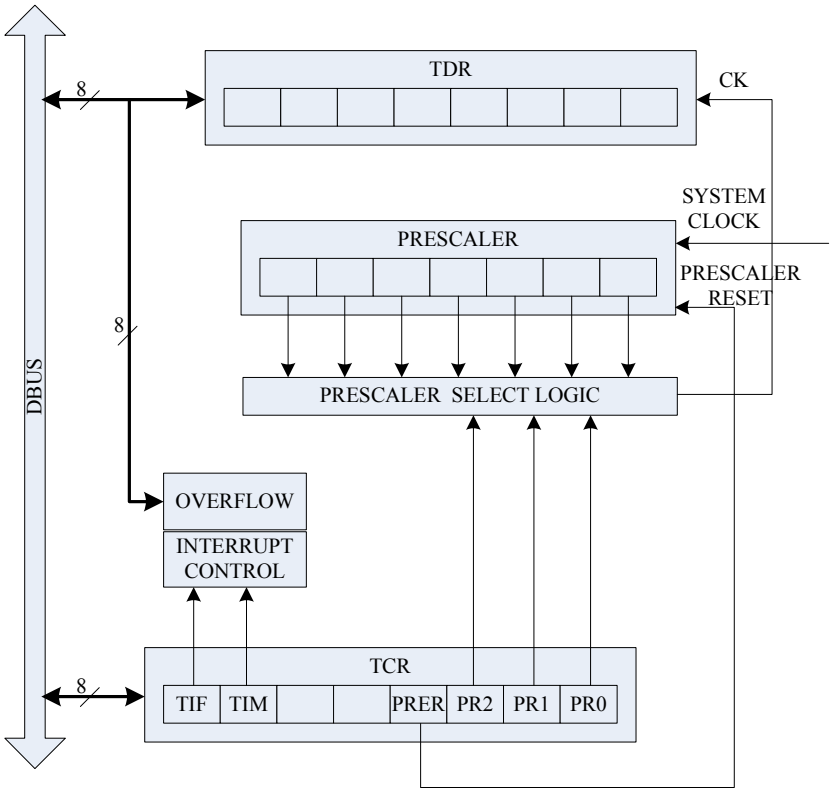


Figure 6.5.1

6.6 Remote Control Carrier Synthesizer

The device has a built carrier synthesizer for infrared or RF remote control circuits. The carrier's duty is 1/3. The carrier synthesizer can be programmed in several different prescaler ratios by setting FC[2:0] of OTP's OPTION BIT. IROUT of the remote control carrier synthesizer output is shown in Figure 6.6.1.

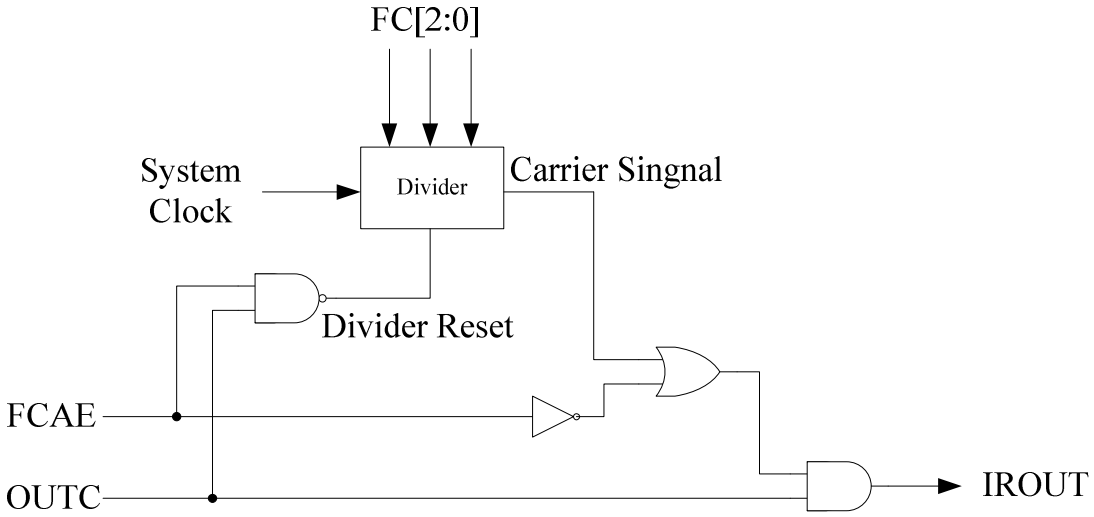
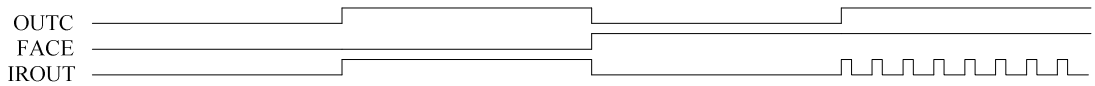


