

SCG9864 Microcontroller Specification

FEATURE SUMMARY

Technology

- CMOS technology.

CPU

- 8-bit CPU core.
- 126 powerful and easy-to-use instruction set.

Memory

- Internal ROM: 64K x 8
- Internal RAM: 128 x 8
- Memory is organized as 16K-byte page.

I/O structure

- Memory mapped I/O structure.

Oscillation Frequency

- RC Oscillator for the main system clock up to 4MHz, instruction cycle ~0.5us.
- Subsystem clock frequency of 32.768kHz for real time timer.

Power Down

- Stop mode (Stop oscillation)

Interrupts

- 2 timer interrupts.
- 1 real time interrupt.
- 1 external event interrupt.
- 1 software interrupt.

I/O ports

- 2 8-bit programmable I/O ports.
- 1 7-bit output port.

Timer

- 2 8-bit pre-scalar auto reload timers which can be cascaded to form one 16-bit timer.
- 1 real time timer.
- 1 watch dog timer.

Operating Voltage

- 2.4V to 5.5V.

Function Overview

Program Memory Map

RAM Area:

<i>Address</i>	<i>Description</i>
\$00-\$0F	H/W registers and I/Os
\$10-\$7F	Reserved
\$80-\$FF	Data memory and Stack

Figure 1. Memory map in RAM area

ROM Area:

<i>Physical Address</i>	<i>Description</i>
\$0000-\$3BFF	Page 0 Program Reset address at \$0000
\$3C00-\$3FFF	IC Test Program
\$4000-\$7FFF	Page 1
\$8000-\$BFFF	Page 2
\$C000-\$FFFF	Page 3 Software interrupt at \$FFC0 External interrupt at \$FFD0 Timer 1 interrupt at \$FFE0 Timer 0/RTC interrupt at \$FFF0

Figure 2. Memory map in ROM area

CPU core

An 8-bit accumulator based CPU core can directly address up to 64 x 16K byte addressing space. Most of the instructions are executed in two cycles. Instruction is generally one byte and will have an extra byte for some addressing modes.

CPU Registers:

Program Counter (PC)

The 14-bit Program Counter stores address for instruction fetch during program execution. It makes up a page size of 16K bytes. Together with the Program Page Register (iPAGE), it becomes a 20-bit address that can access up to 1,048,576 bytes. When the CPU resets, the content of the iPAGE:PC will be 00:0000. If interrupt occurs, the type of interrupt will then determine its content. PC will automatically be incremented to the next instruction after an instruction fetch.

Table 1. Different types of interrupt

Interrupt	iPAGE:PC
Timer 0 / RTC Interrupt	3H:3FF0H
Timer 1 Interrupt	3H:3FE0H
External Interrupt	3H:3FD0H
Software Interrupt	3H:3FC0H

Page Register (PAGE)

An 8-bit Page Register to change the program flow. The most significant two bits are always set to zero.

Data Bank Register (BANK)

An 8-bit Bank Register to access data memory. The most significant two bits are always set to zero.

Program Page Register (iPAGE)

An 8-bit Program Page Register is combined with PC for instruction fetch. The most significant two bits are always set to zero.

Accumulator (A)

An 8-bit Accumulator for arithmetic, logical and data movement operation.

Temporary Register (B)

An 8-bit temporarily storage for accumulator.

Index Registers (X, Y)

These two 8-bit registers can be used for general registers and index registers of the indirect addressing mode. They can also be used as pointer for table read and memory write instructions.

Stack Register (SP)

A 7-bit Stack Register (SP) stores the address for stack operation. After CPU resets, the value is \$00. The SP has to be initialized at \$FF by software. This means the stack frame starts from the highest address memory location.

Program Status (PS)

This is an 8-bit Program Status Register. However only 4-bit is used for controlling ALU operations and instruction execution sequences.

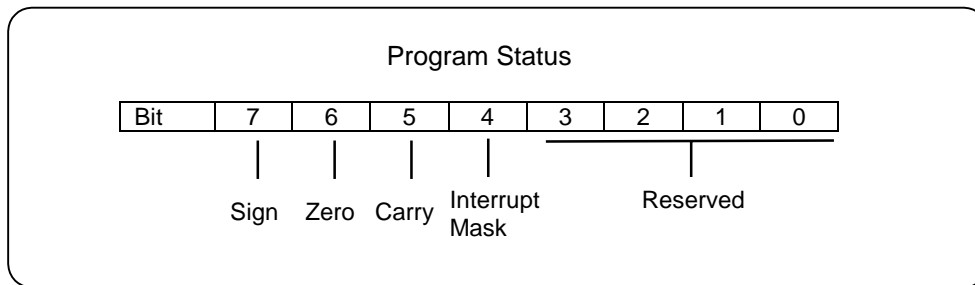


Figure 3. 8-bit Program Status

Carry Flag (C)

Whenever there is a carry or borrow occurs after an arithmetic operation, carry flag is set to 1. Otherwise, it is cleared to 0.

Besides, the “Rotate” instructions can also change the carry flag which value is a bit shifted out of the specified source operand.

Executing single instruction of “SETC” or “CLRC” can also alter this flag.

Upon returning from an interrupt service routine, this flag will be restored.

Zero Flag (Z)

For arithmetic and logical operations, Zero flag will be set to 1 if the result is zero.

Besides, for operation that involves moving source operand to accumulator, zero flag will also be set to 1 if the content of the source operand is zero.

Upon returning from an interrupt service routine, this flag will be restored.

