HOLTEK

HT6220
Remote Controller

HT6220 Specification

Features

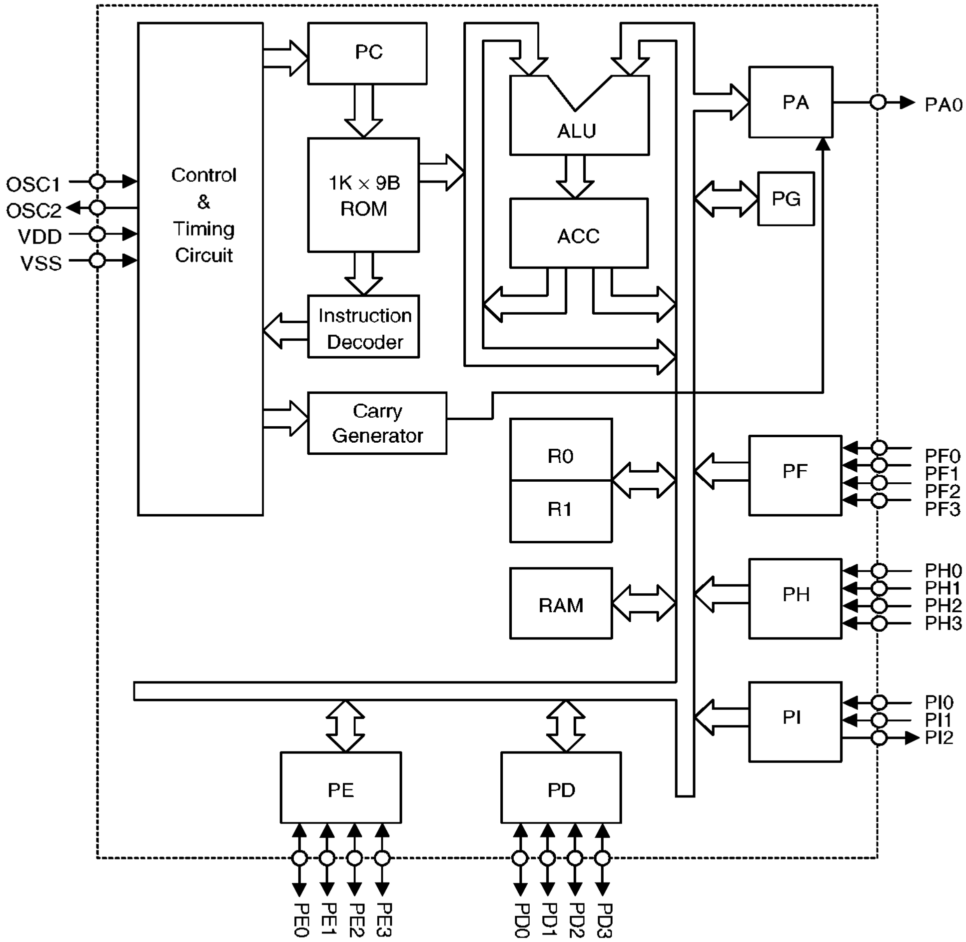
- 455KHz resonator oscillator
- Low operating voltage VDD=1.8V~3.5V
- Low power consumption IDD<1μA in HALT mode
- 32 × 4 Data RAM
- 1K×9 Program ROM
- 1 carrier output line PA0
- 2 bidirectional I/O ports PE and PD
- 10 input lines PH, PF, PI0, PI1
- 1 output line PI2
- 72 powerful instructions
- 2 working registers R1 and R0
- 1 internal port PG
- All instructions in 1 or 2 machine cycles
- 8-bit table read instructions
- HALT function

General Description

The HT6220 is a 4-bit single-chip microcomputer specially designed for the remote control transmitter.

The HT6220 can be used as an infrared remote control transmitter for TV, VCR, stero components, cassette decks, air conditioners and other applications.

Block Diagram



Note:

ACC: Accumulator

PC: Program Counter

R0~R1: Working Registers

PG: Internal Port

PA0: Carrier Ouput

PD, PE: I/O ports

PF, PH: Input Ports

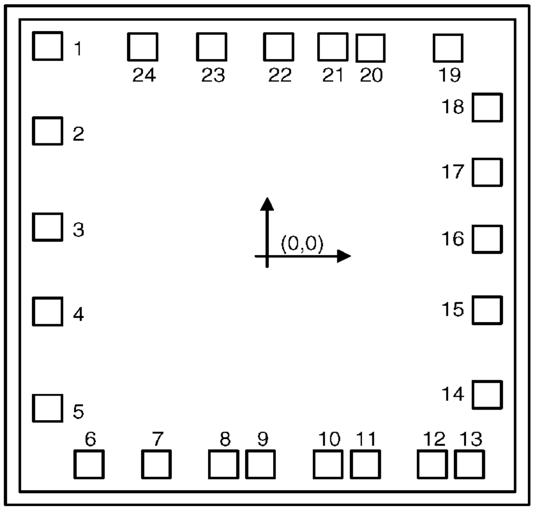
PI0, PI1: Input Lines

PI2: Output Line

Pad Description

Pad No.	Pad Name	I/O	Mask Option	Functional Description
8~11	PH0~PH3	I	—	4-bit input port with pull-low and wake-up
12~15	PF0~PF3	I	—	4-bit input port with pull-low and wake-up
16	PA0	O	38KHz, 1/3 Duty or 57KHz, 1/2 Duty	Special one bit carrier output
17	VDD	I	—	Positive power supply
18	PI0	I	—	One bit input
19 20	OSC2 OSC1	O I	—	OSC1 and OSC2 are connected to an external resonator for the internal system clock.
21	VSS	I	—	Negative power supply (GND)
7	PI1	I	—	One bit input, pull-high or pull-low controlled by the software
2~1 24~23	PE0~PE3	I/O	—	4-bit bidirection I/O port with pull-low
6~3	PD0~PD3	I/O	—	4-bit bidirection I/O port with pull-low
22	PI2	O	—	One bit output with Tri-state