Figure 6.6.1

FCAE and OUTC of MCR are programmed to control the carrier has remote code or not. Either FCAE or OUTC is clear, the carrier prescaler will be reset to make the first remote code with the carrier full. The waves of IROUT, FCAE and OUTC are shown as follows:



The carrier of IROUT is based on the system clock that is half of oscillator frequency. It is programmed in eight different prescaler ratios by setting FC[2:0] of OTP's OPTION that is shown as follows:

FC[2:0]	Prescaler Divide Ratio of System Clock	Oscillator Frequency	The Carrier Frequency of IROUT
000	6	445KHz	37.91K
001	36	4MHz	55.56K
010	50	4MHz	40.00K
011	53	4MHz	37.74K
100	56	4MHz	35.71K
101	61	4MHz	32.78K
110	64	4MHz	31.25K
111	74	4MHz	27.03K

6.7 Interrupts

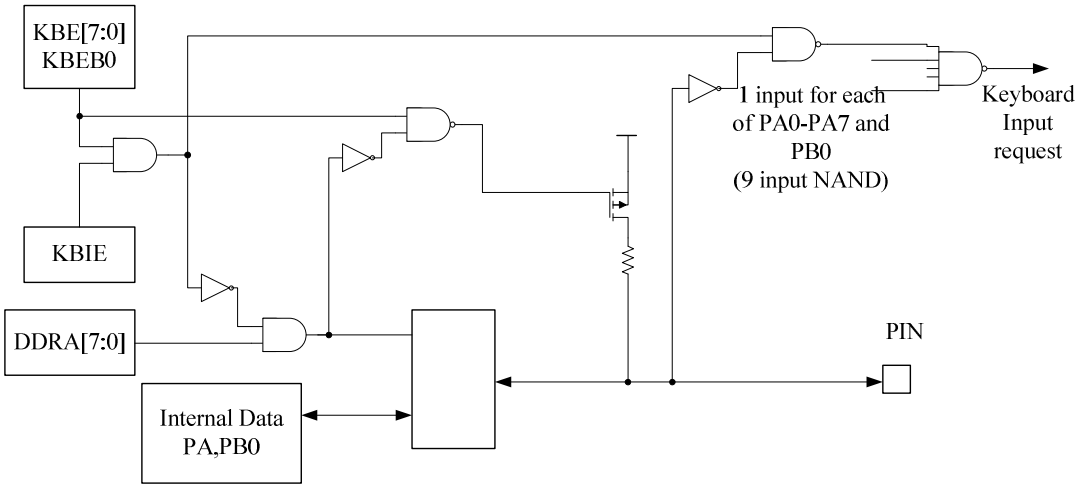
The BL35P02 MCU can be interrupted by different sources including two maskable hardware interrupts Keyboard interrupt (KBI) and Timer Overflow interrupt (TMI) and one non-maskable software interrupt Software interrupt(SWI). If the interrupt mask bit (I-bit) in the Condition Code Register (CCR) is set, all maskable interrupts are disabled. Clearing the I-bit enables interrupts. The software interrupt (SWI) is an executable instruction and a non-maskable interrupt: it is execute regardless of the state of the I-bit in the CCR. If the I-bit is zero (interrupt enabled), SWI is executed after interrupts that were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The SWI interrupt service routine address is specified by the contents of locations $1FFC and $1FFD

6.7.1 Keyboard Interrupt (KBI)

Keyboard interrupt function is associated with Port A pins and PB0 pin. The keyboard interrupt function is enabled by setting the keyboard interrupt enable bit KBIE (bit 7 of MCR at $0C) and the individual enable bits KBE0-KBE7 (bits 0-7 KBIM at $0B) and KBEB0 (bit 0 of DDRB). When the KBEx bit is set, the corresponding Port A pin will be configure as an input pin, regardless of the DDR setting, and a 25kΩ pull-up resistor is connected to the pin, as shown in Figure 6.7.1.1. When a high to low transition is sensed on the pin, a keyboard interrupt will be generated. An interrupt to the CPU will be generated if the I-bi in the CCR is cleared.

The keyboard interrupt flag should be cleared in the interrupt service routine (by writing a “1” to KBIC bit in the MCR at $0C) after the key is debounced. Deboncing will avoid spurious false triggering.

The keyboard interrupt is negative-edge sensitive only, and the interrupt service routine is specified by the contents in $1FF4-$1FF5.



Notice：PB0 has on pull-up resistor

Figure 6.7.1.1

6.7.2 Timer Interrupt

The timer interrupt is generated by the 8-bit timer when a timer overflow has occurred. The interrupt enable and flag for the timer interrupt are located in the Timer Control Register (TCR).

(1) Timer Interrupt Mask (TIM). When TIM is equal to “1”, Timer interrupt is disabled. When TIM is equal to “0”, Timer interrupt is enabled.

(2) Timer Interrupt Flag (TIF). When TIF is equal to “1”, A timer interrupt (timer overflow) has occurred. When TIF is equal to “0”, A timer interrupt (timer overflow) has not occurred.

