

专业PCB打样工厂,24小时加急出货 HT48CA0 8-bit Microcontroller

Features

- Operating voltage: 2.2V~3.6V
- Ten bidirectional I/O lines
- Six schmitt trigger input lines
- One carrier output (1/2 or 1/3 duty)
- On-chip crystal and RC oscillator
- Watchdog timer
- 1K×14 program ROM
- 32×8 data RAM
- Low voltage reset function

General Description

The HT48CA0 is an 8-bit high performance RISC-like microcontroller specifically designed for multiple I/O product applications. The device is particularly suitable for use in products Halt function and wake-up feature reduce power consumption

- 62 powerful instructions
- Up to 1µs instruction cycle with 4MHz system clock
- All instructions in 1 or 2 machine cycles
- 14-bit table read instructions
- **One-level subroutine nesting**
- Bit manipulation instructions

such as remote controllers, fan/light controllers, washing machine controllers, scales, toys and various subsystem controllers. A halt feature is included to reduce power consumption.

Pin Assignment







Block Diagram





Pad Description

Pad No.	Pad Name	I/O	Mask Option	Function
1, 22	PB0, PB1	I/O	Wake-up or None	2-bit bidirectional input/output lines with pull-high resistors. Each bit can be determined as NMOS output or schmitt trigger input by software instructions. Each bit can also be configured as wake-up input by mask option.
2	PC0/REM	0	Level or Carrier	Level or carrier output pin PC0 can be set as CMOS output pin or carrier output pin by mask option.
3	VDD	_		Positive power supply
6	VSS	_	_	Negative power supply, GND
7	RES	Ι	_	Schmitt trigger reset input. Active low.
13~8	PB2~PB7	Ι	Wake-up or None	6-bit schmitt trigger input lines with pull-high resistors. Each bit can be configured as a wake-up input by mask option.
21~14	PA0~PA7	I/O	_	Bidirectional 8-bit input/output port with pull-high resistors. Each bit can be determined as NMOS output or schmitt trigger input by software instructions.
	OSC1 OSC2	I O	Crystal or RC	OSC1, OSC2 are connected to an RC network or a crystal (determined by mask option) for the internal system clock. In the case of RC operation, OSC2 is the output terminal for 1/4 system clock (NMOS open drain output).



Pad Assignment



* The IC substrate should be connected to VSS in the PCB layout artwork.

* The TMR pad must be bonded to VDD or VSS if the TMR pad is not used.

Absolute Maximum Ratings*

Supply Voltage0.3V t	to 4V	Storage Temperature50°C to 1	25°C
Input Voltage $V_{SS}0.3V$ to $V_{DD}\mbox{+}$	0.3V	Operating Temperature –25°C to	70°C

*Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.



Ta=25°C

D.C. Characteristics

Chh	D		st Conditions	N#*	T		I Init	
Symbol	Parameter	V _{DD} Conditions		Min.	тур.	Max.	Unit	
V _{DD}	Operating Voltage	_	—	2.2	_	3.6	V	
I _{DD}	Operating Current	3V	No load, f _{SYS} =4MHz	_	0.7	1.5	mA	
I _{STB} Standby Current		3V	No load, system HALT	_	_	1	μΑ	
V _{IL1}	Input Low Voltage for I/O Ports	3V	_	0	_	1.05	V	
V _{IH1}	Input High Voltage for I/O Ports	3V	—	1.95	_	3	V	
V _{IL2}	Input Low Voltage (RES)	3V	_	_	1.5		V	
V _{IH2}	Input High Voltage (RES)	3V	_	_	2.4		V	
Iol	I/O Ports Sink Current	3V	Vol=0.3V	1.5	2.5		mA	
I _{OH}	I/O Ports Source Current	3V	V _{OH} =2.7V	-1	-1.5		mA	
R _{PH1}	Pull-high Resistance of PA Port, PB0~PB1 and RES	3V	_	_	60	_	kΩ	
R _{PH2}	Pull-high Resistance of PB2~PB7	3V	-	_	60	_	kΩ	
VLVR	Low Voltage Reset	3V	_	1.8	2.0	2.2	V	

A.C. Characteristics

Ta=25°C

Symbol	Parametar	Т	est Conditions	Min	Тур.	Mov	Unit
Symbol	r al ameter	V _{DD}	Conditions	IVIIII.		IVIAA.	
f _{SYS}	System Clock	3V	—	400		4000	kHz
t _{RES}	External Reset Low Pulse Width	_	—	1		_	μs
t _{SST}	System Start-up timer Period	_	Power-up or wake-up from halt		1024		t _{SYS}

Note: t_{SYS}=1/f_{SYS}



Application Circuit



Notes: It is recommended that a $0.1\mu F$ decoupling capacitor is placed between VSS and VDD. If the crystal has a value above 1MHz the capacitors are not required.





System Architecture

Execution flow

The HT48CA0 system clock can be derived from a crystal/ceramic resonator oscillator. It is internally divided into four non-overlapping clocks. One instruction cycle consists of four system clock cycles.

Instruction fetching and execution are pipelined in such a way that a fetch takes one instruction cycle while decoding and execution takes the next instruction cycle. However, the pipelining scheme causes each instruction to effectively execute within one cycle. If an instruction changes the program counter, two cycles are required to complete the instruction.

Program counter - PC

The 10-bit program counter (PC) controls the sequence in which the instructions stored in program ROM are executed and its contents specify a maximum of 1024 addresses.

After accessing a program memory word to fetch an instruction code, the contents of the program counter are incremented by one. The program counter then points to the memory word containing the next instruction code.

When executing a jump instruction, conditional skip execution, loading PCL register, subroutine call, initial reset or return from subroutine, the PC manipulates the program transfer by loading the address corresponding to each instruction. The conditional skip is activated by instruction. Once the condition is met, the next instruction, fetched during the current instruction execution, is discarded and a dummy cycle replaces it to get the proper instruction. Otherwise proceed with the next instruction.

The lower byte of the program counter (PCL) is a readable and writeable register (06H). Moving data into the PCL performs a short jump. The destination will be within 256 locations.

When a control transfer takes place, an additional dummy cycle is required.

Program memory - ROM

The program memory is used to store the program instructions which are to be executed. It also contains data and table and is organized into 1024×14 bits, addressed by the program counter and table pointer.

Certain locations in the program memory are reserved for special usage:

• Location 000H

This area is reserved for the initialization program. After chip reset, the program always begins execution at location 000H.

• Table location

Any location in the ROM space can be used as look-up tables. The instructions TABRDC [m] (the current page, 1 page=256 words) and TABRDL [m] (the last page) transfer the contents of the lower-order byte to the specified

System Clock	T1 T2 T3 T4		
Instruction Cycle		/	<u> </u>
PC	<pre>PC</pre>	PC+1	PC+2
	Fetch INST (PC)		
	Execute INST (PC-1)	Fetch INST (PC+1)	
		Execute INST (PC)	Fetch INST (PC+2)
			Execute INST (PC+1)

Execution flow



data memory, and the higher-order byte to TBLH (08H). Only the destination of the lower-order byte in the table is well-defined, the other bits of the table word are transferred to the lower portion of TBLH, the remaining 2 bits are read as "0". The Table Higher-order byte register (TBLH) is read only. The table pointer (TBLP) is a read/write register (07H), where P indicates the table location. Before accessing the table, the location must be placed in TBLP. The TBLH is read only and cannot be restored. All table related instructions need 2 cycles to complete the operation. These areas may function as normal program memory depending upon the requirements.

Stack register – STACK

This is a special part of the memory used to save the contents of the program counter (PC) only. The stack is organized into one level and is neither part of the data nor part of the program space, and is neither readable nor writeable. The activated level is indexed by the stack pointer (SP) and is neither readable nor write able. At a subroutine call the contents of the program counter are pushed onto the stack. At the end of a subroutine signaled by a return instruction (RET), the program counter is restored to its previous value from the stack. After a chip reset, the SP will point to the top of the stack.