Pad Coordinates



Chip size: 2560 × 2110 (μm)^2

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

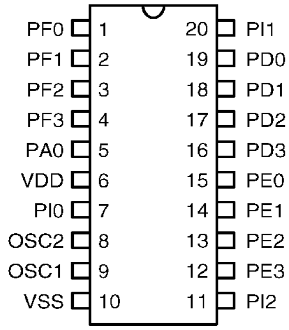
unit: μm

Pad No.	Pad Name	X	Y	Pad No.	Pad Name	X	Y
1*	PE1	–1083.5	832.5	13*	PF1	996	–829.5
2*	PE0	–1083.5	495.5	14*	PF2	1083.5	–551.5
3*	PD3	–1083.5	115.5	15	PF3	1083.5	–217.5
4*	PD2	–1083.5	–221.5	16*	PA0	1083.5	66.5
5*	PD1	–1083.5	–604.5	17*	VDD	1083.5	331.5
6*	PD0	–881	–829.5	18	PI0	1083.5	588.5
7	PI1	–549.5	–829.5	19*	OSCO	893	824.5
8	PH0	–218	–829.5	20*	OSCI	509.5	824.5
9	PH1	–33	–829.5	21*	VSS	324.5	829.5
10	PH2	299	–829.5	22	PI2	57	829.5
11	PH3	484	–829.5	23*	PE3	–274	829.5
12*	PF0	816	–829.5	24*	PE2	–615	829.5

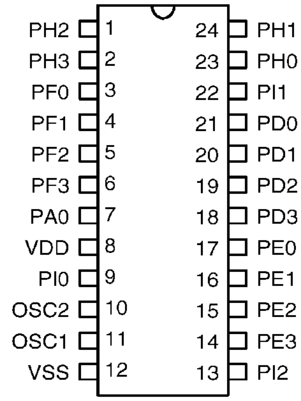
* The test pin, these pins must bonded out in package for testing.

Package & Pin Assignment

20 Pin DIP/SOP package



24 Pin SOP/SDIP package



Absolute Maximum Ratings

Parameter	Symbol	Minimum	Maximum	Unit
Supply voltage	V_{DD}	–0.3	6	V
Input voltage	V_I	V_{SS}–0.3	V_{DD}+0.3	V
Storage temperature	T_{STG}	–50	125	°C
Operating temperature	T_{OP}	0	70	°C

D.C. Characteristics

(Ta=25°C)

Symbol	rameter	Test Condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
V_{DD}	Operating voltage	—	—	1.8	3	3.5	V
I_{DD}	Operating current	3V	F_{OSC}=455KHz No load	—	200	400	μA
I_{STB}	Stand-by current	3V	No load HALT mode	—	0.1	1	μA
I_{OH1}	Output source current for PA0	3V	V_{OH}=2.7V	–2	–4	—	mA
I_{OL1}	Ouput sink current for PA0	3V	V_{OL}=0.3V	20	50	—	μA
I_{OH2}	Output source current for PI2	3V	V_{OH}=2.7V	–20	–60	—	μA

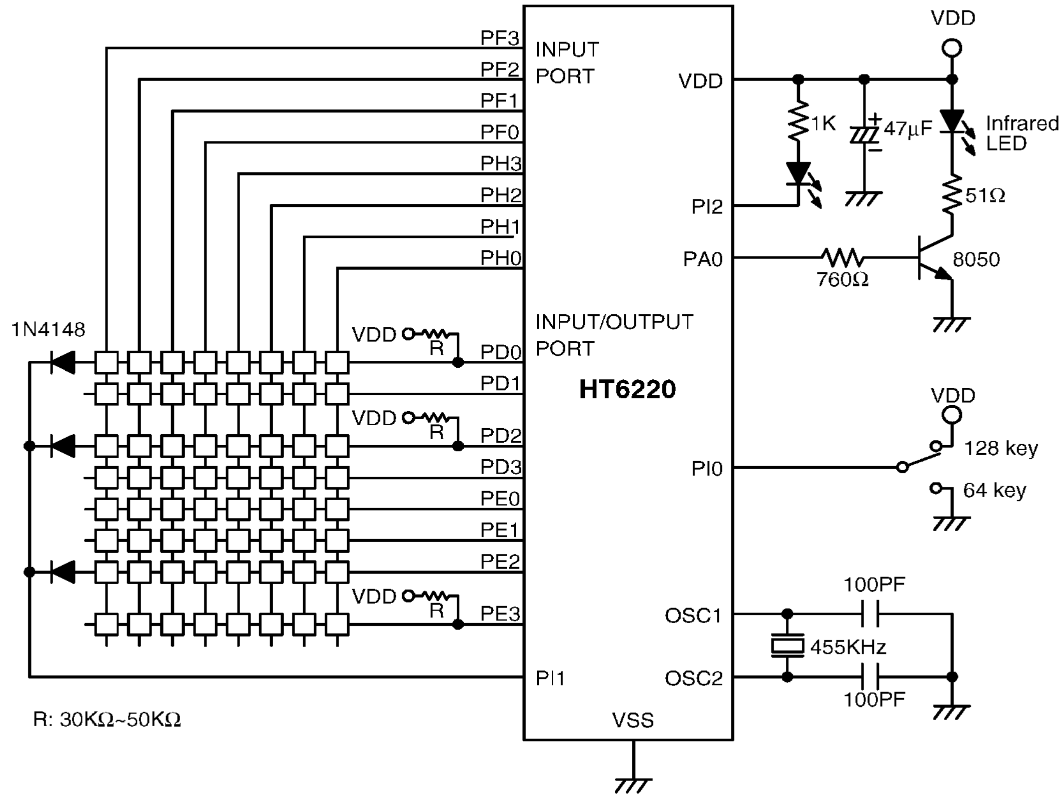
Symbol	rameter	Test Condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
I_{OL2}	Output sink current for PI2	3V	V_{OL}=0.3V	1	2	—	mA
I_{OH3}	Output source current for PD and PE ports	3V	V_{OH}=2.7V	–1	–2	—	mA
I_{OL3}	Output sink current for PD and PE ports	3V	V_{OL}=0.3V	20	50	—	μA
V_{IH1}	Input high voltage for PF and PH port	3V	—	1.9	—	3	V
V_{IL1}	Input low voltage for PF and PH port	3V	—	0	—	1.1	V
V_{IH2}	Input high voltage for PD and PE ports	3V	—	1.1	—	3	V
V_{IL2}	Input low voltage for PD and PE ports	3V	—	0	—	0.6	V
V_{IH3}	Input high voltage for PI1, PI0	3V	—	1.1	—	3	V
V_{IL3}	Input low voltage for PI1, PI0	3V	—	0	—	0.6	V
R_{PH}	Input pull-high resistance for PI1	3V	V_{IN}=0V	100	200	400	KΩ
R_{PL1}	Input pull-low resistance for PI1	3V	V_{IN}=3V	70	150	250	KΩ
R_{PL2}	Input pull-low resistance for PF and PH ports	3V	V_{IN}=3V	250	500	750	KΩ
R_{PL3}	Input pull-low resistance for PD and PE ports	3V	V_{IN}=3V	100	200	300	KΩ

A.C. Characteristics

(Ta=25°C)

Symbol	Parameter	Test Condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
f_{OSC}	System frequency	3V	455KHz ceramic resonator	—	455	—	KHz
t_{CY}	Instruction cycle time	3V	f_{SYS}=455KHz	—	8.8	—	μS
f_{CARRY}	Carrier frequency of PA0	3V	fSYS=455KHz	—	38/57	—	KHz

Application Diagram



SYSTEM ARCHITECTURE

Program Counter - PC

The program counter (PC) is a binary counter organized by 10 bits, namely PC0~PC9. It addresses the program memory (ROM) with 1024 addresses at maximum.