The I-bit in the CCR must be cleared in order for the timer interrupt to be processed. The interrupt will vector to the interrupt service routine at the address specified by the contents in $1FF6-$1FF7.

6.7.3 Interrupts Process

Interrupts cause the processor to save the register contents on the stack and to set the interrupt mask (I-bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume.

Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete. The current instruction is the on already fetched and being operated on. When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I-bit clear) the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. The relative priority of all the possible sources is shown

as follows:

Interrupt	Vector Address	Priority
KBI	$1FF4:$1FF5	Lowest
TMI	$1FF6:$1FF7	
SWI	$1FFC:$1FFD	
RESET	$1FFE:$1FFF	Highest

6.8 Low Power Modes

The BL35p02 has two low-power modes. The WAIT and STOP instructions provide two modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator.

6.8.1 STOP Mode

Execution of the STOP instruction places the MCU in its power consumption mode. In the STOP mode the internal oscillator is turned off, halting all internal processing.

When the CPU enters STOP mode the I-bit in the CCR is cleared automatically, All other registers and memory contents remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of the STOP mode only by a KBI interrupt or an externally generated reset. When exiting the STOP mode the internal oscillator will resume after a pre-defined number of internal processor clock cycles, due to oscillator stabilization.

In STOP mode the current of BL35P02 is less than 1uA.

6.8.2 WAIT Mode

The WAIT instruction places the MCU in a low-power mode, but consumes more power than the STOP mode, In the WAIT mode the internal processor clock is halted, suspending all processor and internal bus activities. Other Internal clocks remain active, permitting interrupts to be generated from the Timer. The Timer may be used to generate a periodic exit from the WAIT mode or in conjunction with the external Timer pin, on the occurrence of external events. Execution of the WAIT instruction automatically clears the I-bit in the CCR, so that the KBI interrupt, Timer interrupt or externally generated reset can “wake” the MCU. All other registers, memory, and input/output lines remain in their previous states.

In WAIT mode the current of BL35P02 is less than 100uA @3V.

6.9 Control Registers Summary

A summary of all Control Registers is shown as follows:

Register Name	Address	R/W	State on reset
PA	$00	R/W	0000 0000
PB	$01	R/W	0000 00-0
DDRA	$04	R/W	0000 0000
DDRB	$05	R/W	0000 00-0
TDR	$08	R/W	uuuu uuuu
TCR	$09	R/W	01-- 0100
KBIM	$0B	R/W	0000 0000

MCR	$0C	R/W	00-0 0000

Notice:

-: is undefined;

u: is unaffected.

PA ($00): Port A Data Registers

.7-.0 PA[7:0]

When a Port A pin is programmed as an output the state of the corresponding data register bit determines the state of the output pin.

When a Port A pin is programmed as an input, a read of the Port A Data Register will return the logic state of the corresponding Port A pin.

DDRA($04): Port A Data Direction Registers

.7-.0 DDRA[7:0]

Port A pin may be programmed as an input or output by clearing or setting the corresponding bit int DDRA.

0 (clear) - Port A pin is used as an input

1 (set) - Port A pin is used as output

PB ($02): Port B Data Registers

.7-.2,.0 PB[7:2,0]

When a Port B pin is programmed as an output the state of the corresponding data register bit determines the state of the output pin.

When a Port B pin is programmed as an input, a read of the Port B Data Register will return the logic state of the corresponding Port B pin.

DDRB($05): Port B Data Direction Registers

.7-.2 DDRB[7:2]

Port B pin may be programmed as an input or output by clearing or setting the corresponding bit int DDRA.

0 (clear) - Port A pin is used as an input

1 (set) - Port A pin is used as output

.0 KBEB0 – PB0 Keyboard Interrupt Enable

KBEB0 is a keyboard Interrupt Enable bit of PB0 pin。

0 (clear) – Keyboard interrupt of PB0 pin disabled.

1 (set) - Keyboard interrupt of PB0 pin enabled. PB0 has no pull-up resistor.

TDR($08): Timer Data Register

The TDR is a read/write register which contains the current value of the 8-bit count-down timer counter when read. Reading this register does not disturb the counter operation.

TCR($09): Timer Control Register

.7 TIF – Timer Interrupt Flag

0 (clear) – The timer has not reached a count of zero.

1 (set) - The timer has reached a count of zero.

The timer interrupt flag is set when the 8-bit counter decrements to zero. This bit is cleared on reset, or by writing a “0” to the TIF bit.

.6 TIM – Timer Interrupt Mask

0 (clear) – Timer interrupt request to the CPU is not masked (enable).

1 (set) – Timer interrupt request to the CUP is masked (disable).

A reset sets this bit to one; it must then be cleared by software to enable the timer interrupt to the CPU. This timer interrupt mask only masks timer interrupt request to the CPU, and does not affect counting of the 8-bit counter or the setting of TIF.