Look-up table (256 words)

-14 bits-

If the stack is full and a "CALL" is subsequently executed, stack overflow occurs and the first entry will be lost (only the most recent return address is stored).

Data memory – RAM

000H

n00H

nFFH

3FFF

The data memory is designed with 42×8 bits. The data memory is divided into two functional groups: special function registers and general purpose data memory (32×8). Most of them are read/write, but some are read only.

Mode		Program Counter								
		*8	*7	*6	*5	*4	*3	*2	*1	*0
Initial reset		0	0	0	0	0	0	0	0	0
Skip	PC+2									
Loading PCL	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, call branch	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return from subroutine		S8	S7	S6	S5	S4	S3	S2	S1	S0

Program counter

Notes: *9~*0: Program counter bits #9~#0: Instruction code bits S9~S0: Stack register bits @7~@0: PCL bits



The special function registers include the indirect addressing register (00H), the memory pointer register (MP;01H), the accumulator (ACC;05H) the program counter lower-order byte register (PCL;06H), the table pointer (TBLP;07H), the table higher-order byte register (TBLH;08H), the status register (STATUS;0AH) and the I/O registers (PA;12H, PB;14H, PC;16H). The remaining space before the 20H is reserved for future expanded usage and reading these locations will return the result 00H. The general purpose data memory, addressed from 20H to 3FH, is used for data and control information under instruction command.

All data memory areas can handle arithmetic, logic, increment, decrement and rotate operations directly. Except for some dedicated bits, each bit in the data memory can be set and reset by the SET [m].i and CLR [m].i instructions, respectively. They are also indirectly accessible through memory pointer register (MP;01H).

Indirect addressing register

Location 00H is an indirect addressing register that is not physically implemented. Any read/write operation of [00H] accesses data memory pointed to by MP (01H). Reading location 00H itself indirectly will return the result 00H. Writing indirectly results in no operation.

The memory pointer register MP (01H) is a 6-bit register. The bit 7~6 of MP is undefined and reading will return the result "1". Any writing operation to MP will only transfer the lower 6-bit data to MP.



RAM mapping

Instruction(s)	Table Location									
Instruction(s)	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
TABRDC [m]	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	1	1	@7	@6	@5	@4	@3	@2	@1	@0

Table location

Notes: *9~*0: Table location bits @7~@0: Table pointer bits P9~P8: Current program counter bits

Accumulator

The accumulator closely relates to ALU operations. It is also mapped to location 05H of the data memory and is capable of carrying out immediate data operations. Data movement between two data memory locations has to pass through the accumulator.

Arithmetic and logic unit – ALU

This circuit performs 8-bit arithmetic and logic operation. The ALU provides the following functions.

- Arithmetic operations (ADD, ADC, SUB, SBC, DAA)
- Logic operations (AND, OR, XOR, CPL)
- Rotation (RL, RR, RLC, RRC)
- Increment and Decrement (INC, DEC)
- Branch decision (SZ, SNZ, SIZ, SDZ)

The ALU not only saves the results of a data operation but also changes the contents of the status register.

Status register – STATUS

This 8-bit status register (0AH) contains the zero flag (Z), carry flag (C), auxiliary carry flag (AC),

overflow flag (OV), power down flag (PD) and watchdog time-out flag (TO). It also records the status information and controls the operation sequence.

With the exception of the TO and PD flags, bits in the status register can be altered by instructions like most other register. Any data written into the status register will not change the TO or PD flags. In addition it should be noted that operations related to the status register may give different results from those intended. The TO and PD flags can only be changed by the Watchdog Timer overflow, chip power-up, clearing the Watchdog Timer and executing the HALT instruction.

The Z, OV, AC and C flags generally reflect the status of the latest operations.

In addition, on executing the subroutine call, the status register will not be automatically pushed onto the stack. If the contents of the status are important and if the subroutine can corrupt the status register, precautions must be taken to save it properly.

Labole	Bite	Function
Labels	Dits	Function
С	0	C is set if the operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation; otherwise C is cleared. C is also affected by a rotate through carry instruction.
AC 1		AC is set if the operation results in a carry out of the low nibbles in addition or no borrow from the high nibble into the low nibble in subtraction; otherwise AC is cleared.
Z	2	\boldsymbol{Z} is set if the result of an arithmetic or logic operation is zero; otherwise \boldsymbol{Z} is cleared.
ov	3	OV is set if the operation results in a carry into the highest-order bit but not a carry out of the highest-order bit, or vice versa; otherwise OV is cleared.
PD	4	PD is cleared when either a system power-up or executing the CLR WDT instruction. PD is set by executing the HALT instruction.
то	5	TO is cleared by a system power-up or executing the CLR WDT or HALT instruction. TO is set by a WDT time-out.
	6	Undefined, read as "0"
_	7	Undefined, read as "0"

Status register



Oscillator configuration

There are two oscillator circuits in the HT48CA0.



System oscillator

Both are designed for system clocks; the RC oscillator and the Crystal oscillator, which are determined by mask options. No matter what oscillator type is selected, the signal provides the system clock. The HALT mode stops the system oscillator and ignores the external signal to conserve power.

If an RC oscillator is used, an external resistor between OSC1 and VSS in needed and the resistance must range from $51k\Omega$ to $1M\Omega$. The system clock, divided by 4, is available on OSC2, which can be used to synchronize external logic. The RC oscillator provides the most cost effective solution. However, the frequency of the oscillation may vary with VDD, temperature and the chip itself due to process variations. It is, therefore, not suitable for timing sensitive operations where accurate oscillator frequency is desired.

If the Crystal oscillator is used, a crystal across OSC1 and OSC2 is needed to provide the feedback and phase shift for the oscillator. No other external components are needed. Instead of a crystal, the resonator can also be connected between OSC1 and OSC2 to get a frequency reference, but two external capacitors in OSC1 and OSC2 are required.

Watchdog timer - WDT

The clock source of the WDT is implemented by instruction clock (system clock divided by 4). The clock source is processed by a frequency divider and a prescaller to yield various time out periods.

WDT time out period =
$$\frac{\text{Clock Source}}{2^n}$$

Where n= 8~11 selected by mask option.

This timer is designed to prevent a software malfunction or sequence jumping to an unknown location with unpredictable results. The Watchdog Timer can be disabled by mask option. If the Watchdog Timer is disabled, all the executions related to the WDT result in no op-





eration and the WDT will lose its protection purpose. In this situation the logic can only be restarted by an external logic.

A WDT overflow under normal operation will initialize "chip reset" and set the status bit "TO". To clear the contents of the WDT prescaler, three methods are adopted; external reset (a low level to RES), software instructions, or a HALT instruction. There are two types of software instructions. One type is the single instruction "CLR WDT", the other type comprises two instructions, "CLR WDT1" and "CLR WDT2". Of these two types of instructions, only one can be active depending on the mask option - "CLR WDT times selection option". If the "CLR WDT" is selected (i.e.. CLRWDT times equal one), any execution of the CLR WDT instruction will clear the WDT. In case "CLR WDT1" and "CLR WDT2" are chosen (i.e.. CLRWDT times equal two), these two instructions must be executed to clear the WDT; otherwise, the WDT may reset the chip due to a time-out.

Power down operation – HALT

The HALT mode is initialized by the HALT instruction and results in the following...

- The system oscillator turns off and the WDT stops.
- The contents of the on-chip RAM and registers remain unchanged.
- WDT prescaler are cleared.
- All I/O ports maintain their original status.
- The PD flag is set and the TO flag is cleared.