The program counter (PC) is incremented by 1 or 2 each time an instruction is executed.

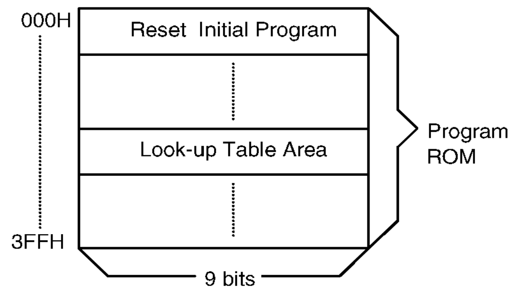
When executing the jump instruction (JMP, JNZ, JC, JB...) or initial reset, the program counter (PC) is loaded with the corresponding address data. For jump and branch instructions, the address space is capable of directly specifying 1024 addresses (jump bit instructions are excluded).

Program Memory - ROM

The program memory is used to store the executed program and non-volatile data. It is organized with 1024 × 9 bits and is addressed by the program counter. There are some special locations in program memory as described below:

Location 0
After the power is turned on or the system is initiated, the first instruction is fetched by the processor from location 0. Care must be taken when preparing the program.

Table location
The look-up table can be located at any ROM position. The instruction "READn MA" is available when reading the table and transferring the table data to the ACC and data memory which are addressed by the register pair R1 and R0.



READ0 MA	Page0 (000H~0FFH)
READ1 MA	Page1 (100H~1FFH)
READ2 MA	Page2 (200H~2FFH)
READ3 MA	Page3 (300H~3FFH)

Look-up Table Areas

Program Memory

Working Registers - R1,R0

The working registers consist of register R0 with 4 bits wide and register R1 with 1 bit wide. They are usually used to store the frequently accessed intermediate results. The working register R0 can operate incremention (+1) or decremention (-1). The register pair R1 and R0 can be used as the data memory or data memory pointer when the data memory transfer instruction is executed.

Mode	Program Counter									
	PC9	**PC8**	**PC7**	**PC6**	**PC5**	**PC4**	**PC3**	**PC2**	**PC1**	**PC0**
Initial state	0	0	0	0	0	0	0	0	0	0
Jump, Jump carry, Jump zero	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Jump bit	*	*	P7	P6	P5	P4	P3	P2	P1	P0

Program Counter

Note: PC9~PC0: Bits of instruction codes
P9~P0: Program location defined by instructions
*: Current page number

Data Memory - RAM

The static data memory (RAM) is organized by 32×4 bits to store data. The data memory can be directly accessed by "MOV A,[XXH]"and "MOV [XXH],A" or be indirectly addressed through the working register pair R1 and R0.

Each bit of data memory can be set or reset by instructions, which is helpful for data manipulation.

The data memory can be affected by binary addition, logical operation, increment and decrement operation, data memory movement and bit manipulation. The relationship between the data pointer and RAM locations is shown below:

Data Pointer		RAM Location
R1	R0	
0	0000	00H
0	0001	01H
:	:	:
:	:	:
:	:	:
0	1111	0FH
1	0000	10H
1	0001	11H
:	:	:
:	:	:
:	:	:
1	1111	1FH

Accumulator - ACC

The accumulator is the most inportant data register in data operation and control. It is one of the sources of input to the ALU and the destination of the result of operations performed in the ALU. Data transferring between I/O ports and memory may also pass through accumulator.

Arithmetic and Logic Unit - ALU

This is a circuit which performs arithmetic and logical operation. The ALU provides the following functions:

- Add with or without carry flag
- AND, OR, Exclusive OR, Complement, Rotate
- Increment, Decrement
- Data transfer
- Brancd decision

The ALU not only output the results of data operation but also sets the status of carry flag(C) in some instructions.

Initial Reset

After the power is turned on, the system goes into the initial state. The initial state performs the following functions:

- Set the program counter PC to 000H
- Set PA0 to low
- Set port PD, PE and PI2 to Tri-state

Note: The initial reset happens not only after power is turned on but also system is waked up.

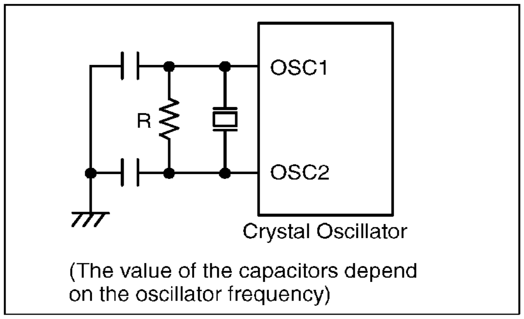
Oscillator Circuit

The HT6220 system clock oscillation circuit is a crystal oscillator connected with a crystal or ceramic resonator across OSC1 and OSC2. It provides the feedback and phase shift required for the oscillation. The relationship of the system frequency and oscillator frequency is showed below:

$f_{SYS}=f_{OSC}$

If the system operates in 455KHz, the carrier frequency can be 38KHz or 57 KHz depending on the mask option.

The machine cycle of the HT6220 consists of a



Oscillator configurations

sequence of 4 states, namley T1~T4. Each state lasts for an oscillator period. The system oscillator frequency 455KHz, and the instruction cycle is 8.8μS.

Prescaler

The HT 6220 includes a prescaler to generate carrier signal and an enable signal which is to switch on the one shoot circuit. The first stage of the prescaler is determined by the carrier frequency. If the carrier frequency is 38KHz, ÷3 is selected. However, if the carrier frequency is 57KHz, ÷2 is chosen.

Because the crystal oscillator has a unstable time in beginning, it can get to stable state after delaying 11 stages frequency divider. When the enable signal is produced, the one shoot circuit will be active. The oscillator frequency (f_{OSC}) will pass through the one shoot circuit, the system frequency (f_{SYS}) is got.