.3 PRER – PREscaler Reset

Writing a "1" to this write-only bit will reset the prescaler to zero, which is necessary for any new counts set by writing to the Timer Data Register. This bit always reads as zero, and is not affected by reset.

.2-.0 PR[2:0]

These three bits enable the program to select the division ratio of the prescaler. On reset, these three bits are set to "100", which corresponds to a division ratio of 16.

PR2	PR1	PR0	Divide Ration
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

KBIM($0B): Keyboard Interrupt Mask Register

.7-.0 KBE[7:0]

The Keyboard Interrupt Mask Register (KBIM) masks individual keyboard interrupt pins and setting of the internal pull-up resistors on Port A.

KBEi – PAi Keyboard Interrupt Enable

0 (clear) – Keyboard interrupt for PAi pin is masked. Any transitions on PAi will not set any flags.

1 (set) – Keyboard interrupt enabled for PAi. A 25KΩ internal pull-up resistor is connected. High to low transition on PAi will cause a keyboard interrupt.

MCR($0C): Miscellaneous Control Register

.7 KBIE – Keyboard Interrupt Enable

0 (clear) - Keyboard interrupts master disabled.

1 (set) – keyboard interrupts master enabled.

On reset, KBIE bit is clear to "0". KBIE and KBEi control the master enable for the keyboard interrupts.

.6 KBIC – KeyBoard Interrupt Clear

0 (clear) – Writing a "0" has no effect.

1 (set) – writing a "1" clears the keyboard interrupt latch.

On reset, KBIC bit is clear to "0". This is a write-only bit and always read as "0".

.5 reserved

.4 PBP – PB7:PB4 Pull-up

0 (clear) – No pull-up resistor is connected to the inputs of PB7-PB4.

1 (set) – The internal 25KΩ pull-up resistors are connected to the inputs of PB7-PB4

.3 PBP3 – PB3 Pull-up

0 (clear) – No pull-up resistor is connected to the inputs of PB3.

1 (set) – The internal 25KΩ pull-up resistor is connected to the inputs of PB3

.2 PBP2 – PB2 Pull-up

0 (clear) – No pull-up resistor is connected to the inputs of PB2.

1 (set) – The internal 25KΩ pull-up resistor is connected to the inputs of PB2

.1 OUTC

0 (clear) – IROUT output logic 0

1 (clear) – IROUT output logic 1

.0 FCAE

0 (clear) - IROUT output without carrier

1 (set) – IROUT output with carrier

6.10 OPTION BIT

OPTION BIT (OPBIT) is a special Byte in OTP ROM and used to config some initial functions for the device. OPBIT is set when OTP is programming.

.7 ENCR

0: OTP read protection

1: OTP can be read

.6-.3 Reserved

.2-.0 FC[2:0]

FC[2:0]	Carrier Divide Ratio	Carrier Frequency @ Oscillator frequency
000	Fsys/6	38KHz@455KHz OSC
001	Fsys/36	56KHz@4MHz OSC
010	Fsys/50	40KHz@4MHz OSC
011	Fsys/53	38KHz@4MHz OSC
100	Fsys/56	36KHz@4MHz OSC
101	Fsys/61	33KHz@4MHz OSC
110	Fsys/64	31.5KHz@4MHz OSC
111	Fsys/74	27KHz@4MHz OSC

7. Instruction Set

7.1 Addressing Modes

The addressing modes define the manner in which an instruction is to obtain the data required for its execution. There are 8 modes:

1) Inherent
2) Immediate
3) Direct
4) Extended
5) Indexed, no offset
6) Indexed, 8-bit offset
7) Indexed, 16-bit offset
8) Relative

7.1.1 Inherent Addressing Mode

In inherent addressing mode, all information required for the operation is already inherently known to the CPU, and no external operand from memory or from the program is needed. The operands, if any, are only the index register and accumulator, and are always 1-byte instructions.

7.1.2 Immediate Addressing Mode

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. This mode is used to hold a value or constant which is known at the time the program is written and which is not changed during program execution. These are 2-byte instructions, one for the opcode and one for the immediate data byte.

7.1.3 Direct Addressing Mode

The direct addressing mode is similar to the extended addressing mode except the upper byte of the operand address is assumed to be $00. Thus, only the lower byte of the operand address needs to be included in the instruction. Direct addressing allows you to efficiently address the lowest 256 bytes in memory. This area of memory is called the direct page and includes on-chip RAM and I/O registers. Direct addressing is efficient in both memory and time. Direct addressing mode instructions are usually two bytes, one for the opcode and one for the low-order byte of the operand address.

7.1.4 Extended Addressing Mode

In the extended addressing mode, the address of the operand is contained in the two bytes following the opcode. Extended addressing references any location in the MCU memory space including I/O, RAM, ROM and EPROM. Extended addressing mode instructions are three bytes, one for the opcode and two for the address of the operand.