The system can quit the HALT mode by means of an external reset or an external falling edge signal on port B. An external reset causes a device initialization. Examining the TO and PD flags, the reason for chip reset can be determined. The PD flag is cleared when the system powers up or execute the CLR WDT instruction and is set when the HALT instruction is executed. The TO flag is set if the WDT time-out occurs, and causes a wake-up that only resets the PC (Program Counter) and SP, the others keep their original status.

The port B wake-up can be considered as a continuation of normal execution. Each bit in port B can be independently selected to wake up the device by the mask option. Awakening from an I/O port stimulus, the program will resume execution of the next instruction.

Once a wake-up event(s) occurs, it takes $1024 t_{SYS}$ (system clock period) to resume normal operation. In other words, a dummy cycle period will be inserted after the wake-up.

To minimize power consumption, all I/O pins should be carefully managed before entering the HALT status.

Reset

There are three ways in which a reset can occur:

- RES reset during normal operation
- RES reset during HALT
- WDT time-out reset during normal operation

Some registers remain unchanged during reset conditions. Most registers are reset to the "initial condition" when the reset conditions are met. By examining the PD and TO flags, the program can distinguish between different "chip resets".

то	PD	RESET Conditions
0	0	RES reset during power-up
u	u	RES reset during normal operation
0	1	RES wake-up HALT
1	u	WDT time-out during normal operation

Note: "u" means "unchanged".







Reset configuration

To guarantee that the system oscillator has started and stabilized, the SST (System Startup Timer) provides an extra-delay of 1024 system clock pulses when the system powers up or when the system awakes from a HALT state.

HT48CA0

When a system power up occurs, an SST delay is added during the reset period. But when the reset comes from the $\overline{\text{RES}}$ pin, the SST delay is disabled. Any wake-up from HALT will enable the SST delay.



Reset timing chart

The functional unit chip reset status is shown below.

PC	000H		
WDT Prescaler	Clear		
Input/output Ports	Input mode		
SP	Points to the top of the stack		
Carrier Output	Low level		

Γ	The chip reset status of	f the registers is summarized i	in the following table:

Register	Reset (power on)	WDT time-out (normal operation)	RES reset (normal operation)	RES reset (HALT)
PC (Program Counter)	000H	000H	000H	000H
MP	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xx xxxx	uu uuuu	uu uuuu	uu uuuu
STATUS	00 xxxx	1u uuuu	uu uuuu	01 uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111
PB	1111 1111	1111 1111	1111 1111	1111 1111
PC	1	1	1	1

Notes: "u" means "unchanged"

"x" means "unknown"



Low voltage reset - LVR

The HT48CA0 provides low voltage reset circuit in order to monitor the supply voltage of the device. If the supply voltage of the device is within the range 0.9V~2.2V, such as changing a battery, the LVR will automatically reset the device internally.

The LVR includes the following specifications:

- The low voltage (0.9V~2.2V) has to remain in their original state to exceed 1 ms. If the low voltage state does not exceed 1 ms, the LVR will ignore it and do not perform a reset function.
- The LVR uses the "OR" function with the external $\overline{\text{RES}}$ signal to perform chip reset.
- During HALT mode, if the LVR occurs, the device will wake-up and the PD flag will be set as "1", the same as the external RES.

Because the operating voltage (V_{DD}) is 2.2V~3.6V and the LVR operating voltage (V_{LVR}) is 0.9V~2.2V, therefore one margin voltage about 0.1V is needed for proper chip operation. The relationship between V_{DD} and V_{LVR} is shown below.



Note: V_{OPR} is the voltage range for proper chip operation at 4MHz system clock.

Carrier

The HT48CA0 provides a carrier output which shares the pin with PC0. It can be selected to be a carrier output (REM) or level output pin (PC0) by mask option. If the carrier output option is selected, setting PC0="0" to enable carrier output and setting PC0="1" to disable it at low level output.



- *1: To make sure that the system oscillator has stabilized, the SST provides an extra delay of 1024 system clock pulses before entering the normal operation.
- *2: Since the low voltage has to maintain in its original state and exceed 1 ms, therefore 1 ms delay is needed to enter the reset mode.



The clock source of the carrier is implemented by instruction clock (system clock divided by 4) and processed by a frequency divider to yield various carry frequency.

Carry Frequency= $\frac{Clock \ Source}{m \times 2^n}$

where m=2 or 3 and n=0~3, both are selected by mask option. If m=2, the duty cycle of the carrier output is 1/2 duty. If m=3, the duty cycle of the carrier output can be 1/2 duty or 1/3 duty also determined by mask option (with the exception of n=0).

Detailed selection of the carrier duty is shown below:

m ×2 ⁿ	Duty Cycle
2, 4, 8, 16	1/2
3	1/3
6, 12, 24	1/2 or 1/3

Input/output ports

There are an 8-bit bidirectional input/output port, a 6-bit input with 2-bit I/O port and one-bit

output port in the HT48CA0, labeled PA, PB and PC which are mapped to [12H], [14H], [16H] of the RAM, respectively. Each bit of PA can be selected as NMOS output or schmitt trigger with pull-high resistor by software instruction. PB0~PB1 have the same structure with PA, while PB2~PB7 can only be used for input operation (schmitt trigger with pull-high resistors). PC is only one-bit output port shares the pin with carrier output. If the level option is selected, the PC is CMOS output.

Both PA and PB for the input operation, these ports are non-latched, that is, the inputs should be ready at the T2 rising edge of the instruction "MOV A, [m]" (m=12H or 14H). For PA, PB0~PB1 and PC output operation, all data are latched and remain unchanged until the output latch is rewritten.

When the PA and PB0~PB1 is used for input operation, it should be noted that before reading data from pads, a "1" should be written to the related bits to disable the NMOS device. That is, the instruction "SET [m].i" (i=0~7 for PA, i=0~1 for PB) is executed first to disable related NMOS device, and then "MOV A, [m]" to get stable data.



PB input lines



After chip reset, PA and PB remain at a high level input line while PC remain at high level output, if the level option is selected.

Each bit of PA, PB0~PB1 and PC output latches can be set or cleared by the "SET [m].i" and "CLR [m].i" (m=12H, 14H or 16H) instructions respectively.

Some instructions first input data and then follow the output operations. For example, "SET [m].i", "CLR [m]", "CPL [m]", "CPLA [m]" read the entire port states into the CPU, execute the defined operations (bit-operation), and then write the results back to the latches or to the accumulator.

Each line of PB has a wake-up capability to the device by mask option. The highest seven bits of PC are not physically implemented, on reading them a "0" is returned and writing results in a no-operation.



PA, PB Input/output lines



Mask option

The following table shows eight kinds of mask option in the HT48CA0. All the mask options must be defined to ensure proper system functioning.

No.	Mask Option
1	WDT time-out period selection Time-out period= $\frac{\text{Clock Source}}{2^n}$ where n=8~11.
2	WDT enable/disable selection. This option is to decide whether the WDT timer is enabled or disabled.
3	CLRWDT times selection. This option defines how to clear the WDT by instruction. "One time" means that the CLR WDT instruction can clear the WDT. "Two times" means only if both of the CLR WDT1 and CLR WDT2 instructions have been executed, the WDT can be cleared.
4	Wake-up selection. This option defines the wake-up activity function. External input pins (PB only) all have the capability to wake-up the chip from a HALT.
5	Carrier/level output selection. This option defines the activity of PC0 to be carrier output or level output.
6	Carry frequency selection. Carry frequency= $\frac{\text{Clock Source}}{(2 \text{ or } 3) \times 2^n}$ where n=0~3.
7	Carrier duty selection. There are two types of selection: 1/2 duty or 1/3 duty. If carrier frequency= Clock Source / (2, 4, 8 or 16), the duty cycle will be 1/2 duty. If carrier frequency= Clock Source / 3, the duty cycle will be 1/3 duty. If carrier frequency= Clock Source / (6, 12 or 24), the duty cycle can be 1/2 duty or 1/3 duty.
8	OSC type selection. This option is to decide if an RC or Crystal oscillator is chosen as system clock. If the Crystal oscillator is selected, the XST (Crystal Start-up Timer) default is activated, otherwise the XST is disabled.