Sign Flag (N)

Sign flag stores the most significant bit of a result after the following operations: -

- a. Arithmetic
- b. Logic
- c. Move from source operand to accumulator

This flag will also be restored upon returning from an interrupt service routine.



Interrupt Mask Flag (I)

The flag will be set to 1 when entering an interrupt service routine. By that time, all other interrupt events will be pending.

After exiting from the interrupt service routine, this interrupt mask flag will be cleared to 0. Then interrupt handling will be resumed.

Memory

There are 64K bytes internal ROM with each page size is 16K bytes. Regarding the instruction pointer, it is organized as iPAGE:PC for instruction fetch.

Change of program flow between pages is by modifying the PAGE register and then followed by executing a JMP or CALL instruction. These two instructions will load the PAGE register to iPAGE and change the content in PC for long JMP or CALL instruction. It does not need to change the PAGE register if it is a short JMP/CALL (JMP/CALL within page) because the PAGE register is normally the same as the iPAGE register. To exit a subroutine, long or short return type must be specified for long or short CALL respectively.

The microcontroller has 128 bytes internal RAM data storage of address \$80-\$FF. This area includes stack frame and data memory. The stack frame is usually initialized at the highest RAM address location, i.e. \$FF.

Oscillation Circuits

Main system and subsystem oscillation circuitry generates the internal clock signal for the CPU and other hardware timings. The main system clock uses the RC oscillation source. The operating frequency is up to 4 MHz. This clock is for the CPU and the two timers.

The subsystem clock is for the real time signals. It uses 32.768 kHz crystal. It has to be tied to a voltage level, either HIGH or LOW, if the real time timer and the watch dog timer are not used.

Power Down

The microcontroller supports power down mode for saving power.

Executing a STOP instruction will stop the main system oscillation to save most of microcontroller power.

Only an external interrupt will release the microcontroller from STOP mode.

Interrupts

The MICROCONTROLLER has 2 timer interrupts, one real time interrupt, one external event interrupt and one software interrupt. When interrupt occurs, the content of PC, iPAGE and PS are pushed onto the stack in sequence. And then, the corresponding interrupt vector is loaded into iPAGE:PC. Upon executing a RTI instruction, the registers are popped out of the stack in the reversed order.

The preference of interrupt priority is Timer 0 / RTC interrupt, Timer 1 interrupt and then external interrupt.



I/O ports

The microcontroller has one 7-bit output port and two software controllable 8-bit I/O ports.

For I/O port 0, there is a pull-low resistor when it is configured as in input mode, and this resistor is disabled when it is in output mode.

For I/O port 1, it is floating when in input mode.

For Port 2, it is a 7-bit output port only.

Timers

The microcontroller has two programmable 8-bit timers (T0 and T1) for system timing. It has also a real time timer and a watch dog timer when subsystem clock is employed. All the timers can be enabled or disabled by configuring an internal register, TCONG. At CPU resets, all of them are disabled.

T0 and T1 are up-counters and can be configured as either two 8-bit pre-scalar auto reload timer or as a 16-bit pre-scalar auto reload timer. The timer overflow flag will be set if the timer overflows. Then an interrupt will be generated if the corresponding interrupt enable bit is set to 1.

The real time timer provides 0.5 sec interrupt for RTC functions. Watch dog timer will overflow in ~1 sec and then will reset the CPU.