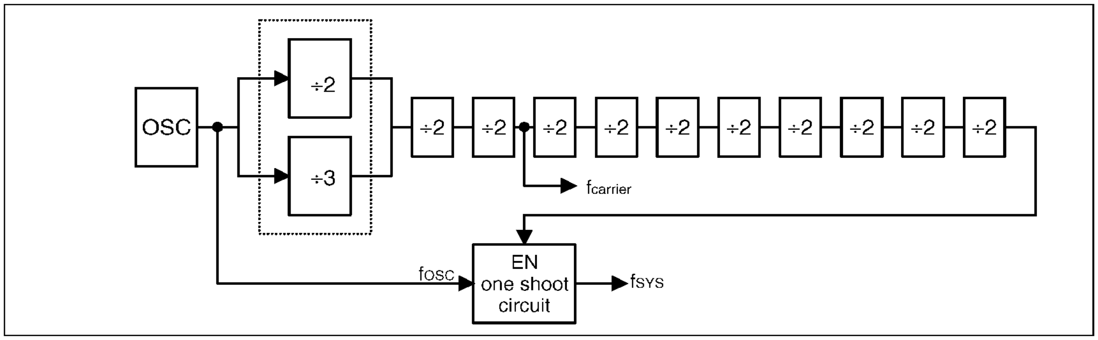
Output Line - PA0

The output line of the HT6220 is configured below:

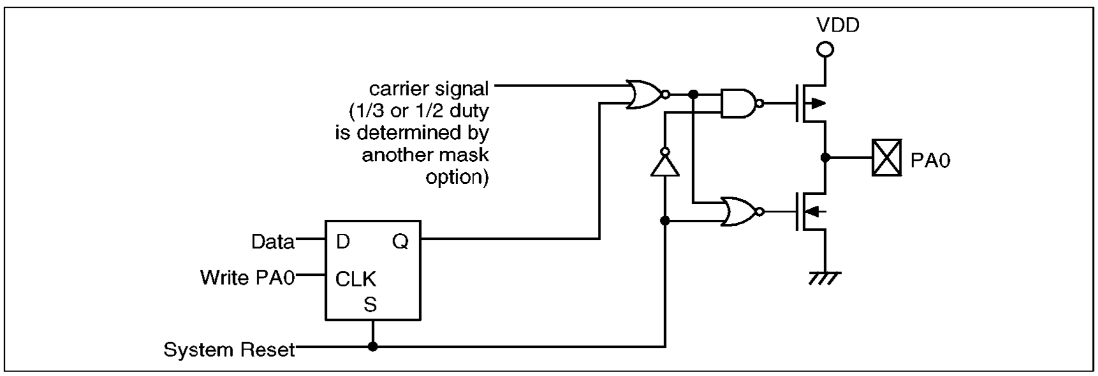
PA0 is the bit 0 of port PA. It can be configured as a CMOS output with or without carrier driving capacity.

When the system oscillator operates in 455KHz, it is easy to interface with an infrared diode. The carrier frequency can be 38KHz or 57KHz depending on the mask option. In addition, 1/3 or 1/2 duty cycle can be optioned. Writing "0" to the PA0 latch (OUT PA,A where bit 0 of ACC=0) results in a carrier output. On the other hand, writing "1" to the PA0 latch (OUT PA,A where bit 0 of ACC=1) keeps the state of PA0 at normal low level.

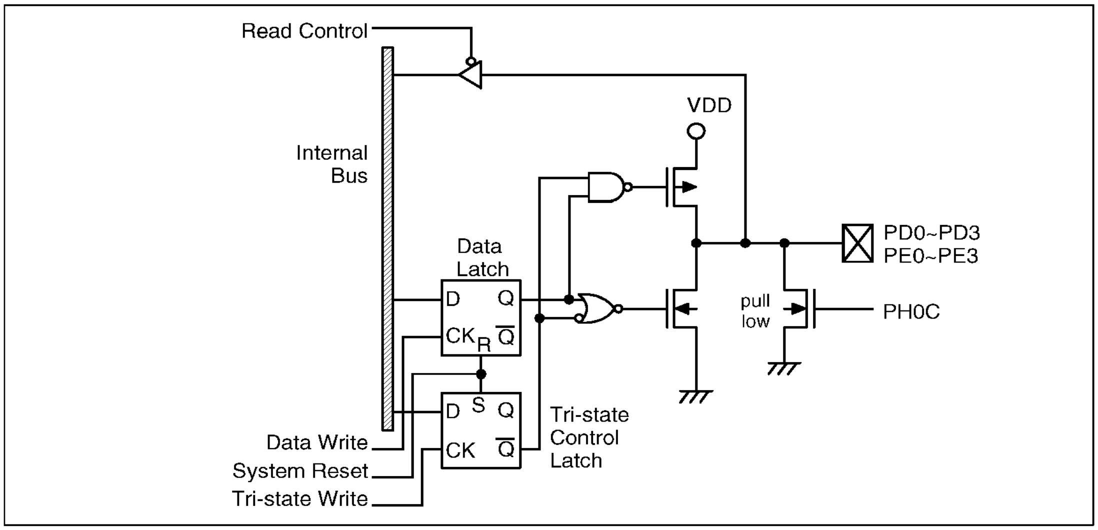
During system reset, the PA0 will be floating.



Prescaler



PA0 Output Line



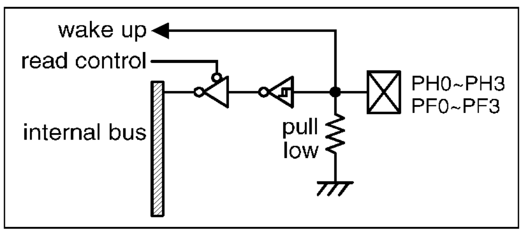
Input/Output Ports - PD, PE

However, the PA0 will stay at a low level when the carrier output is finished.

Input/Output Ports - PD, PE

The physical I/O ports (PD and PE) are with the same configuration and structure: the output is in the CMOS mode and the pull-low resistor is controlled by internal output control line PH0C of the port PH. When PH0C is 1 (OUT PH,A where bit 0 of ACC=1), the pull-low resistor exists. Otherwise, PH0C=0 (OUT PH,A where bit 0 of ACC=0), the pull-low resistor doesn't exist.

With the above basic configuration, if a line performs an input function, a "TRI" instruction can be invoked (writing a "1" to the corresponding Tri-state control latch), which makes the output part of the input/output port exhibit a floating state (in the case of no pull-low resistor) to minimize the loading effect.



Input Ports - PH, PF

After the power on is reset, the tri-state control latch will be"1", which implies all the I/O lines are floating. The PH0C will be clear to zero and the pull-low resistor doesn't exist.

Input Ports - PH, PF

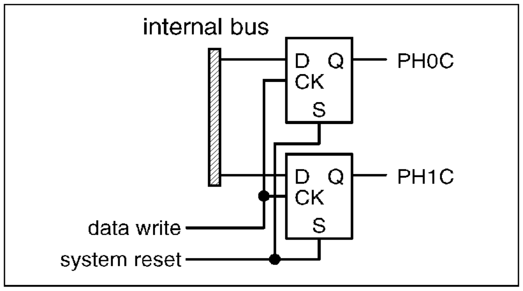
The input ports PH and PF are configured in the following way:

Every bit of the ports PH and PF has 2 configuration: pull-low resistor and wake-up capacity. A low-to-high edge trigger input will force the HT6220 to leave the HALT mode and reset the system. That pull-low resistor is physical and cannot be controlled by any software or selected by option.