Instruction Set Summary

Mnemonic	Description	Instruction Cycle	Flag Affected
Arithmetic			
ADD A,[m] ADDM A,[m] ADD A,x ADC A,[m] ADCM A,[m] SUB A,x SUB A,[m] SUBM A,[m] SBC A,[m] SBCM A,[m]	Add data memory to ACC Add ACC to data memory Add immediate data to ACC Add data memory to ACC with carry Add ACC to register with carry Subtract immediate data from ACC Subtract data memory from ACC Subtract data memory from ACC with result in data memory Subtract data memory from ACC with carry Subtract data memory from ACC with carry with result in data memory Decimal adjust ACC for addition with result in	$1 \\ 1^{(1)} \\ 1 \\ 1^{(1)} \\ 1 \\ 1^{(1)} \\ 1^{(1)} \\ 1^{(1)} \\ 1^{(1)} \\ 1^{(1)}$	Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV Z,C,AC,OV
Logic Operation	data memory		
AND A,[m] OR A,[m] XOR A,[m] ANDM A,[m] ORM A,[m] XORM A,[m] AND A,x OR A,x XOR A,x CPL [m] CPLA [m] Increment &	AND data memory to ACC OR data memory to ACC Exclusive-OR data memory to ACC AND ACC to data memory OR ACC to data memory Exclusive-OR ACC to data memory AND immediate data to ACC OR immediate data to ACC Exclusive-OR immediate data to ACC Complement data memory Complement data memory with result in ACC	$1 \\ 1 \\ 1^{(1)} \\ 1^{(1)} \\ 1^{(1)} \\ 1 \\ 1 \\ 1^{(1)} \\ 1$	Z Z Z Z Z Z Z Z Z Z Z
Decrement INCA [m] INC [m] DECA [m] DEC [m]	Increment data memory with result in ACC Increment data memory Decrement data memory with result in ACC Decrement data memory	$1 \\ 1^{(1)} \\ 1 \\ 1^{(1)}$	Z Z Z Z



Mnemonic	Description	Instruction Cycle	Flag Affected
Rotate			
RRA [m]	Rotate data memory right with result in ACC	1	None
RR [m]	Rotate data memory right	1 ⁽¹⁾	None
RRCA [m]	Rotate data memory right through carry with result in ACC	1	С
RRC [m]	Rotate data memory right through carry	1 ⁽¹⁾	С
RLA [m]	Rotate data memory left with result in ACC	1	None
RL [m]	Rotate data memory left	1 ⁽¹⁾	None
RLCA [m]	Rotate data memory left through carry with result in ACC	1	C
RLC [m]	Rotate data memory left through carry	1 ⁽¹⁾	С
Data Move			
MOV A,[m]	Move data memory to ACC	1	None
MOV [m],A	Move ACC to data memory	1 ⁽¹⁾	None
MOV A,x	Move immediate data to ACC	1	None
Bit Operation			
CLR [m].i	Clear bit of data memory	1 ⁽¹⁾	None
SET [m].i	Set bit of data memory	1 ⁽¹⁾	None
Branch			
JMP addr	Jump unconditionally	2	None
SZ [m]	Skip if data memory is zero	$1^{(2)}_{(2)}$	None
SZA [m]	Skip if data memory is zero with data movement to ACC	1 ⁽²⁾	None
SZ [m].i	Skip if bit i of data memory is zero	1 ⁽²⁾	None
SNZ [m].i	Skip if bit i of data memory is not zero	1 ⁽²⁾	None
SIZ [m]	Skip if increment data memory is zero	$1^{(3)}$	None
SDZ [m]	Skip if decrement data memory is zero	1 ⁽³⁾	None
SIZA [m]	Skip if increment data memory is zero with result	1 ⁽²⁾	None
SDZA [m]	in ACC Skip if decrement data memory is zero with result in ACC	1 ⁽²⁾	None
CALL addr	Subroutine call	2	None
RET	Return from subroutine	2	None
RET A,x	Return from subroutine and load immediate data to ACC	2	None



Mnemonic	Description	Instruction Cycle	Flag Affected
Table Read			
TABRDC [m]	Read ROM code (current page) to data memory and TBLH	2 ⁽¹⁾	None
TABRDL [m]	Read ROM code (last page) to data memory and TBLH	2 ⁽¹⁾	None
Miscellaneous			
NOP	No operation	1	None
CLR [m]	Clear data memory	1 ⁽¹⁾	None
SET [m]	Set data memory	1 ⁽¹⁾	None
CLR WDT	Clear Watchdog timer	1	TO,PD
CLR WDT1	Pre-clear Watchdog timer	1	$TO_{(4)}^{(4)}, PD_{(4)}^{(4)}$
CLR WDT2	Pre-clear Watchdog timer	1	$TO^{(4)}, PD^{(4)}$
SWAP [m]	Swap nibbles of data memory	1 ⁽¹⁾	None
SWAPA [m]	Swap nibbles of data memory with result in ACC	1	None
HALT	Enter power down mode	1	TO,PD

Notes: x: 8 bits immediate data

m: 7 bits data memory address

A: accumulator

i: 0~7 number of bits

addr: 11 bits program memory address

 $\sqrt{1}$: Flag(s) is affected

- -: Flag(s) is not affected
- ⁽¹⁾: If a loading to the PCL register occurs, the execution cycle of instructions will be delayed for one more cycle (4 system clocks).
- ⁽²⁾: If a skip to the next instruction occurs, the execution cycle of instructions will be delayed for one more cycle (4 system clocks). Otherwise the original instruction cycle(s) is unchanged.
- ⁽³⁾: ⁽¹⁾ and ⁽²⁾
- ⁽⁴⁾: The flags may be affected by the execution status. If the watchdog timer is cleared by executing the CLR WDT1 or CLR WDT2 instruction, the TO is set and the PD is cleared. Otherwise the TO and PD flags remain unchanged.



Instruction Definition

ADC A,[m]	Add da	ita me	mory a	nd car	ry to a	ccumu	ılator			
Description	The co are ade	The contents of the specified data memory, accumulator and the carry flag are added simultaneously, leaving the result in the accumulator.								
Operation	$ACC \leftarrow$	$ACC \leftarrow ACC \text{+}[m] \text{+}C$								
Affected flag(s)										
	TC2	TC1	ТО	PD	OV	Ζ	AC	С		
	-	-	-	-	\checkmark	\checkmark	\checkmark	\checkmark		
ADCM A,[m]	Add ac	cumul	ator aı	nd carr	y to da	ata me	mory			
Description	The co are add	ntents ded sin	of the nultan	specif eously,	ied da leavir	ta mei 1g the 1	mory, a result i	iccumi in the	lator and the carry f specified data memor	lag 'y.
Operation	[m] ← .	ACC+[m]+C	Ū		0			-	
Affected flag(s)										
	TC2	TC1	ТО	PD	OV	Z	AC	С		
	-	-	_	_	\checkmark	\checkmark	\checkmark	\checkmark		
ADD A,[m]	Add da	ita me	mory t	o accui	mulato	r				
Description	The co The re	ntents sult is	of the stored	specif in the	ied dat accum	ta mer nulator	nory a r.	nd the	accumulator are add	ed.
Operation	ACC ←	- ACC-	+[m]							
Affected flag(s)										
	TC2	TC1	ТО	PD	OV	Z	AC	С		
	-	-	-	-	\checkmark	\checkmark	\checkmark	\checkmark		
ADD A,x	Add in	nmedia	te data	a to ac	cumula	ator				
Description	The cor result	ntents in the	of the a accum	accum ulator.	ulatora	and th	e speci	fied da	ta are added, leaving	the
Operation	ACC ←	- ACC-	⊦x							
Affected flag(s)										
	TC2	TC1	ТО	PD	OV	Ζ	AC	С		
	_	_	_	_	\checkmark	\checkmark	\checkmark	\checkmark		