BLOCK DIAGRAM

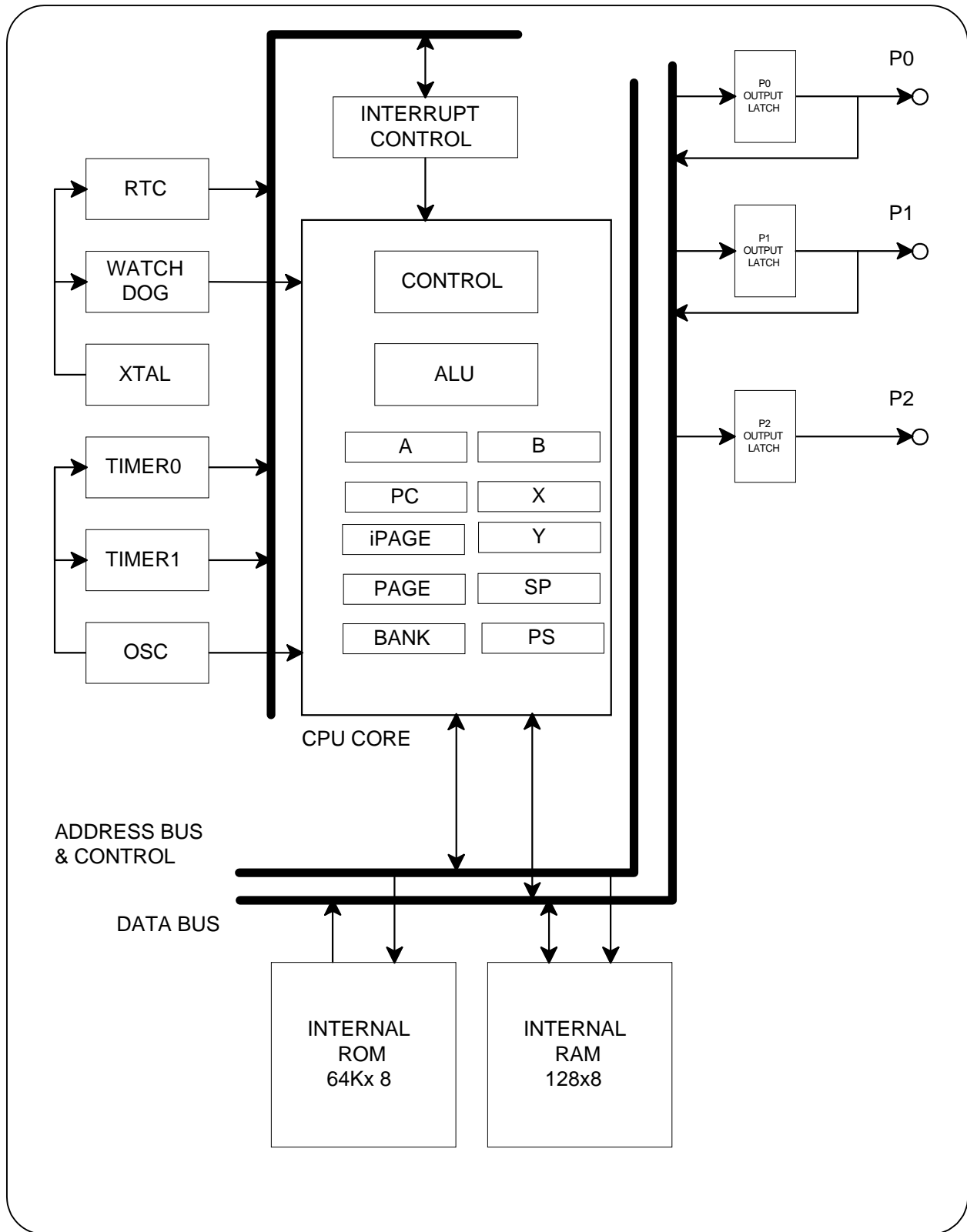


Figure 4. Block diagram of SCG9864

PIN DESCRIPTIONS

Table 2. Pin Description

Pin	Name	Type	Description	Pad Type
1	P_CLK	I/P	RC Oscillator for the Main System clock	Pad type 5
2	CLK320	O/P	32.768KHz RTC	Pad type 6
3	CLK32I	I/P	32.768KHz RTC	Pad type 6
4	GND	Power	Ground (optional)	
5	P0RW0	I/O	Port 0	Pad type 1
6	P0RW1	I/O	Port 0	Pad type 1
7	P0RW2	I/O	Port 0	Pad type 1
8	P0RW3	I/O	Port 0	Pad type 1
9	VDD	Power	Supply voltage	
10	P0RW4	I/O	Port 0	Pad type 1
11	P0RW5	I/O	Port 0	Pad type 1
12	P0RW6	I/O	Port 0	Pad type 1
13	P0RW7	I/O	Port 0	Pad type 1
14	GND	Power	Ground (optional)	
15	P1RW0	I/O	Port 0	Pad type 1
16	P1RW1	I/O	Port 1	Pad type 1
17	P1RW2	I/O	Port 1	Pad type 1
18	P1RW3	I/O	Port 1	Pad type 1
19	VDD	Power	Supply Voltage (optional)	
20	P1RW4	I/O	Port 1	Pad type 1
21	P1RW5	I/O	Port 1	Pad type 1
22	P1RW6	I/O	Port 1	Pad type 1
23	P1RW7	I/O	Port 1	Pad type 1
24	GND	Power	Ground (optional)	
25	P2W0	O/P	Port 2	Pad type 2
26	P2W1	O/P	Port 2	Pad type 2
27	P2W2	O/P	Port 2	Pad type 2
28	P2W3	O/P	Port 2	Pad type 2
29	VDD	Power	Supply Voltage (optional)	
30	P2W4	O/P	Port 2	Pad type 2
31	P2W5	O/P	Port 2	Pad type 2
32	P2W6	O/P	Port 2	Pad type 2
33	PWMP	O/P	Speaker +	Pad type 2
34	PWMM	O/P	Speaker -	Pad type 2
35	GND	Power	Ground	
36	VDD	Power	Supply Voltage	
37	TEST	I/P	Test pin, no connection	Pad type 4
38	RESET	I/P	Reset pin	Pad type 3
39	EXT_INT	I/P	External Interrupt pin	Pad type 4

Note: The Optional pins for VDD and GND can be left open as NC pins.

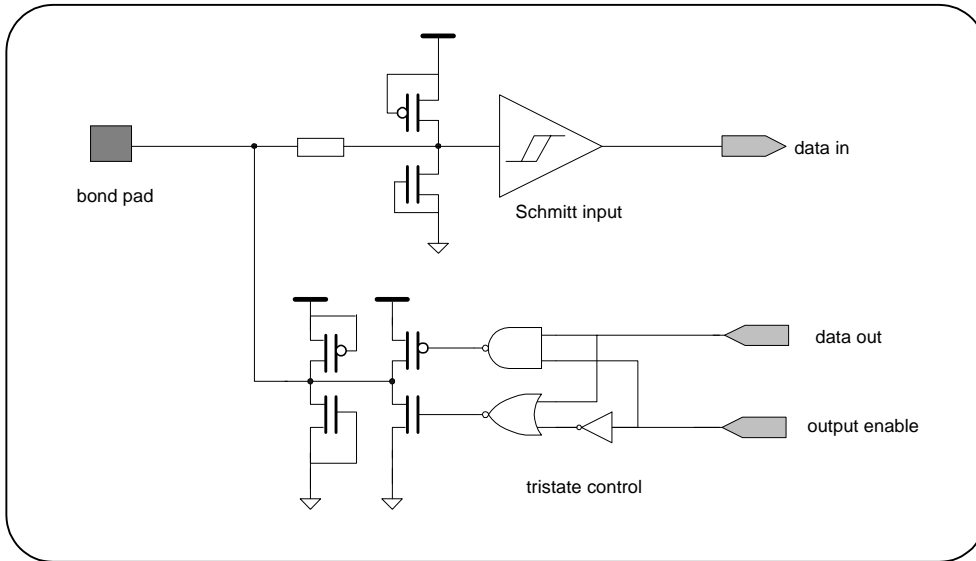


Figure 5. Pad type 1 (bi-directional pad)