Internal output control line - PH0C, PH1C

The PH0C and PH1C are designed specially for the internal output control line of port PH. The PH0C is used to control the pull-low resistor of the ports PD and PE. However, the PH1C is used to control the pull-high or pull-low resistor of bit PI1 of the port PI.

When the instruction "OUT PH, A (where bit 0 of ACC=1)" is executed, the pull-low resistor of ports PD and PE exists. However, when the instruction OUT PH, A (where bit 1 of ACC=1) is executed, the pull-low resistor of bit PI2 of port PI is selected. Finally, when A is equal to 0 (where bit 1 of ACC=0), the pull-high resistor of bit PI2 of port PI is selected.



Internal Output Control Lines - PH0C, PH1C

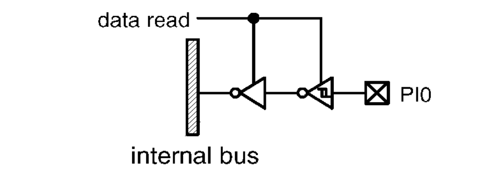
Internal Port - PG

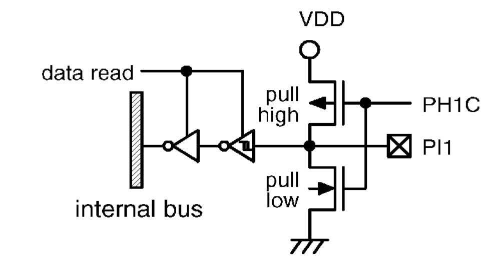
The PG is a special internal I/O port of 4-bit wide. This port can be read/written by "IN A,PG" and "OUT PG,A". Each bit of the port can be set or reset by a software instruction.

Port PI

The port PI includes 3 pins, namely PI0~PI2. Among these pins, PI0~PI1 are input lines and PI2 is the output line. The 3 pins have different configurations and structures.

The configuration of PI0, PI1

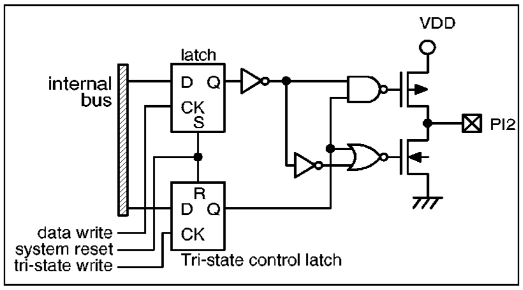




The pin PI0 is a normal input line.

The special input line PI1 has a selection of pull-low or pull-high resistor controlled by the software which is from the internal output control line PH1C of port PH. When the instruction OUT PH, A (where bit 1 of ACC=1) is executed, the pull-low resistor is selected. On the other hand, when the instruction OUT PH, A (where bit 1 of ACC=0) is executed, the pull-high resistor is selected.

The configuration of PI2



The output line PI2, a CMOS mode output, includes a Tri-state control. When the instruction "TRI PI, A" is executed, the output port exhibits a floating effect.

Halt

When the instruction "HALT" is executed, the system clock is stopped and the system is driven into a low power consumption state. At this time, the contents of the on-chip RAM and registers remain unchanged. The halt state can be terminated by inputting a low-to-high edge trigger to ports PH and PF. Then, the system is waken up. Notice that the system will be executed the hardware reset when the system is waken up.

Mask Option

The HT6220 only owns a mask option, it is shown below:

The carrier frequency of PA0 can be 38KHz, 1/3 duty or 57KHz, 1/2 duty.

INSTRUCTION SET

Instruction Set Summary

Mnemonic	Description	Word	Cycle	CF	ZF
Arithmetic					
ADD A,M	Add data memory to ACC	1	1	√	√
ADD M,A	Add ACC to data memory	1	1	√	√
ADC A,M	Add data memory with carry to ACC	1	1	√	√
ADC M,A	Add ACC with carry to data memory	1	1	√	√
ADD A,XH	Add immediate data to ACC	1	1	√	√
ANC A,XH	Add immediate data to ACC with CF not affected	2	2	—	√
CPL A	Complement ACC	1	1	—	√
CPL R1	Complement R1	1	1	—	√
Logic operation					
AND A,M	AND data memory to ACC	1	1	—	√
AND M,A	AND ACC to data memory	1	1	—	√
AND A,XH	AND immediate data to ACC	2	2	—	√
OR A,M	OR data memory to ACC	1	1	—	√
OR M,A	OR ACC to data memory	1	1	—	√
OR A,XH	OR immediate data to ACC	2	2	—	√
XOR A,M	Exclusive-OR data memory to ACC	1	1	—	√
XOR M,A	Exclusive-OR ACC to data memory	1	1	—	√
XOR A,XH	Exclusive-OR immediate data to ACC	2	2	—	√
Increment & Decrement					
INC R0	Increment register R0	1	1	—	√
INC M	Increment data memory	1	1	√	√
DEC R0	Decrement register R0	1	1	—	√
DEC M	Decrement data memory	1	1	√	√
Rotate					
RLC A	Rotate ACC left through the carry	1	1	√	—
RRC A	Rotate ACC right through the carry	1	1	√	—
Input & Output					
IN A,Pi	Input port-i to ACC, port-i=PD,PE,PF,PG, PH, PI	1	1	—	√
OUT Po,A	Output ACC to port-o, port-o=PD,PE,PG, PH,PI	1	1	—	—
TRI Pn,A	Output ACC to tri-state latch of port-n, port-n =PD,PE,PI	1	1	—	—
OUT PA,A	Output ACC0 to port A	1	1	—	—