ADDM A,[m]	Add ad	cumul	ator to	data 1	memor	у			
Description	The co The re	ntents sult is	of the stored	specif in the	ied dat data r	ta mer nemor	nory a y.	nd the	accumulator are added.
Operation	$[m] \leftarrow$	ACC+[m]						
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	_	-	_	_	\checkmark	\checkmark	\checkmark	\checkmark	
AND A,[m]	Logica	l AND	accum	ulator	with d	lata m	emory		
Description	Data i logical	n the a _AND	iccumu operat	ılator a ion. Tl	and the	e speci Ilt is st	fied da tored in	ita me n the a	mory performs a bitwise ccumulator.
Operation	ACC ←	- ACC	"AND'	' [m]					
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	-	-	-	-	-	\checkmark	-	-	
AND A,x	Logica	l AND	imme	diate d	ata to :	accum	ulator		
Description	Data i cal_AN	in the ND ope	accum ration.	ulator The r	and t esult is	he spe s store	ecified d in th	data j e accu	performs a bitwise logi- mulator.
Operation	ACC ←	- ACC	"AND'	'x					
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	-	-	-	\checkmark	-	-	
ANDM A,[m]	Logica	l AND	data n	nemory	y with	accum	ulator		
Description	Data i logical	n the s _AND	pecifie operat	d data ion. Tl	n memo he resu	ory and 11t is st	l the a tored in	ccumu n the d	lator performs a bitwise ata memory.
Operation	$[m] \leftarrow$	ACC "A	AND"	[m]					
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	-	-	-	-	-	\checkmark	-	-	



CALL addr	Subroutine call								
Description	The instruction unconditionally calls a subroutine located at the indicated address. The program counter increments once to obtain the address of the next instruction, and pushes this onto the stack. The indicated address is then loaded. Program execution continues with the instruction at this address.								
Operation	$\begin{array}{l} Stack \leftarrow PC+1 \\ PC \leftarrow addr \end{array}$								
Affected flag(s)									
	TC2 TC1 TO PD OV Z AC	C C							
		-							
CLR [m]	Clear data memory								
Description	The contents of the specified data memory a	are cleared to zero.							
Operation	$[m] \gets 00H$								
Affected flag(s)									
	TC2 TC1 TO PD OV Z AC	C C							
		-							
CLR [m].i	Clear bit of data memory								
Description	The bit i of the specified data memory is cle	ared to zero.							
Operation	$[m].i \gets 0$								
Affected flag(s)									
	TC2 TC1 TO PD OV Z AC	C C							
		-							
CLR WDT	Clear watchdog timer								
Description	The WDT and the WDT Prescaler are clea power down bit (PD) and time-out bit (TO)	red (re-counting from zero). The are cleared.							
	WDT and WDT Prescaler $\leftarrow 00H$ PD and TO $\leftarrow 0$								
Operation	PD and TO $\leftarrow 0$								
Operation Affected flag(s)	PD and TO $\leftarrow 0$								
Operation Affected flag(s)	PD and TO $\leftarrow 0$ TC2 TC1 TO PD OV Z AC	C							



CLR WDT1	Precle	ar wate	chdog	timer					
Description	The PD, TO flags, WDT and the WDT Prescaler are cleared (re-counting from zero), if the other preclear WDT instruction had been executed. Only execution of this instruction without the other preclear instruction sets the indicating flag which implies this instruction was executed. The PD and TO flags remain unchanged.								
Operation	WDT a PD an	WDT and WDT Prescaler \leftarrow 00H* PD and TO \leftarrow 0*							
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	_	0*	0*	-	_	-	-	
CLR WDT2	Precle	ar wate	chdog 1	timer					
Description	The PI zero), i tion of indicat flags r	D, TO f if the o f this ting fla emain	lags, W ther p instru g whio uncha	/DT an reclear ction ch impl nged.	d the V WDT withou lies thi	VDT P instru t the s instr	rescale ction h other ruction	er are c ad bee precles was es	eleared (re-counting from en executed. Only execu- ar instruction, sets the xecuted. The PD and TO
Operation	WDT and WDT Prescaler $\leftarrow 00H^*$ PD and TO $\leftarrow 0^*$								
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	_	0*	0*	_	_	_	_	
CPL [m]	Compl	ement	data n	nemor	у				
Description	Each b ment). versa.	it of th Bits v	e spec vhich	ified d previo	ata me usly co	mory i ntain	s logica a one	ally co are ch	mplemented (1's comple- anged to zero and vice-
Operation	$[m] \leftarrow$	[<u>m</u>]							
Affected flag(s)									7
	TC2	TC1	ТО	PD	OV	Z	AC	С	4
	-	-	-	-	-	\checkmark	-	-	



CPLA [m]	Compl	ement	data n	nemor	y and p	olace re	esult ir	1 accur	nulator
Description	Each h ment). vice-ve conten	oit of th Bits ersa. Th its of th	e spec which he con ne data	ified d previo npleme n memo	ata me ously o ented r ory ren	emory i contain result i nain ui	s logica ied a s store nchang	ally cor one ar ed in t ged.	mplemented (1's comple- e changed to zero and he accumulator and the
Operation	ACC ←	– [<u>m</u>]							
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	-	-	_	_	-	\checkmark	_	_	
DAA [m]	Decim	al-Adju	ist acc	umula	tor for	additi	on		
Description	The ac The ac code a accum the ori is set; in the	ccumula cumula nd an ulator ginal v otherw data m	ator va ator is intern is grea value in vise the nemory	alue is divide al car ater th f the or e origin 7 and o	adjust ed into ry (AC an 9. 7 riginal nal val nly the	ed to t two nil (1) will The BC value ue rem e carry	he BC bbles. I l be cr D adju is grea nains u flag (C	D (Bin Each n reated istmer iter tha inchan C) may	ary Code Decimal) code. ibble is adjusted to BCD if the low nibble of the it is done by adding 6 to an 9 or a carry (AC or C) ged. The result is stored be affected.
Operation	If (AC) then () else ([n If (AC) then () else ([n	C.3~A([m].3~[n].3~[n C.7~A([m].7~[n].10].10].10].10].10].10].10].10].10].10	CC.0) > m].0) ← n].0) ← CC.4)+ m].4) ←	>9 or A (ACC (ACC AC1 > (ACC (ACC	C=1 C.3~AC C.3~AC 9 or C= C.7~AC	CC.0)+(C.0), A =1 CC.4)+(C.4)+A	6, AC1 C1=0 6+AC1 C1, C:	=AC , C=1 =C	
Affected flag(s)									_
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	-	-	-	-	-		-	\checkmark	
DEC [m]	Decrei	nent da	ata me	emory					
Description	Data i	n the s	pecifie	d data	memo	ry is d	ecreme	ented b	oy one
Operation Affected flag(s)	[m] ←	[m]–1							
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	_	_	_	-	-	\checkmark	_	_	