7.1.5 Indexed, No Offset Addressing Mode

In the indexed, no-offset addressing mode, the effective address of the instruction is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte.

7.1.6 Indexed, 8-bit Offset Addressing Mode

In the indexed, 8-bit offset addressing mode, the effective address is obtained by adding the contents of the byte following the opcode to the contents of the index register. This mode of addressing is useful for selecting the kth element in an n element table. To use this mode, the table must begin in the lowest 256 memory locations and may extend through the first 511 memory locations (IFE is the last location which the instruction may access). Indexed 8-bit offset addressing can be used for ROM, RAM, or I/O. This is a 2-byte instruction with the offset contained in the byte following the opcode. The content of the index register (X) is not changed. The offset byte supplied in the instruction is an unsigned 8-bit integer.

7.1.7 Indexed, 16-bit Offset Addressing Mode

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the 8-bit index register and the two bytes following the opcode. The content of the index register is not changed. These instructions are three bytes, one for the opcode and two for a 16-bit offset.

7.1.8 Relative Addressing Mode

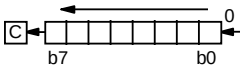
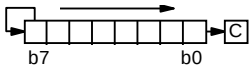
The relative addressing mode is used only for branch instructions. Branch instructions, other than the branching versions of bit-manipulation instructions, generate two machine-code bytes: one for the opcode and one for the relative offset. Because it is desirable to branch in either direction, the offset byte is a signed twos-complement offset with a range of –127 to +128 bytes (with respect to the address of the instruction immediately following the branch instruction). If the branch condition is true, the contents of the 8-bit signed byte following the opcode (offset) are added to the contents of the program counter to form the effective branch address; otherwise, control proceeds to the instruction immediately following the branch instruction.

7.2 Instruction Type

There are 65 instructions in CPU, and can be divided into 5 types.

1) Register/Memory Instructions
2) Read/Modify-Write Instructions
3) Branch Instructions
4) Control Instructions
5) bit manipulate Instructions

7.3 Instruction Set

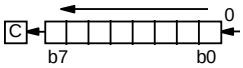
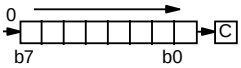
Instructions	Operating	Function	Status					Addressing Modes	Opcode	Opdata	#Cycle
			H	I	N	Z	C				
ADC #opr	Add with Carry	A← (A)+(M)+(C)	*	-	*	*	*	IMM	A9	ii	2
ADC opr								DIR	B9	dd	3
ADC opr								EXT	C9	hh	4
ADC opr,X								IX2	D9	ll	5
ADC opr,X								IX1	E9	ee	4
ADC ,X								IX	F9	ff	3
										ff	
ADD #opr	Add without Carry	A← (A)+(M)	*	-	*	*	*	IMM	AB	ii	2
ADD opr								DIR	BB	dd	3
ADD opr								EXT	CB	hh	4
ADD opr,X								IX2	DB	ll	5
ADD opr,X								IX1	EB	ee	4
ADD ,X								IX	FB	ff	3
										ff	
AND #opr	Logical AND	A← (A) ∧(M)	-	-	*	*	-	IMM	A4	ii	2
AND opr								DIR	B4	dd	3
AND opr								EXT	C4	hh	4
AND opr,X								IX2	D4	ll	5
AND opr,X								IX1	E4	ee	4
AND ,X								IX	F4	ff	3
										ff	
ASL opr	Arithmetic Shift Left (Same as LSL)		-	-	*	*	*	DIR	38	dd	5
ASLA								INH	48		3
ASLX								INH	58		3
ASL opr,X								IX1	68	ff	6
ASL ,X								IX	78		5
ASR opr	Arithmetic Shift Right		-	-	*	*	*	DIR	37	dd	5
ASRA								INH	47		3
ASRX								INH	57		3
ASR opr,X								IX1	67	ff	6
ASR ,X								IX	77		5
BCC rel	Branch if Carry Bit Clear	PC ←(PC)+2+rel ? C=0	-	-	-	-	-	REL	24	rr	3