Mnemonic	Description	Word	Cycle	CF	ZF
Data Move					
MOV A,R0	Move R0 to ACC	1	1	—	√
MOV R0,A	Move ACC to R0	1	1	—	—
MOV A,R1	Move R1 to ACC	1	1	—	√
MOV R1,A	Move ACC to R1	1	1	—	—
MOV A,M	Move data memory to ACC	1	1	—	√
MOV M,A	Move ACC to data memory	1	1	—	—
MOV A,XH	Move immediate data to ACC	1	1	—	—
MOV R1R0,XXH	Move immediate data to R1 and R0	1	1	—	—
MOV R0,M	Move data memory to R0	1	1	—	—
MOV A,[XXH]	Move data memory to ACC directly	1	1	—	√
MOV [XXH],A	Move ACC to data memory directly	1	1	—	—
Branch					
JMP addr	Jump unconditional	2	2	—	—
JC addr	Jump on carry=1	2	2	—	—
JNC addr	Jump on carry=0	2	2	—	—
JZ addr	Jump on zero flag=1	2	2	—	—
JB A.i,addr	Jump on A.i=1	2	2	—	—
JB Pm.i,addr	Jump on Pm.i=1, Pm=PE,PG	2	2	—	—
JB M.i,addr	Jump on M[R1,R0].i=1	2	2	—	—
JNZ addr	Jump on zero flag=0	2	2	—	—
JNB A.i,addr	Jump on A.i=0	2	2	—	—
JNB M.i,addr	Jump on M[R1,R0].i=0	2	2	—	—
Miscellaneous					
HALT	Enter power down mode	1	2	—	—
NOP	No operation	1	1	—	—
Flag					
CLR C	Clear carry flag	1	1	0	—
SET C	Set carry flag	1	1	1	—
Table Read					
READn MA	Rread page 0~3 of ROM code to M[R1,R0] & ACC	1	2	—	√
Bit Set/Reset					
SET M.i	Set bit of data memory	1	1	—	—
CLR M.i	Clear bit of data memory	1	1	—	—
SET Pn.i	Set bit of Pn.i, Pn=PD,PE,PG	1	1	—	—
CLR Pn.i	Clear bit of Pn.i, Pn=PD,PE,PG	1	1	—	—

Instruction Definition

ADC A,M Add data memory content and carry to accumulator

Machine code 0 0 1 0 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0", the carry flag and the accumulator are added simultaneously. The result is stored in the accumulator. The carry and zero flag are affected.

Operation ACC ← ACC+M(R1,R0)+C

ADC M,A Add accumulator and carry to data memory

Machine code 0 0 0 0 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0", the carry flag and the accumulator are added simultaneousely. The result is stored in the data memory. The carry and zero flag are affected.

Operation M(R1,R0) ← ACC+M(R1,R0)+C

ADD A,M Add data memory to accumulator

Machine code 0 0 1 1 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0" and the accumulator are added. The result is stored in the accumulator. The carry and zero flag are affected.

Operation ACC ← ACC+M(R1,R0)

ADD A,XH Add immediate data to accumulator

Machine code 0 1 1 1 d d d d

Description The immediate data and the accumulator content are added. The result is stored in the accumulator. The carry and zero flag are affected.

Operation ACC ← ACC+XH

ADD M,A Add accumulator to data memory

Machine code 0 0 0 1 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0", and the accumulator content are added. The result is stored in the data memory. The carry and zero flag are affected.

Operation M(R1,R0) ← ACC+M(R1,R0)

ANC A,XH Add immediate data to ACC with CF not affected

Machine code 0 0 0 0 d d d d 0 0 0 0 0 0 1 1

Description The immediate data and the accumulator content are added. The result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC+XH

AND A,M Logical AND data memory to accumulator

Machine code 0 0 0 1 0 0 1 1

Description Data in the accumulator and the data memory addressed by the register pair "R1,R0" performs the bitwise logical-AND operation and the result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "AND" M(R1,R0)

AND A,XH Logical AND accumulator with immediate data

Machine code 0 0 0 0 0 0 1 1 0 0 1 0 d d d d

Description Data in the accumulator and the specified data perform the bitwise logical-AND operation and the result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "AND" XH

AND M,A Logical AND accumulator to data memory

Machine code 1 1 0 0 0 0 0 0

Description Data in the accumulator and the data memory addressed by the register pair "R1,R0" performs the bitwise logical-AND operation and the result is stored in the data memory. The zero flag is affected.

Operation M(R1,R0) ← ACC "AND" M(R1,R0)

CLR C Clear carry flag

Machine code 1 1 1 0 0 0 1 0

Description The carry is reset to zero.

Operation C ← 0

CLR M.i	Clear bit of data memory
Machine code	0 0 1 1 i3 i2 i1 i0
	i0~i3 are determined by operand i. The corresponding bit will be "0" if i is reset.Otherwise the bit is set to 1. For example, if i=0 then i3~i0=1110.
Description	The specified bit of data memory addressed by register pair "R1,R0" is reset to zero.
Operation	M(R1,R0).i ← 0

CLR Pn.i	Clear bit of port
Machine code	PD 0 0 1 0 i3 i2 i1 i0
	PE 0 0 0 0 i3 i2 i1 i0
	PG 0 0 0 1 i3 i2 i1 i0
	i0~i3 are determined by operand i. The corresponding bit will be "0" if i is reset. Otherwise the bit is set to 1. For example, if i=0 then i3~i0=1110.
Description	The specified bit of port "Pn" is reset to zero. Pn can be PD,PE,PG
Operation	Pn.i ← 0; Pn=PD,PE,PG

CPL A	Complement accumulator
Machine code	0 0 1 1 1 1 1 1
Description	Each bit of the accumulator is logically complemented. The zero flag is affected.
Operation	ACC ← $\overline{\text{ACC}}$

CPL R1	Complement R1
Machine code	1 1 1 1 0 0 0 1
Description	Each bit of the register R1 is logically complemented. The zero flag is affected.
Operation	R1 ← $\overline{\text{R1}}$

DEC M Decrement data memory

Machine code 0 0 0 0 1 1 1 1

Description Data in the data memory specified by the register pair "R1,R0" is decremented by one. The carry flag and zero flag are affected. Carry is set if a borrow does not take place in DEC[M] operation; otherwise carry is cleared.

Operation M(R1,R0) ← M(R1,R0)–1

DEC R0 Decrement register R0

Machine code 0 0 0 1 1 1 1 1

Description Data in the working register R0 is decremented by one. Only the zero flag is affected.

Operation R0 ← R0–1

HALT Enter halt state

Machine code 0 0 1 1 0 1 0 1

Description HALT stops instruction execution and places the controller in power down mode. Reset or a active signal in the "PE,PF,PH" ports (by mask option) will resume execution. No flags are affected.

IN A,Pi Input port to accumulator

Machine code

PD	0 0 1 0 1 0 0 1
PE	0 0 0 0 1 0 0 1
PF	0 0 1 0 1 1 0 0
PG	0 0 0 1 1 0 0 1
PH	0 0 1 0 0 1 0 1
PI	1 0 0 1 0 1 0 0 1

Description The data on port "Pi" is transferred to the accumulator. The zero flag is affected.

Operation ACC ← Pi; Pi=PD,PE,PF,PG,PH,PI

INC M — Increment data memory

Machine code: 1 1 0 0 0 0 0 1

Description: Data in the data memory specified by the register pair “R1,R0” is incremented by one. The carry and zero flag are affected. Carry is set if the operation results in a carry out; otherwise carry is cleared.