DECA [m]	Decrer	nent da	ata me	mory a	and pla	ice res	ult in a	accumu	ılator
Description	Data i in the	n the sj accum	pecifie ulator.	d data The co	memo ontents	ry is d s of the	ecreme e data i	ented b memor	y one, leaving the result y remain unchanged.
Operation	ACC ←	- [m]–1	l						
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	-	-	-	-	-	\checkmark	-	-	
HALT	Enter	power	down i	node					
Description	This ir	Istructi	ion sto	ps prog	gram e	xecuti	on and	turns	off the system clock. The
F	conten cleared cleared	ts of th 1. The 1.	e RAN power	A and r down	egiste bit (Pl	rs are D) is s	retaine et and	ed. The the W	WDT and prescaler are /DT time-out bit (TO) is
Operation	$\begin{array}{l} PC \leftarrow \\ PD \leftarrow \\ TO \leftarrow \end{array}$	PC+1 1 0							
Affected flag(s)									_
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	_	_	0	1	_	_	_	_	
INC [m]	Increm	ient da	ta me	mory					
Description	Data i	n the s	pecifie	d data	memo	ry is ii	ncreme	ented b	y one.
Operation	$[m] \leftarrow$	[m]+1							
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	_	-	-	-	\checkmark	-	-	
INCA [m]	Incren	ient da	ta mei	morv a	nd pla	ce resi	ılt in a	ccumu	lator
Description	Data i in the	n the sj accum	pecifie ulator.	d data The co	memo ontents	ry is ii s of the	ncreme e data i	ented b memor	y one, leaving the result y remain unchanged.
Operation	ACC ←	- [m]+1	L						
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	_	-	-	\checkmark	-	-	



JMP addr	Direct Jump							
Description	Bits $0~9$ of the program counter are replaced with the directly-specified address unconditionally, and control passed to this destination.							
Operation	$PC \leftarrow addr$							
Affected flag(s)								
	TC2 TC1 TO PD OV Z AC C							
MOV A,[m]	Move data memory to accumulator							
Description	The contents of the specified data memory is copied to the accumulator.							
Operation	$ACC \leftarrow [m]$							
Affected flag(s)								
	TC2 TC1 TO PD OV Z AC C							
MOV A,x	Move immediate data to accumulator							
Description	The 8-bit data specified by the code is loaded into the accumulator.							
Operation	$ACC \leftarrow x$							
Affected flag(s)								
	TC2 TC1 TO PD OV Z AC C							
MOV [m],A	Move accumulator to data memory							
Description	The contents of the accumulator is copied to the specified data memory (one of the data memory locations).							
Operation	$[m] \leftarrow ACC$							
Affected flag(s)								
	TC2 TC1 TO PD OV Z AC C							
NOP	No operation							
Description	No operation is performed. Execution continues with the next instruction.							
Operation	$PC \leftarrow PC+1$							
Affected flag(s)								
	TC2 TC1 TO PD OV Z AC C							
OR A,[m]	Logical OR accumulator with data memory							

	5							HT48CA0
Description	Data in t memory stored in	the accur locations the accu	nulator) perfor mulator	and th ms a l	he spe pitwise	cified logica	data n 1_OR	nemory (one of the data operation. The result is
Operation	$ACC \leftarrow A$	CC "OR"	[m]					
Affected flag(s)								
	TC2 T	С1 ТО	PD	OV	Z	AC	С	
	_		-	-	\checkmark	_	_	
OR A,x	Logical O	R immed	liate da	ta to a	ccumu	lator		
Description	Data in th operation	ne accum 1. The res	ulator a sult is st	nd the tored ir	specifi 1 the a	ed data ccumu	a perfo lator.	rms a bitwise logical_OR
Operation	$ACC \leftarrow A$	CC "OR"	x					
Affected flag(s)								
	TC2 T	С1 ТО	PD	OV	Ζ	AC	С	
	-		-	-	\checkmark	-	-	
ORM A,[m]	Logical O	R data n	nemory	with a	ccumu	lator		
Description	Data in accumula the data	the data tor perfo memory.	memo rms a b	ory (on itwise	e of tl logical	he dat _OR op	a men peratio	nory locations) and the n. The result is stored in
Operation	$[m] \leftarrow AC$	C "OR"	m]					
Affected flag(s)								
	TC2 T	С1 ТО	PD	OV	Z	AC	С	
	-		-	-	\checkmark	-	-	
RET	Return fr	om subro	outine					
Description	The prog tion.	ram cour	nter is r	estored	l from	the sta	ick. Th	is is a two-cycle instruc-
Operation	$\text{PC} \leftarrow \text{Sta}$	nck						
Affected flag(s)								
	TC2 T	C1 TO	PD	OV	Z	AC	С	
	-		-	-	-	-	-	



RET A,x	Return	n and p	lace ir	nmedia	ate dat	a in ac	cumul	ator	
Description	The pr with t	rogram he spec	count cified 8	er is re -bit im	stored	from t te data	the sta a.	ck and	the accumulator loaded
Operation	PC ← ACC ∢	Stack – x							
Affected flag(s)									
	TC2	TC1	то	PD	OV	Z	AC	С]
	_	-	_	_	_	_	-	_	_
RETI	Return	n from	interru	ıpt					
Description	The pr setting registe	rogram g the E er INT(count MI bit C).	er is ro t. EMI	estored is the	l from enabl	the sta e mast	ack, ar er (glo	nd interrupts enabled by abal) interrupt bit (bit 0;
Operation	PC ← EMI ←	Stack – 1							
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С]
	_	_	_	_	_	_	_	_	
RL [m]	Rotate	e data r	nemor	y left					
Description	The co rotate	ontents d into l	of the oit 0.	specif	ied da	ta mer	nory is	s rotat	ed left one bit with bit 7
Operation	[m].(i+ [m].0 «	-1) ← [ı ← [m].7	n].i; [n ⁄	n].i:bit	i of the	e data	memo	ry (i=0	~6)
Affected flag(s)									
-	TC2	TC1	ТО	PD	OV	Z	AC	С]
	_	_	_	_	_	_	_	_	-
RLA [m]	Rotate	e data r	nemor	y left a	nd pla	ce rest	ult in a	iccumi	ılator
Description	Data i into bi	n the s it 0, lea	pecifie ving t	ed data he rota	n memo nted res	ory is a sult in	rotated the ac	l left o cumul	ne bit with bit 7 rotated ator. The contents of the
	data n	nemory	remai	in uncł	nanged	•			
Operation	ACC.(ACC.0	$i+1) \leftarrow i+1) \leftarrow m$	[m].i; .7	[m].i:bi	t i of tl	ne dat	a mem	ory (i=	0~6)
Affected flag(s)									~
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	_	-	-	-	_	_	-	-]



RLC [m]	Rotate data	memor	y left t	hrougł	ı carry	7			
Description	The content rotated left rotated into	s of the one bit the bit	e speci . Bit 7 0 posit	fied da ' repla tion.	ta me ces th	mory a e carry	nd the bit; t	e carry flag are together he original carry flag is	
Operation	$[m].(i+1) \leftarrow$ $[m].0 \leftarrow C$ $C \leftarrow [m].7$	m].i; [n	n].i:bit	i of the	e data	memoi	ry (i=0	~6)	
Affected flag(s)									
	TC2 TC1	ТО	PD	OV	Z	AC	С		
		-	-	-	-	-	\checkmark		
RLCA [m]	Rotate left t	hrough	carry	and pla	ace res	sult in a	accum	ulator	
Description	Data in the s one bit. Bit 7 bit 0 position of the data r	pecifie 7 replac 1. The r nemory	d data ces the otated r remai	memon carry l result n unch	ry and bit and is store anged	the car l the or ed in th l.	rry flag iginal ie accu	; are together rotated left carry flag is rotated into mulator but the contents	
Operation	ACC.(i+1) \leftarrow [m].i; [m].i:bit i of the data memory (i=0~6) ACC.0 \leftarrow C C \leftarrow [m].7								
Affected flag(s)									
	TC2 TC1	ТО	PD	OV	Z	AC	С		
		-	-	-	-	-	\checkmark		
RR [m]	Rotate data	memor	y right						
Description	The content 0 rotated to	s of the bit 7.	specif	ied dat	a men	nory ar	e rota	ted right one bit with bit	
Operation	$[m].i \leftarrow [m].i$ $[m].7 \leftarrow [m].i$	i+1); [n 0	n].i:bit	i of the	e data	memoi	ry (i=0	~6)	
Affected flag(s)									
	TC2 TC1	ТО	PD	OV	Z	AC	С		
		_	_	_	_	_	_		
RRA [m]	Rotate right	and pl	ace res	ult in a	accum	ulator			
Description	Data in the into bit 7, le data memor	specifie aving t y remai	d data he rota in uncl	memo nted res nanged	ry is r sult in	otated the ac	one bi cumul	t right with bit 0 rotated ator. The contents of the	
Operation	$\begin{array}{l} \text{ACC.(i)} \leftarrow [\text{r} \\ \text{ACC.7} \leftarrow [\text{m} \end{array} \end{array}$	n].(i+1)].0	; [m].i:	bit i of	the da	ita mer	nory (i	l=0∼6)	
Affected flag(s)								7	
	TC2 TC1	ТО	PD	OV	Z	AC	С		
		-	-	-	-	-	-		