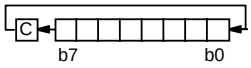
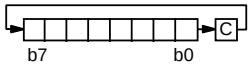
BCLR n opr	Clear Bit n	Mn←0	-	-	-	-	-	DIR(bo) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS rel	Branch if Carry Bit Set (Same as BLO)	PC ← (PC)+2+rel ? C=1 （the same as BLO）	-	-	-	-	-	REL	25	rr	3
BEQ rel	Branch if Equal	PC ← (PC)+2+rel ? Z=1	-	-	-	-	-	REL	27	rr	3
BHCC rel	Branch if Half Carry Bit Clear	PC← (PC)+2+rel ? H=0	-	-	-	-	-	REL	28	rr	3
BHCS rel	Branch if Half Carry Bit Set	PC← (PC)+2+rel ? H=1	-	-	-	-	-	REL	29	rr	3
BHI rel	Branch if Higher	PC← (PC)+2+rel ? (C∨Z)=0	-	-	-	-	-	REL	22	rr	3
BHS rel	Branch if Higher or Same	PC ← (PC)+2+rel ? C=0	-	-	-	-	-	REL	24	rr	3
BIT #opr BIT opr BIT opr BIT opr,X BIT opr,X BIT ,X	Bit Test Accumulator with Memory Byte	(A)∧(M)	-	-	*	*	-	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff	2 3 4 5 4 3
BLO rel	Branch if Lower (Same as BCS)	PC ← (PC)+2+rel ? C=1	-	-	-	-	-	REL	25	rr	3
BLS rel	Branch if Lower or Same	PC← (PC)+2+rel ? (C∨Z)=1	-	-	-	-	-	REL	23	rr	3
BMC rel	Branch if Interrupt Mask Clear	PC← (PC)+2+rel ? I=0	-	-	-	-	-	REL	2C	rr	3
BMI rel	Branch if Minus	PC← (PC)+2+rel ? N=1	-	-	-	-	-	REL	2B	rr	3

BMS rel	Branch if Interrupt Mask Set	PC← (PC)+2+rel ? I=1	-	-	-	-	-	REL	2D	rr	3
BNE rel	Branch if Not Equal	PC← (PC)+2+rel ? Z=0	-	-	-	-	-	REL	26	rr	3
BPL rel	Branch if Plus	PC← (PC)+2+rel ? N=0	-	-	-	-	-	REL	2A	rr	3
BRA rel	Branch Always	PC← (PC)+2+rel	-	-	-	-	-	REL	20	rr	3
BRCLR n opr rel	Branch if Bit n Clear	PC← (PC)+2+rel ? Mn=0	-	-	-	-	*	DIR(bo) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN rel	Branch Never	PC← (PC)+2	-	-	-	-	-	REL	21	rr	3
BRSET n opr rel	Branch if Bit n Set	PC← (PC)+2+rel ? Mn=1	-	-	-	-	*	DIR(bo) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5

										dd rr dd rr	
BSET n opr	Set Bit n	Mn←1	-	-	-	-	-	DIR(bo) DIR(b1) DIR(b2) DIR(b3) DIR(b4) DIR(b5) DIR(b6) DIR(b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR rel	Branch to Subroutine	PC←(PC)+2 push(PCL);SP←(SP)-1 push(PCH);SP←(SP)-1 PC← (PC)+rel	-	-	-	-	-	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	-	-	-	-	0	INH	98		2
CLI	Clear Interrupt Mask	I ←0	-	0	-	-	-	INH	9A		2
CLR opr CLRA CLRX CLR opr,X CLR ,X	Clear Byte	M←$00 A ←$00 X ←$00 M←$00 M←$00	-	-	0	1	-	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #opr CMP opr CMP opr CMP opr,X CMP opr,X CMP ,X	Compare Accumulator with Memory Byte	(A) -(M)	-	-	*	*	*	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff	2 3 4 5 4 3
COM opr COMA COMX COM opr,X COM ,X	Complement Byte (One's Complement)	M←$FF-(M) A ←$FF-(A) X ←$FF-(X) M←$FF-(M) M←$FF-(M)	-	-	*	*	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5

CPX #opr CPX opr CPX opr CPX opr,X CPX opr,X CPX ,X	Compare Index Register with Memory Byte	(X) -(M)	-	-	*	*	*	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff	2 3 4 5 4 3
DEC opr DECA DECX DEC opr,X DEC ,X	Decrement Byte	M←(M)-1 A ←(A)-1 X ←(X)-1 M←(M)-1 M←(M)-1	-	-	*	*	-	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #opr EOR opr EOR opr EOR opr,X EOR opr,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ←(A) ⊕ (M)	-	-	*	*	-	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff	2 3 4 5 4 3
INC opr INCA INCX INC opr,X INC ,X	Increment Byte	M←(M)+1 A ←(A)+1 X ←(X)+1 M←(M)+1 M←(M)+1	-	-	*	*	-	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP opr JMP opr JMP opr,X JMP opr,X JMP ,X	Unconditional Jump	PC ←Jump Address	-	-	-	-	-	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR opr JSR opr JSR opr,X JSR opr,X JSR ,X	Jump to Subroutine	PC←(PC)+n(n=1,2,or 3) push (PCL);SP←(SP)-1 push(PCH);SP←(SP)-1 PC← Effective	-	-	-	-	-	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 7 6 5