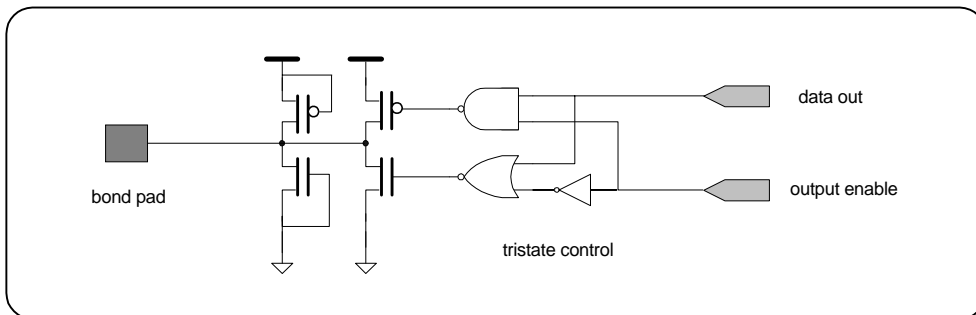


Figure 6. Pad type 2 (output pad)

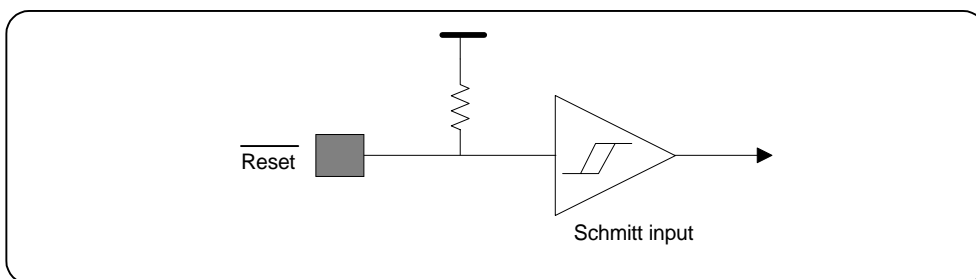


Figure 7. Pad type 3 ($\overline{\text{Reset}}$)

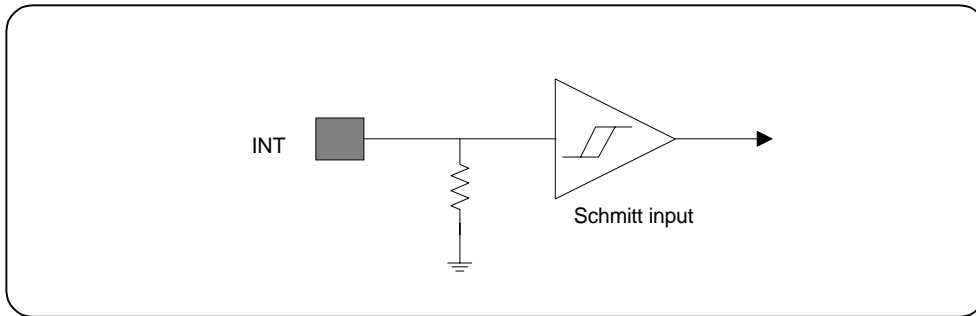


Figure 8. Pad type 4 (external interrupt)

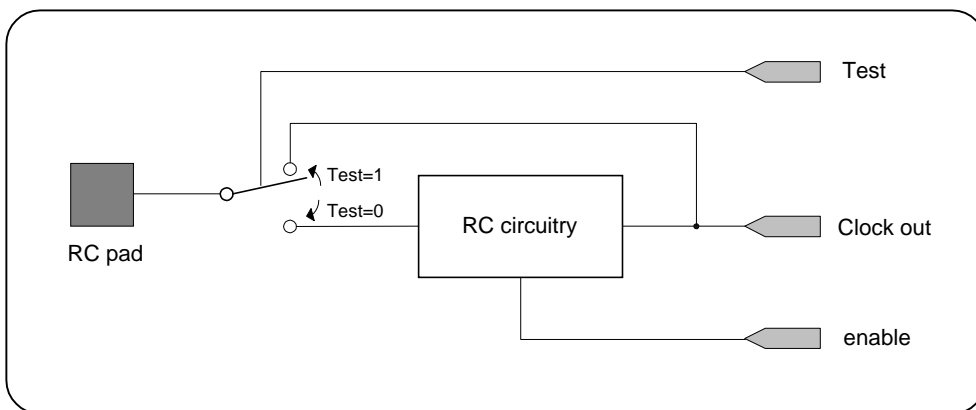


Figure 9. Pad type 5 (RC oscillation pad)

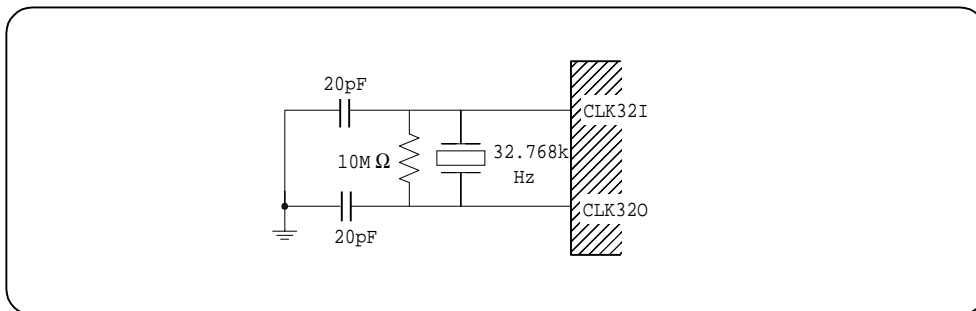


Figure 10. Pad type 6 (Clock circuit)