RRC [m]	Rotate	data n	nemor	y right	throu	gh car	ry		
Description	The co rotated rotated	ontents d one b d into t	of the oit righ he bit	e specif nt. Bit 7 posit	fied da 0 repla tion.	ta me aces th	mory a ne carr	nd th y bit;	e carry flag are together the original carry flag is
Operation	[m].i ← [m].7 ← C ← [n	– [m].(i – C n].0	+1); [n	n].i:bit	i of the	e data	memoi	ry (i=0	~6)
Affected flag(s)	Ľ								
U U	TC2	TC1	ТО	PD	OV	Z	AC	С]
	_	_	_	_	_	_	_	\checkmark	
RRCA [m]	Rotate	right t	hroug	h carry	y and p	olace re	esult ir	n accu	mulator
Description	Data o bit rigl the bi conten	f the sp ht. Bit (t 7 pos its of th	ecifieo) repla sition. 1e data	d data aces the The r memo	memor e carry otated ory ren	y and t bit and resul nain u	the car d the or t is st nchang	ry flag riginal ored i ged.	; are together rotated one l carry flag is rotated into n the accumulator. The
Operation	$\begin{array}{l} ACC.i \\ ACC.7 \\ C \leftarrow [n] \end{array}$	← [m]. ← C n].0	(i+1);	m].i:bi	t i of tl	he data	a mem	ory (i=	0~6)
Affected flag(s)									_
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	-	-	-	-	-	-	-	\checkmark]
SBC A.[m]	Subtra	act data	a mem	orv an	d carry	from	accum	ulator	
Description	The co flag ar accum	ontents re toget ulator.	of the her su	specif btract	ied dat ed fron	a men n the a	nory ar accumu	nd the ılator,	complement of the carry leaving the result in the
Operation	ACC ←	- ACC+	-[<u>m</u>]+C	2					
Affected flag(s)									_
	TC2	TC1	ТО	PD	OV	Ζ	AC	С	
	-	-	-	-	\checkmark	\checkmark	\checkmark	\checkmark]
SBCM A,[m]	Subtra	act data	a mem	ory an	d carry	from	accum	ulator	
Description	The co flag ar data n	ontents re toget nemory.	of the her su	specif btract	ied dat ed fron	a men n the a	nory ar accumu	nd the 1lator,	complement of the carry leaving the result in the
Operation	$[m] \leftarrow$	ACC+[m]+C						
Affected flag(s)	[7
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	-	-	\checkmark	\checkmark	\checkmark	\checkmark	



SDZ [m]	Skip if	decrer	nent d	ata me	emory i	is zero			
Description	The co result followi discaro makes	ontents is zero ng ins ded ano a 2-cyo	of the o, the tructic d a du cle ins	e speci next i on, fetc mmy c tructio	fied da instruc ihed du ycle is n. Oth	ta men tion is uring t replac erwise	mory a s skipp the cur ed to g procee	are dec ped. If rrent i get the ed with	remented by one. If the the result is zero, the nstruction execution, is proper instruction. This the next instruction.
Operation	Skip if	([m]–1	l)=0, [r	n] ← ([m]-1)				
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	_	-	-	-	-	-	-	-	
SDZA [m]	Decrer	nent da	ata me	emory a	and pla	ice res	ult in A	ACC, s	kip if zero
Description	The co result accum the foll discare makes	ontents is zero ulator lowing ded and a 2-cyo	of the b, the but th instru d a du cle ins	e speci next ir e data ction, f mmy c tructio	fied da nstruct memo etched ycle is n. Oth	ta men ion is ry rem durin replac erwise	mory a skippe ains u g the c ed to g procee	are dec ed. The nchang urrent get the ed with	remented by one. If the e result is stored in the ged. If the result is zero, instruction execution, is proper instruction, that the next instruction.
Operation	Skip if	[[m]–1	l)=0, A	CC ←	([m]–1])			
Affected flag(s)	-								
0	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	-	-	-	-	-	-	
SET [m]	Set da	ta men	nory						
Description	Each b	oit of th	ne spec	ified d	ata me	mory	is set t	o one.	
Operation	[m] ←	FFH							
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	-	-	-	-	-	-	
SET [m].i	Set bit	of dat	a mem	ory					
Description	Bit i of	f the sp	ecified	l data	memor	y is se	t to on	e.	
Operation	[m].i ←	- 1							
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	-	-	-	-	-	-	
	L		1						1



SIZ [m]	Skip if	f increr	nent d	ata me	mory i	s zero			
Description	The co is zero execut instrue instrue	ntents o, the ion, is ction. T ction.	of the s followi discau Fhis is	specific ing ins rded a a 2-cy	ed data structio nd a d /cle ins	n memo on, fet lummy structi	ory is in ched d y cycle on. Otl	ncreme luring is rep herwis	nted by one. If the result the current instruction laced to get the proper e proceed with the next
Operation	Skip if	f ([m]+1	1)=0, [1	m] ← ([m]+1)				
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	-	-	-	-	-	-	
SIZA [m]	Incren	nent da	ata me	mory a	nd pla	ce res	ult in A	CC, sł	cip if zero
Description	The co is zero accum followi discaro is a 2-o	ntents o, the ulator. ing ins ded and cycle in	of the s next i The da tructic d a dua	specific nstruc ata me on, fetc mmy c ion. Of	ed data tion is mory r ched du ycle is cherwis	a memo skipj emain uring replac se proc	ory is ir ped an is unch the cur ced to g ceed wi	ncreme d the anged rrent i get the th the	nted by one. If the result result is stored in the If the result is zero, the nstruction execution, is proper instruction. This next instruction.
Operation	Skip if	f ([m]+1	1)=0, A	CC ←	([m]+1)			
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	_	-	-	-	-	-	-	-	
SNZ [m].i	Skip if	f bit i o	f the d	ata me	emory i	s not z	zero		
Description	If bit i If bit i the cur get the the nex	of the s of the o rrent in proper xt instr	specifie data m structi r instru uction.	ed data emory on exec iction.	memoris not z cution, This is	ry is n zero, tl is disc a 2-cy	ot zero, he follov arded a cle inst	the ne wing ir ind a d ruction	ext instruction is skipped. Istruction, fetched during ummy cycle is replaced to a. Otherwise proceed with
Operation	Skip if	f [m].i≠	0						
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	-	-	-	-	-	-	-	
SUB A,[m]	Subtra	act data	a mem	ory fro	m accu	ımulat	tor		
Description	The sp tor. lea	ecified aving tl	data i he resi	memor ılt in t	y is sul he accu	btract imula	ed fron tor.	n the co	ontents of the accumula-
Operation	ACC ←	- ACC-	+[<u>m</u>]+1						
Affected flag(s)									
5	TC2	TC1	ТО	PD	OV	Z	AC	С	
	_	-	-	_	\checkmark		\checkmark	\checkmark	