		Address									
LDA #opr LDA opr LDA opr LDA opr,X LDA opr,X LDA ,X	Load Accumulator with Memory Byte	A ←(M)	-	-	*	*	-	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #opr LDX opr LDX opr LDX opr,X LDX opr,X LDX ,X	Load Index Register with Memory Byte	X ←(M)	-	-	*	*	-	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL opr LSLA LSLX LSL opr,X LSL ,X	Logical Shift Left (Same as ASL)		-	-	*	*	*	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
LSR opr LSRA LSRX LSR opr,X LSR ,X	Logical Shift Right		-	-	0	*	*	DIR INH INH IX1 IX	34 44 54 64 74	dd ff	5 3 3 6 5
MUL	Unsigned Multiply	X:A ← (X)X(A)	0	-	-	-	0	INH	42		1 1
NEG opr NEGA NEGX NEG opr,X NEG ,X	Negate Byte (Two's Complement)	M←-(M) A ←-(A) X ←-(X) M←-(M) M←-(M)	-	-	*	*	*	DIR INH INH IX1 IX	30 40 50 60 70	dd ff	5 3 3 6 5
NOP	No Operation		-	-	-	-	-	INH	9D		2
ORA #opr ORA opr ORA opr ORA opr,X	Logical OR Accumulator with Memory	A ←(A) ∨(M)	-	-	*	*	-	IMM DIR EXT IX2	AA BA CA DA	ii dd hh ll	2 3 4 5

ORA opr,X ORA ,X								IX1 IX	EA FA	ee ff ff	4 3
ROL opr ROLA ROLX ROL opr,X ROL ,X	Rotate Byte Left through Carry Bit		-	-	*	*	*	DIR INH INH IX1 IX	39 49 59 69 79	dd ff 	5 3 3 6 5
ROR opr RORA RORX ROR opr,X ROR ,X	Rotate Byte Right through Carry Bit		-	-	*	*	*	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP← $00FF	-	-	-	-	-	INH	9C		2
RTI	Return from Interrupt	SP←(SP)+1; Pull(CCR) SP← (SP)+1; Pull(A) SP← (SP)+1; Pull(X) SP←(SP)+1; Pull(PCH) SP←(SP)+1; Pull(PCL)	*	*	*	*	*	INH	80		9
RTS	Return from Subroutine	SP←(SP)+1; Pull(PCH) SP←(SP)+1; Pull(PCL)	-	-	-	-	-	INH	81		6
SBC #opr SBC opr SBC opr SBC opr,X SBC opr,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A)-(M)-(C)	-	-	*	*	*	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	-	-	-	-	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	-	1	-	-	-	INH	9B		2
STA opr STA opr	Store Accumulator in							DIR EXT	B7 C7	dd hh	4 5

STA opr,X STA opr,X STA ,X	Memory	M ← (A)	-	-	*	*	-	IX2 IX1 IX	D7 E7 F7	ll ee ff ff	6 5 4
STOP	Stop Oscillator and Enable IRQ Pin		-	0	-	-	-	INH	8E		2
STX opr STX opr STX opr,X STX opr,X STX ,X	Store Index Register In Memory	M ← (X)	-	-	*	*	-	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4
SUC #opr SUB opr SUB opr SUB opr,X SUB opr,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A)- (M)	-	-	*	*	*	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC←(PC)+1;Push(PCL) SP←(SP)-1;Push(PCH) SP← (SP)-1; Push(X) SP←(SP)-1; ush(CCR) SP← (SP)-1;I ←1 PCH←Interrupt Vector High Byte PCL ←Interrupt Vector Low Byte	-	1	-	-	-	INH	83		1 0
TAX	Transfer Accumulator to Index Register	X ←(A)	-	-	-	-	-	INH	97		2
TST opr TSTA TSTX TST opr,X	Test Memory Byte for Negative or Zero	(M)-$00	-	-	*	*	-	DIR INH INH IX1	3D 4D 5D 6D	dd ff	4 3 3 5

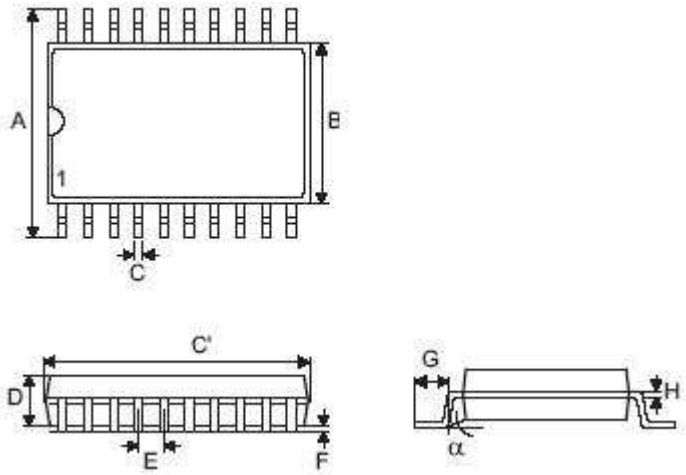
TST ,X								IX	7D		4
TXA	Transfer Index Register to Accumulator	A ← (X)	-	-	-	-	-	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		-	0	-	-	-	INH	8F		2