Control Registers

Summary

Table 3. Summary of Control Registers

Hex	Mnemonic	Control Registers Name	R/W
00H	P0DIR	Port 0 Direction Control Register	W
01H	-	(Reserved)	-
02H	P0RW	Port 0 Read/Write Port	R/W
03H	P1DIR	Port 1 Direction Control Register	W
04H	P1RW	Port 1 Read/Write Port	R/W
05H	P2W	Port 2 Output Port	W
06H	-	(Reserved)	-
07H	-	(Reserved)	-
08H	-	(Reserved)	-
09H	TCONG	Timer Configuration Port	W
0AH	INTR	Interrupt Control Register	W
0BH	TFLAG	Timer Flag Status Register/Watch-dog Reset	R/W
0CH	T0VAL	Timer 0 Preset Value Register	W
0DH	T1VAL	Timer 1 Preset Value Register	W
0EH	SPHEN	D/A Output Control Register	W
0FH	-	(Reserved)	-

Descriptions

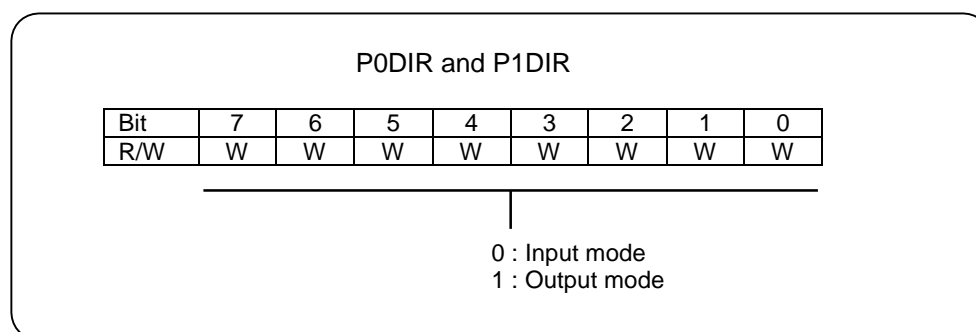


Figure 11. Port 0 and Port 1 Direction Control Registers

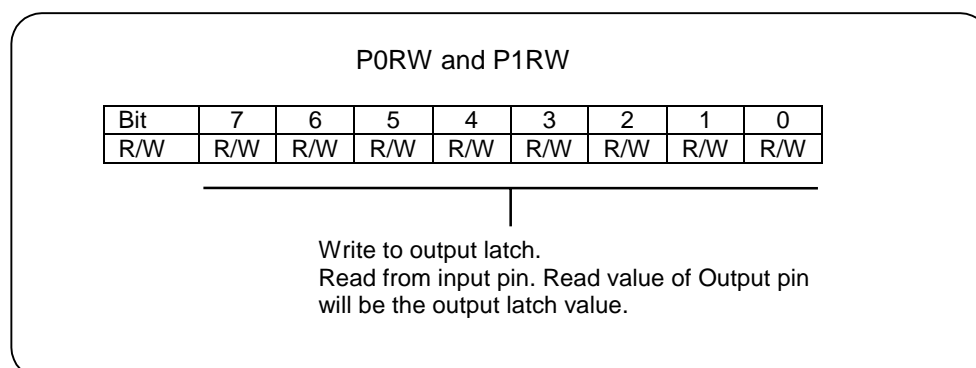


Figure 12. Port 0 and Port 1 Read/Write Port

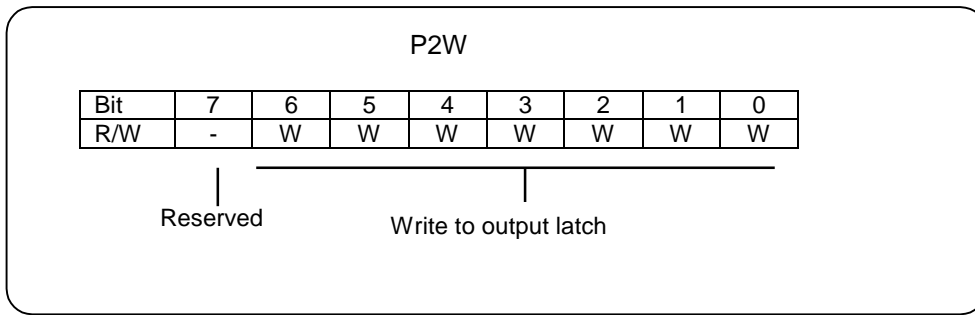


Figure 13. Port 2 Output Port

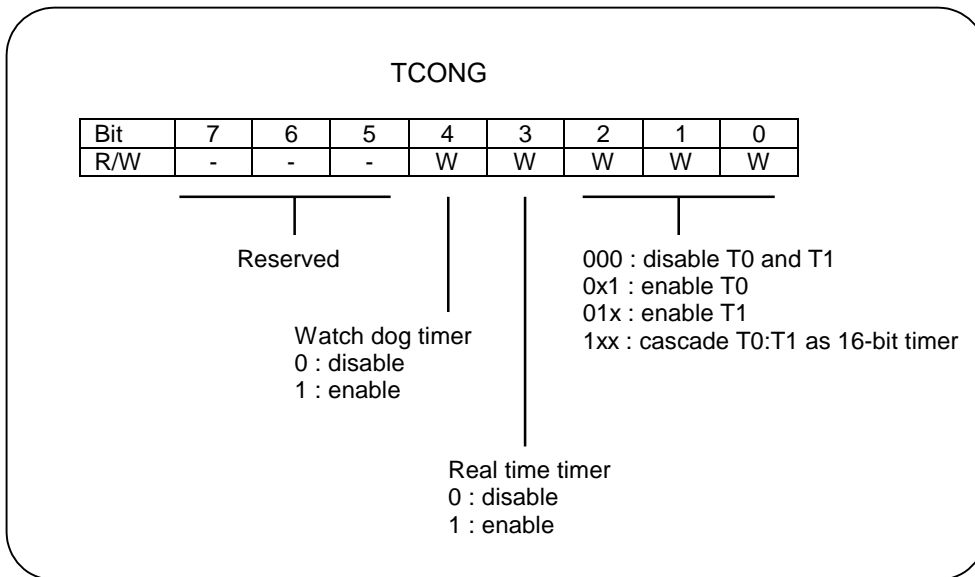


Figure 14. Timer Configuration Port

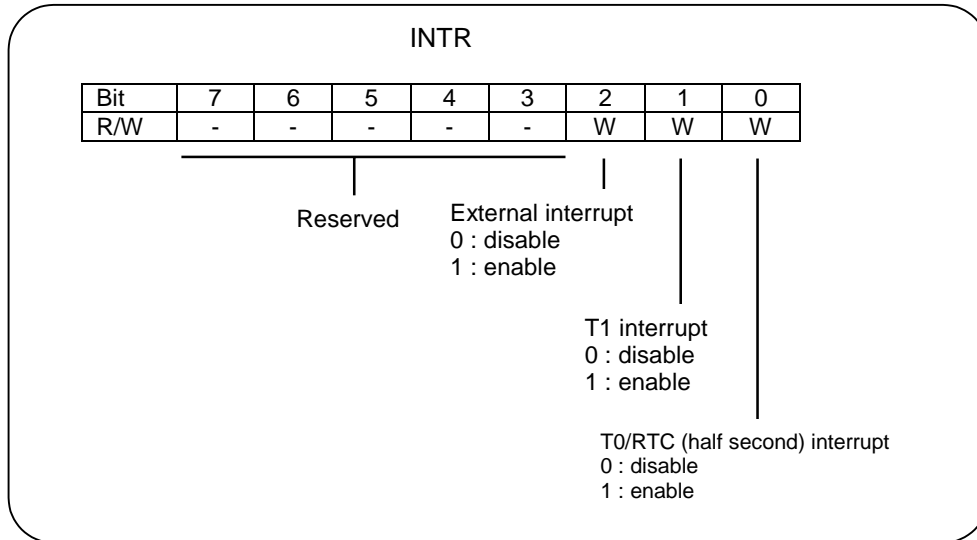


Figure 15. Interrupt Control Register

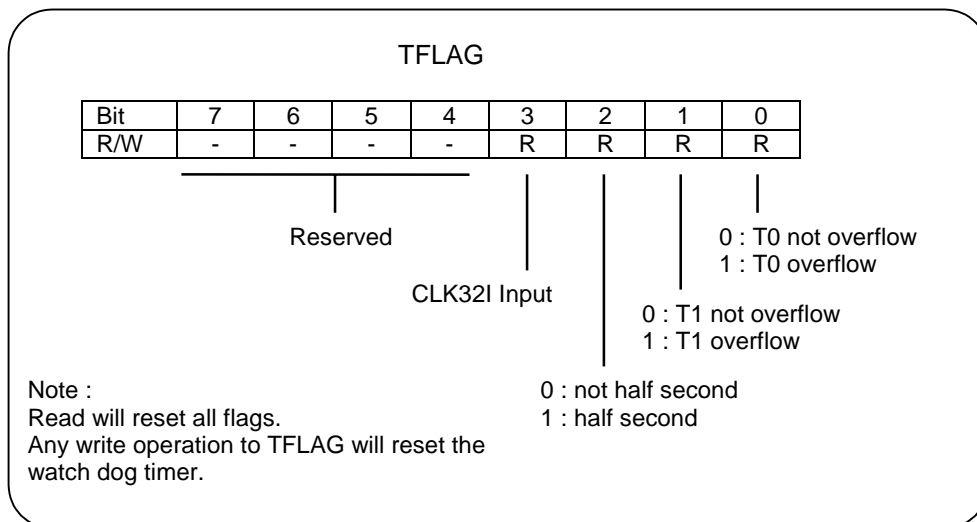


Figure 16. Timer Flag Status Register/Watch-dog Reset

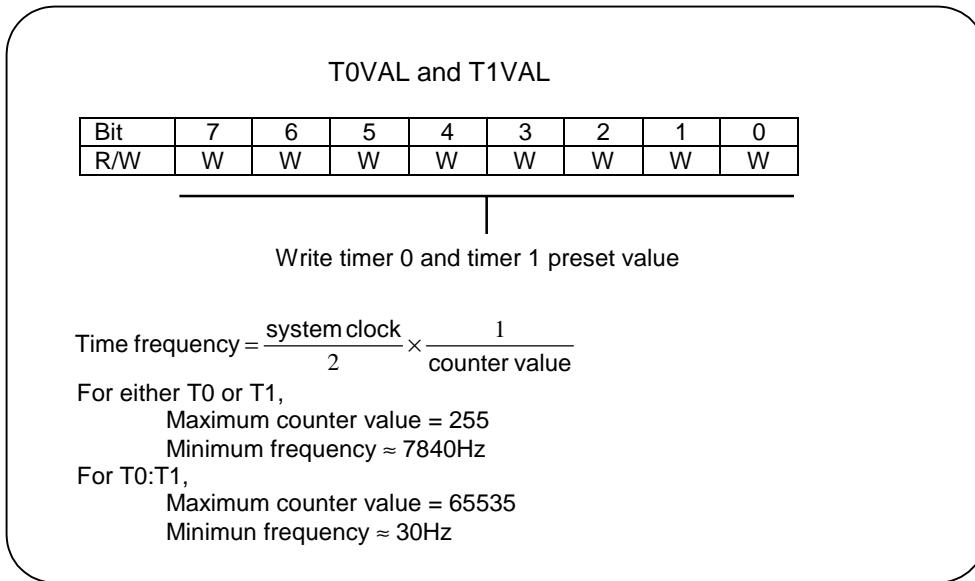


Figure 17. Timer 0 and Timer 1 Preset Value Register

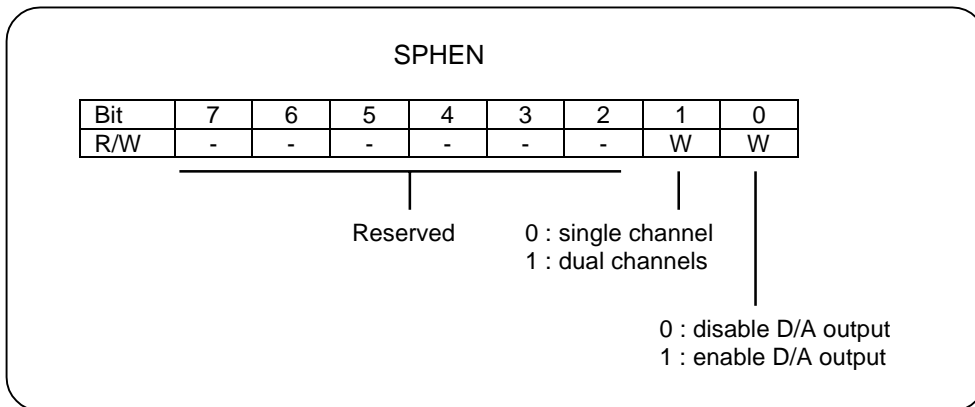


Figure 18. D/A Output Control Register



INSTRUCTION SUMMARY

There are 126 instructions. All instructions are one or two byte instructions.
The followings are the notations used:

A	Accumulator
B	B Register
C	Carry bit
Z	Zero bit
N	Negative bit
X	X Index Register
Y	Y Index Register
SP	Stack Pointer Register
STACK	Stack
BANK	Data Bank Register
PAGE	Page Register
iPAGE	The page register for PC19-PC14
PC	Program Counter PC13-0
PS	Program Status
I	Interrupt flag
(X)	RAM pointed by X
(Y)	RAM pointed by Y
label	A 8-bit RAM/Register label
(label)	RAM pointed by the label
ADDR14	A 14-bit address
ADDR 8	A 8-bit address
#CONSTANT	A 8-bit constant