SUBM A,[m]	Subtra	act data	a mem	ory fro	m accu	ımulat	tor					
Description	The sp tor, lea	The specified data memory is subtracted from the contents of the accumula- tor, leaving the result in the data memory.										
Operation	$[m] \leftarrow$	ACC [<u>n</u>]+1									
Affected flag(s)												
	TC2	TC1	ТО	PD	OV	Ζ	AC	С				
	-	-	-	-	\checkmark	\checkmark	\checkmark	\checkmark	I			
SUB A,x	Subtra	act imn	nediate	e data :	from a	ccumu	lator					
Description	The in the acc	The immediate data specified by the code is subtracted from the contents of the accumulator, leaving the result in the accumulator.										
Operation	ACC ←	- ACC-	$+\overline{x}+1$									
Affected flag(s)												
	TC2	TC1	ТО	PD	OV	Z	AC	С				
	_	_	_	_	\checkmark	\checkmark		\checkmark	l			
SWAP [m]	Swap	nibbles	withi	n the d	ata me	emory						
Description	The lov data m	w-orde nemory	r and h locati	igh-or ons) ar	der nib e inter	bles of chang	f the sp ged.	ecified	data memory (one of	the		
Operation	[m].3~	[m].0 ‹	→ [m].7	7~[m].4	1							
Affected flag(s)												
	TC2	TC1	ТО	PD	OV	Z	AC	С				
	-	_	_	_	_	_	_	_	l			
SWAPA [m]	Swap	data m	emory	and p	lace re:	sult in	accum	ulator				
Description	The lo interch memor	ow-orde nanged ry rema	er and , writi ain une	high- ng the change	order result ed.	nibble to the	s of th accum	ne speo ulator.	ified data memory a The contents of the data	are ata		
Operation	ACC.3 ACC.7	~ACC. ~ACC.	$0 \leftarrow [n] 4 \leftarrow [n]$	n].7~[n n].3~[n	1].4 1].0							
Affected flag(s)									_			
	TC2	TC1	ТО	PD	OV	Z	AC	С				
	-	-	-	-	-	_	-	-]			



SZ [m]	Skip if	data n	nemor	y is zei	0				
Description	If the c fetchec cycle is Otherv	ontent l durin s repla vise pr	s of the g the c ced to oceed	e speci current get the with th	fied da t instru e prope ne next	ta mer action er instr instru	nory is execut ruction iction.	zero, t ion, is 1. This	he following instruction, discarded and a dummy is a 2- cycle instruction.
Operation	Skip if	[m]=0							
Affected flag(s)									
	TC2	TC1	то	PD	OV	Ζ	AC	С	
	-	_	-	-	-	-	-	-	
SZA [m]	Move o	lata m	emory	to AC	C, skip	if zero)		
Description	The content content tion ex instruct instruct	ntents ts is ze ecution ction. T ction.	of the ro, the n, is di This is	specifie follow scarde a 2-cy	ed data ing ins d and a cle ins	a memo tructio a dum structio	ory is c on, fetc my cyc on. Otl	opied (hed du le is ro herwis	to the accumulator. If the bring the current instruc- eplaced to get the proper e proceed with the next
Operation	Skip if	[m]=0							
Affected flag(s)									
	TC2	TC1	то	PD	OV	Z	AC	С	
	-	_	-	-	-	-	-	-	
SZ [m].i	Skip if	bit i of	f the d	ata me	mory i	s zero			
Description	If bit i during replace procee	of the s the cu ed to ge d with	specifie rrent t the p the ne	ed data instruc proper i ext inst	n memo ction ex instruc ruction	ory is z kecutic tion. T n.	ero, th on, is d 'his is a	e follov iscard a 2-cycl	wing instruction, fetched ed and a dummy cycle is le instruction. Otherwise
Operation	Skip if	[m].i=	0						
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	_	-	-	-	-	_	-	-	
TABRDC [m]	Move I	ROM co	ode (cu	Irrent j	page) t	o TBL	H and	data n	nemory
Description	The lo (TBLP to TBL	w byte) is mo .H dire	e of RO ved to ectly.	OM coo the spo	le (cur ecified	rent p data n	age) a nemory	ddress y and t	ed by the table pointer he high byte transferred
Operation	[m] ← TBLH	$\begin{array}{l} ROM \ c \\ \leftarrow ROI \end{array}$	ode (la M code	ow byte e (high	e) byte)				
Affected flag(s)									1
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	_	_	-	-	_	-	_	



TABRDL [m]	Move	ROM c	ode (la	st page	e) to TI	BLH a	nd data	a men	nory
Description	The lo is mov	w byte ed to tł	of ROI ne data	M code 1 memo	(last p ory and	age) a the hi	ddress igh byt	ed by e tran	the table pointer (TBLP) sferred to TBLH directly.
Operation	[m] ← TBLH	$\begin{array}{l} \text{ROM of } \\ \leftarrow \text{RO} \end{array}$	ode (la M code	ow byte e (high	e) byte)				
Affected flag(s)									
	TC2	TC1	то	PD	OV	Z	AC	С	
	-	-	-	-	_	-	-	_	
XOR A,[m]	Logica	l XOR	accum	ulator	with d	ata m	emory		
Description	Data i logical	n the a Exclus	ccumu sive_O	ılator a R oper	nd the ation a	indicand the	ated da e resul	ata me t is sto	emory performs a bitwise pred in the accumulator.
Operation	ACC <	- ACC	"XOR"	[m]					
Affected flag(s)									
	TC2	TC1	ТО	PD	OV	Z	AC	С	
	-	_	_	_	_	\checkmark	_	_]
XORM A,[m]	Logica	l XOR	data n	nemory	with a	accum	ulator		
Description	Data i logical zero fl	n the i Exclus ag is af	ndicat sive_O fected	ed data R opera	a mem ation. '	ory an The re	nd the a sult is a	accum stored	ulator perform a bitwise in the data memory. The
Operation	[m] ←	ACC "Z	XOR" [m]					
Affected flag(s)									
U	TC2	TC1	то	PD	OV	Z	AC	С]
	_	_	_	_	_	V	_	_	-
						•			
XOR A,x									
	Logica	l XOR	immeo	liate da	ata to a	accum	ulator		
Description	Logica Data i Exclus flag is	l XOR n the tl sive_OI affecte	immeo he accu R opera d.	liate da umulat ation. '	ata to a or and The re	accum the sp sult is	ulator pecified stored	l data l in th	perform a bitwise logical ne accumulator. The zero
Description Operation	Logica Data i Exclus flag is ACC \leftarrow	l XOR n the tl sive_OF affecte – ACC	immeo he accu R opera d. "XOR"	liate da umulat ation. 7	ata to a or and The re	accum the sp sult is	ulator becified stored	l data l in th	perform a bitwise logical le accumulator. The zero
Description Operation Affected flag(s)	Logica Data i Exclus flag is ACC \leftarrow	ll XOR n the tl sive_OH affecte – ACC	immeo he accu R opera d. "XOR"	liate da umulat ation. 7 x	ata to a or and The re	accum the sp sult is	ulator becified stored	l data l in th	perform a bitwise logical ae accumulator. The zero
Description Operation Affected flag(s)	Logica Data i Exclus flag is ACC \leftarrow TC2	l XOR n the tl sive_OF affecte - ACC TC1	immeo he accu R opera d. "XOR" TO	liate da umulat ation. 7 x PD	ata to a or and The re OV	accum the sp sult is Z	ulator pecified stored	l data l in th C	perform a bitwise logical le accumulator. The zero