A Accumulator
C Carry/borrow flag
CCR Condition code register
dd Direct address of operand
dd rr Direct address of operand and relative offset of branch instruction
DIR Direct addressing mode
ee ff High and low bytes of offset in indexed, 16-bit offset addressing
EXT Extended addressing mode
ff Offset byte in indexed, 8-bit offset addressing
H Half-carry flag
hh ll High and low bytes of operand address in extended addressing
I Interrupt mask
ii Immediate operand byte
IMM Immediate addressing mode
INH Inherent addressing mode
IX Indexed, no offset addressing mode
IX1 Indexed, 8-bit offset addressing mode
IX2 Indexed, 16-bit offset addressing mode
M Memory location
N Negative flag
n Any bit — Not affected

opr Operand (one or two bytes)
PC Program counter
PCH Program counter high byte
PCL Program counter low byte
REL Relative addressing mode
rel Relative program counter offset byte
rr Relative program counter offset byte
SP Stack pointer
X Index register
Z Zero flag
Immediate value
∧ Logical AND
∨ Logical OR
⊕ Logical EXCLUSIVE OR
( ) Contents of
–( ) Negation (twos complement)
← Loaded with
? If
: Concatenated with
↔ Set or cleared
— Not affected

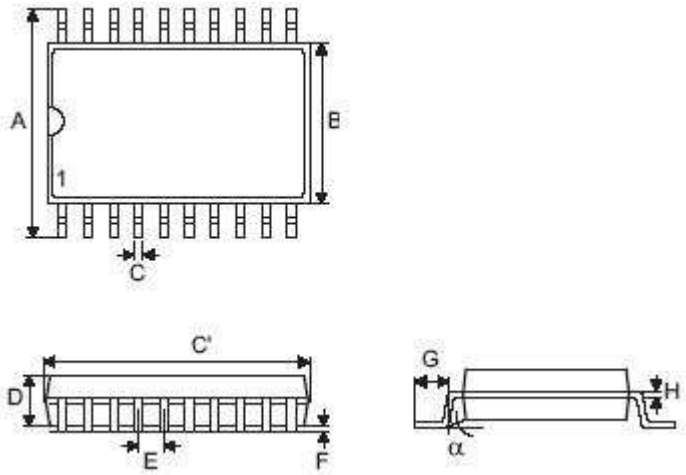
8. Package

SOP20（300mil）



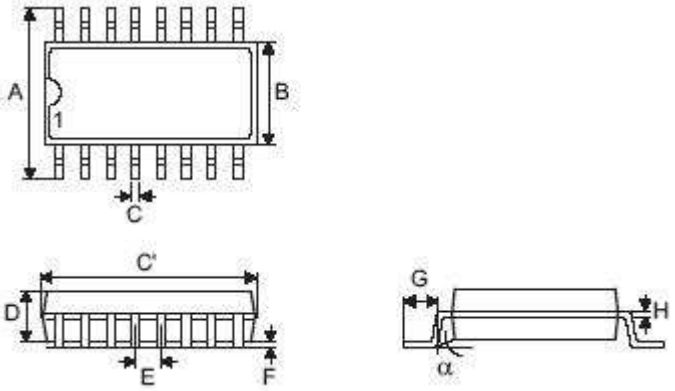
Symbol	Dimensions in mil			Dimensions in milimeter		
	Max.	Nom.	Min.	Max.	Nom.	Min.
A	394	-	420	10.01	-	10.67
B	290	-	300	7.37	-	7.62
C	14	-	20	0.36	-	0.51
C’	495	-	512	12.57	-	13.00
D	92	-	104	2.34	-	2.64
E	-	50	-	-	1.27	-
F	4	-	-	0.10	-	-
G	32	-	38	0.81	-	0.97
H	4	-	12	0.10	-	0.30
α	0°	-	8°	0°	-	8°

SSOP20（200mil）



Symbol	Dimensions in mil			Dimensions in milimeter		
	Max.	Nom.	Min.	Max.	Nom.	Min.
A	299	307	315	7.60	7.80	8.00
B	201	209	217	5.10	5.30	5.50
C	11	-	15	0.29	-	0.37
C’	276	283	291	7.00	7.20	7.40
D	51	59	67	1.30	1.50	1.70
E	-	25.6	-	-	0.65	-
F	2	6	10	0.05	0.15	0.25
G	30	35	40	0.75	0.90	1.05
H	6	-	8	0.15	-	0.20
α	0°	-	8°	0°	-	8°

SOP16（150mil）



Symbol	Dimensions in mil			Dimensions in milimeter		
	Max.	Nom.	Min.	Max.	Nom.	Min.
A	238	-	244	6.05	-	6.20
B	150	-	157	3.80	-	4.00
C	14	-	19	0.36	-	0.48
C’	386	-	398	9.80	-	10.10
D	53	-	62	1.35	-	1.57
E	-	50	-	-	1.27	-
F	4	-	-	0.10	-	-
G	22	-	32	0.56	-	0.82
H	4	-	12	0.10	-	0.30
α	0°	-	8°	0°	-	8°