Operation: M(R1,R0) ← M(R1,R0)+1

INC R0 — Increment register R0

Machine code: 1 1 0 1 0 0 0 1

Description: Data in the working register “R0” is incremented by one. The zero flag is affected.

Operation: R0 ← R0+1

JB A.i,addr — Jump if bit of accumulator is set

Machine code: 1 1 0 0 0 1 i1 i0 a a a a a a a a

i0,i1 indicate which bit of accumulator will be detected. For example, i0=i1=0 means that if bit 0 of accumulator=1, the jump will execute.

Description: If the indicated bit of accumulator is set to 1, control passes to specified address; otherwise proceed with the next instruction. Note that the branch destination is only available in the same page. (only bits 0~7 of the program counter will be replaced by the destination address.)

Operation:
PC ← address, if bit i of ACC=1
PC ← PC+2, if bit i of ACC=0

JB Pm.i,addr — Jump if bit of I/O port is set

Machine code:
PE 1 1 0 0 1 1 i1 i0 a a a a a a a a
PG 1 1 0 1 1 1 i1 i0 a a a a a a a a

i0,i1 indicate which bit of port PE,PG will be detected. For example, i0=i1=0 means that if bit 0 of port Pm=1, the jump will execute.

Description: If the indicated bit of port PE, PG is set to 1, control passes to specified address; otherwise proceed with the next instruction. Note that the branch destination is only available in the same page. (Only bits 0~7 of the program counter will be replaced by the destination address.)

Operation:
PC ← address, if bit i of Pm=1, Pm=PE,PG
PC ← PC+2, if bit i of Pm=0, Pm=PE,PG

JB M.i,addr Jump if bit of data memory is set

Machine code: 1 1 1 0 0 1 i1 i0   a a a a a a a a

i0,i1 indicate which bit of data memory will be detected. For example, i0=i1=0 means that if bit 0 of data memory is equal to 1, the jump will execute.

Description: If the indicated bit of data memory addressed by register pair "R1,R0" is set to 1, control passes to specified address; otherwise proceed with the next instruction. Note that the branch destination is only available in the same page. (Only bits 0~7 of program counter is replaced by the destination address.)

Operation: PC ← address, if bit i of M(R1,R0)=1
PC ← PC+2, if bit i of M(R1,R0)=0

JC addr Jump if carry flag is set

Machine code: 1 1 1 0 1 0 a a   a a a a a a a a

Description: If the carry flag is set to one, control passes to the specified address; otherwise proceed with the next instruction.

Operation: PC ← address, if C=1
PC ← PC+2, C=0

JMP addr Direct jump

Machine code: 1 1 1 1 1 1 a a   a a a a a a a a

Description: All bits of the program counter are replaced with the directly specified address, and control passes to the destination.

Operation: PC ← address

JNB A.i,addr Jump if bit of accumulator is not set

Machine code: 1 1 0 1 0 1 i1 i0   a a a a a a a a

i0,i1 indicate which bit of accumulator will be detected. For example, i0=i1=0 means that if bit 0 of accumulator =0, the jump will execute.

Description: If the indicated bit of accumulator is reset to 0, control passes to specified address; otherwise proceed with the next instruction. Note that the branch destination is only available in the same page. (Only bits 0~7 of program counter is replaced by the destination address.)

Operation: PC ← address, if bit i of ACC=0
PC ← PC+2, if bit i of ACC=1

JNB M.i,addr Jump if bit of data memory is not set

Machine code 1 1 1 1 0 1 i1 i0 a a a a a a a a

i0,i1 indicate which bit of data memory will be detected. For example, i0=i1=0 means that if bit 0 of data memory =0, the jump will execute.

Description If the indicated bit of data memory addressed by register pair "R1,R0" is reset to 0, control passes to specified address; otherwise proceed with the next instruction. Note that the branch destination is only available in the same page. (Only bits 0~7 of program counter is replaced by the destination address.)

Operation PC ← address, if bit i of M(R1,R0)=0
PC ← PC+2, if bit i of M(R1,R0)=1

JNC addr Jump if carry flag is not set

Machine code 1 1 1 1 1 0 a a a a a a a a a a

Description If the carry flag is reset to zero, control passes to the specified address; otherwise proceed with the next instruction.

Operation PC ← address, if C=0
PC ← PC+2, if C=1

JNZ addr Jump if zero flag is not set

Machine code 1 1 0 1 1 0 a a a a a a a a a a

Description If the zero flag is reset to zero, control passes to the specified address; otherwise proceed with the next instruction.

Operation PC ← address, if Z=0
PC ← PC+2, if Z=1

JZ addr Jump if zero flag is set

Machine code 1 1 0 0 1 0 a a a a a a a a a a

Description If the zero flag is set to one, control passes to the specified address; otherwise proceed with the next instruction.

Operation PC ← address, if Z=1
PC ← PC+2, if Z=0

MOV A,M Move data memory to accumulator

Machine code	0 0 1 1 1 0 0 1
Description	The content of the data memory addressed by the register pair "R1,R0" is moved to the accumulator. If the contents of data memory is zero, the zero flag will be set.
Operation	ACC ← M(R1,R0)

MOV A,R0 Move register R0 content to accumulator

Machine code	0 0 0 0 0 1 0 1
Description	The content of register R0 is moved into the accumulator. If the content of register R0 is zero, the zero flag will be set.
Operation	ACC ← R0

MOV A,R1 Move register R1 content to accumulator

Machine code	0 0 0 0 0 1 1 0
Description	The content of register R1 is moved into the bit 0 and 1 of the accumulator. The bit 1 and 3 of the accumulator are reset to 0. If the content of register R1 is zero, the zero flag will be set.
Operation	ACC ← R1

MOV A,XH Move immediate data to accumulator

Machine code	0 1 1 0 d d d d
Description	The 4-bit data specified by code is loaded in the accumulator. No flags are affected.
Operation	ACC ← XH

MOV A,[XXH] Move data memory to accumulator directly

Machine code	1 0 0 m4 m3 m2 m1 m0 m4~m0: address of data memory
Description	The content of the data memory directly addressed by code is moved to the accumulator. The zero flag is affected
Operation	ACC ← M(m4~m0)

MOV M,A — Move accumulator to data memory

Machine code: 0 0 1 1 1 0 1 0

Description: The content of the accumulator is moved to the data memory addressed by register pair “R1,R0”.

Operation: M(R1,R0) ← ACC

MOV R0,A — Move accumulator to the register R0

Machine code: 0 0 0 1 0 1 0 1

Description: The content of accumulator is moved into the register “R0”.

Operation: R0 ← ACC

MOV R1,A — Move accumulator to the register R1

Machine code: 0 0 0 1 0 1 1 0

Description: The bit 0 of the accumulator is moved into the register “R1”.

Operation: R1 ← ACC

MOV R0,M — Move data memory to register R0

Machine code: 0 0 1 0 0 1 1 0

Description: The content of the data memory addressed by the register pair “R1,R0” is moved to the register “R0”.

Operation: R0 ← M(R1,R0)

MOV R1R0,XXH — Move immediate data to register R0 and R1

Machine code: 0 1 0 d d d d d

Description: The 5-bit data specified by code is loaded in the register pair “R1,R0”. No flags are affected.

Operation: R1,R0 ← XH

MOV [XXH],A Move accumulator to data memory directly

Machine code 1 0 1 m4 m3 m2 m1 m0

m4~m0:address of data memory

Description The content of accumulator is directly moved to the data memory addressed by code. No flags are affected.

Operation M (m4~m0) ← ACC

NOP No operation

Machine code 1 1 1 1 0 0 0 0

Description No operation is performed. Execution continues with the next instruction.

Operation (PC) ← (PC)+1

OR A,M Logical OR data memory to accumulator

Machine code 0 0 1 0 0 0 1 1

Description Data in the accumulator is logically ORed with the data memory addressed by register pair "R1,R0". The result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "OR" M(R1,R0)

OR A,XH Logical OR accumulator with immediate data

Machine code 0 0 0 0 0 0 1 1 0 0 0 1 d d d d

Description Data in the accumulator is logically ORed with the immediate data. The result is stored in the accumulator and the zero flag is affected.

Operation ACC ← ACC "OR" XH

OR M,A Logical OR accumulator to data memory

Machine code 1 1 0 1 0 0 0 0

Description Data in the accumulator is logically ORed with the data memory addressed by register pair "R1,R0". The result is stored in the data memory. The zero flag is affected.

Operation M(R1,R0) ← ACC "OR" M(R1,R0)

OUT PA,A — Output accumulator data to port A

Machine code: 0 0 1 1 1 1 0 0

Description: The bit 0 on the accumulator is transferred to the output port PA0 for no carrier option.

If carrier output option is selected, writting “0” to the PA0 will results in a carrier output, and writting “1” to the PA0 will keeps the state of PA0 at normal low level.

Operation: PA0 ← ACC0 (no carrier option).
ACC=0, PA0: carrier (carrier output option).
ACC=1, PA0: 0 (carrier output option).

OUT Po,A — Output accumulator data to port

Machine code:

PD 0 0 1 0 1 0 1 0

PE 0 0 0 0 1 0 1 0

PG 0 0 0 1 1 0 1 0

PH 0 0 1 1 0 1 1 0

PI 1 0 0 1 0 1 0 1 0

Description: The data on the accumulator is transferred to the “Po” port.

Operation: Po ← ACC; Po=PD,PE,PG,PH,PI

READn MA — Read ROM code to data memory and accumulator

Machine code: 1 1 1 0 1 1 n n

nn: page number (0~3)

Description: The 8 bits of ROM code addressed by ACC and M(R1,R0) in page n are moved to the data memory addressed by register pair “R1,R0” and the accumulator. The high nibble of the ROM code is loaded to the data memory and the low nibble of the ROM code is loaded to the accumulator. The ACC is zero, the zero flag will be set. The address of the ROM code are specified by the following description:

ROM code address bit 9~8 ← Page “nn”
ROM code address bit 7~4 ← ACC
ROM code address bit 3~0 ← M(R1,R0)

ROM code address bit 10~8 ← Page “nnn”
ROM code address bit 7~4 ← ACC
ROM code address bit 3~0 ← M(R1,R0)

Operation: M(R1,R0) ← ROM code (high nibble)
ACC ← ROM code (low nibble)

RLC A Rotate accumulator left through carry

Machine Code: 1 1 1 1 0 0 1 1

Description: The contents of the accumulator are rotated left one bit. Bit 3 replaces the carry bit; the carry bit is rotated into the bit 0 position.

Operation: An+1 ← An; An: Accumulator bit n (n=0,1,2)
A0 ← CF
CF ← A3

RRC A Rotate accumulator right through carry

Machine Code: 1 1 1 1 0 0 1 0

Description: The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 3 position.

Operation: An ← An+1; An: Accumulator bit n (n=0,1,2)
A3 ← CF
CF ← A0

SET C Set carry flag

Machine code: 1 1 1 0 0 0 1 1

Description: The carry flag is set to one.

Operation: C ← 1

SET M.i Set bit of data memory

Machine code: 0 0 1 1 i3 i2 i1 i0

i0~i3 are determined by operand i. The corresponding bit will be "1" if the bit i of the memory is set to 1. For example, if i=0 then i3~i0=0001.

Description: The bit of memory addressed by "R1,R0" is set to one.

Operation: M(R1,R0).i ← 1

SET Pn.i

Set bit of I/O port

Machine code

PD 0 0 1 0 i3 i2 i1 i0

PE 0 0 0 0 i3 i2 i1 i0

PG 0 0 0 1 i3 i2 i1 i0

i0~i3 are determined by operand i. The correponding bit will be “1” if i is setting. For example, if i=0 then i3~i0=0001

Description

The specified bit i of port “Pn” is set to one. Pn=PD,PE,PG.

Operation

Pn.i ← 1; Pn=PD,PE,PG

TRI Po,A

Output accumulator to tri-state latch

Machine code

PD 0 0 0 0 1 1 0 0

PE 0 0 0 1 1 1 0 0

PI 1 0 0 1 0 1 0 1 0

Description

Data in the accumulator is transferred to the tri–state latch of port “Po”. The “1” witting to the tri-state latch makes the corresponding output part becomes floating.

Operation

Po ← ACC ; Po=PD,PE,PI

XOR A,M

Logical Exclusive-OR data memory to accumulator

Machine code

0 0 1 1 0 0 1 1

Description

Data in the accumulator is Exclusive-ORed with the data memory addressed by register pair “R1,R0”. The result is stored in the accumulator. The zero flag is affected.

Operation

ACC ← ACC “XOR” M(R1,R0)

XOR A,XH Logical Exclusive-OR accumulator with immediate data

Machine code 0 0 0 0 0 0 1 1   0 0 1 1 d d d d

Description Data in the accumulator is Exclusive-ORed with the immediate data specified by code. The result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "XOR" XH

XOR M,A Logical Exclusive-OR accumulator to data memory

Machine code 1 1 1 0 0 0 0 0

Description Data in the accumulator is Exclusive-ORed with the memory addressed by register pair "R1,R0". The result is stored in the data memory. The zero flag is affected.

Operation M(R1,R0) ← ACC "XOR" M(R1,R0)