INSTRUCTION SET

- | | | | | | | | | | | | | | | | | | | | |
|----------------------|--|----------------------|--|------|---------------------|-----------|-------------------------------------|-------|---------|-----|---|----------------------|--|------|---------------------|-----------|--------------------------------------|-------|-------|
| 1. | <table border="1"> <tr><td colspan="2">ADC A</td></tr> <tr><td>Code</td><td>0100 0000</td></tr> <tr><td>Operation</td><td>$A + A + C \rightarrow A$</td></tr> <tr><td>Flags</td><td>N, Z, C</td></tr> </table> | ADC A | | Code | 0100 0000 | Operation | $A + A + C \rightarrow A$ | Flags | N, Z, C | 9. | <table border="1"> <tr><td colspan="2">AND X</td></tr> <tr><td>Code</td><td>0100 1010</td></tr> <tr><td>Operation</td><td>$X \wedge A \rightarrow A$</td></tr> <tr><td>Flags</td><td>N,Z</td></tr> </table> | AND X | | Code | 0100 1010 | Operation | $X \wedge A \rightarrow A$ | Flags | N,Z |
| ADC A | | | | | | | | | | | | | | | | | | | |
| Code | 0100 0000 | | | | | | | | | | | | | | | | | | |
| Operation | $A + A + C \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N, Z, C | | | | | | | | | | | | | | | | | | |
| AND X | | | | | | | | | | | | | | | | | | | |
| Code | 0100 1010 | | | | | | | | | | | | | | | | | | |
| Operation | $X \wedge A \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N,Z | | | | | | | | | | | | | | | | | | |
| 2. | <table border="1"> <tr><td colspan="2">ADC X</td></tr> <tr><td>Code</td><td>0100 0010</td></tr> <tr><td>Operation</td><td>$X + A + C \rightarrow A$</td></tr> <tr><td>Flags</td><td>N, Z, C</td></tr> </table> | ADC X | | Code | 0100 0010 | Operation | $X + A + C \rightarrow A$ | Flags | N, Z, C | 10. | <table border="1"> <tr><td colspan="2">AND Y</td></tr> <tr><td>Code</td><td>0100 1011</td></tr> <tr><td>Operation</td><td>$Y \wedge A \rightarrow A$</td></tr> <tr><td>Flags</td><td>N,Z</td></tr> </table> | AND Y | | Code | 0100 1011 | Operation | $Y \wedge A \rightarrow A$ | Flags | N,Z |
| ADC X | | | | | | | | | | | | | | | | | | | |
| Code | 0100 0010 | | | | | | | | | | | | | | | | | | |
| Operation | $X + A + C \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N, Z, C | | | | | | | | | | | | | | | | | | |
| AND Y | | | | | | | | | | | | | | | | | | | |
| Code | 0100 1011 | | | | | | | | | | | | | | | | | | |
| Operation | $Y \wedge A \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N,Z | | | | | | | | | | | | | | | | | | |
| 3. | <table border="1"> <tr><td colspan="2">ADC Y</td></tr> <tr><td>Code</td><td>0100 0011</td></tr> <tr><td>Operation</td><td>$Y + A + C \rightarrow A$</td></tr> <tr><td>Flags</td><td>N, Z, C</td></tr> </table> | ADC Y | | Code | 0100 0011 | Operation | $Y + A + C \rightarrow A$ | Flags | N, Z, C | 11. | <table border="1"> <tr><td colspan="2">AND (X)</td></tr> <tr><td>Code</td><td>0100 1110</td></tr> <tr><td>Operation</td><td>$(X) \wedge A \rightarrow A$</td></tr> <tr><td>Flags</td><td>N,Z</td></tr> </table> | AND (X) | | Code | 0100 1110 | Operation | $(X) \wedge A \rightarrow A$ | Flags | N,Z |
| ADC Y | | | | | | | | | | | | | | | | | | | |
| Code | 0100 0011 | | | | | | | | | | | | | | | | | | |
| Operation | $Y + A + C \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N, Z, C | | | | | | | | | | | | | | | | | | |
| AND (X) | | | | | | | | | | | | | | | | | | | |
| Code | 0100 1110 | | | | | | | | | | | | | | | | | | |
| Operation | $(X) \wedge A \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N,Z | | | | | | | | | | | | | | | | | | |
| 4. | <table border="1"> <tr><td colspan="2">ADC (X)</td></tr> <tr><td>Code</td><td>0100 0110</td></tr> <tr><td>Operation</td><td>$(X) + A + C \rightarrow A$</td></tr> <tr><td>Flags</td><td>N, Z, C</td></tr> </table> | ADC (X) | | Code | 0100 0110 | Operation | $(X) + A + C \rightarrow A$ | Flags | N, Z, C | 12. | <table border="1"> <tr><td colspan="2">AND (Y)</td></tr> <tr><td>Code</td><td>0100 1111</td></tr> <tr><td>Operation</td><td>$(Y) \wedge A \rightarrow A$</td></tr> <tr><td>Flags</td><td>N,Z</td></tr> </table> | AND (Y) | | Code | 0100 1111 | Operation | $(Y) \wedge A \rightarrow A$ | Flags | N,Z |
| ADC (X) | | | | | | | | | | | | | | | | | | | |
| Code | 0100 0110 | | | | | | | | | | | | | | | | | | |
| Operation | $(X) + A + C \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N, Z, C | | | | | | | | | | | | | | | | | | |
| AND (Y) | | | | | | | | | | | | | | | | | | | |
| Code | 0100 1111 | | | | | | | | | | | | | | | | | | |
| Operation | $(Y) \wedge A \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N,Z | | | | | | | | | | | | | | | | | | |
| 5. | <table border="1"> <tr><td colspan="2">ADC (Y)</td></tr> <tr><td>Code</td><td>0100 0111</td></tr> <tr><td>Operation</td><td>$(Y) + A + C \rightarrow A$</td></tr> <tr><td>Flags</td><td>N, Z, C</td></tr> </table> | ADC (Y) | | Code | 0100 0111 | Operation | $(Y) + A + C \rightarrow A$ | Flags | N, Z, C | 13. | <table border="1"> <tr><td colspan="2">AND label</td></tr> <tr><td>Code</td><td>0100 1100 zzzz zzzz</td></tr> <tr><td>Operation</td><td>$(zzzz zzzz) \wedge A \rightarrow A$</td></tr> <tr><td>Flags</td><td>N,Z</td></tr> </table> | AND label | | Code | 0100 1100 zzzz zzzz | Operation | $(zzzz zzzz) \wedge A \rightarrow A$ | Flags | N,Z |
| ADC (Y) | | | | | | | | | | | | | | | | | | | |
| Code | 0100 0111 | | | | | | | | | | | | | | | | | | |
| Operation | $(Y) + A + C \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N, Z, C | | | | | | | | | | | | | | | | | | |
| AND label | | | | | | | | | | | | | | | | | | | |
| Code | 0100 1100 zzzz zzzz | | | | | | | | | | | | | | | | | | |
| Operation | $(zzzz zzzz) \wedge A \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N,Z | | | | | | | | | | | | | | | | | | |
| 6. | <table border="1"> <tr><td colspan="2">ADC label</td></tr> <tr><td>Code</td><td>0100 0100 zzzz zzzz</td></tr> <tr><td>Operation</td><td>$(zzzz zzzz) + A + C \rightarrow A$</td></tr> <tr><td>Flags</td><td>N, Z, C</td></tr> </table> | ADC label | | Code | 0100 0100 zzzz zzzz | Operation | $(zzzz zzzz) + A + C \rightarrow A$ | Flags | N, Z, C | 14. | <table border="1"> <tr><td colspan="2">AND #CONSTANT</td></tr> <tr><td>Code</td><td>0100 1001 zzzz zzzz</td></tr> <tr><td>Operation</td><td>$Zzzz zzzz \wedge A \rightarrow A$</td></tr> <tr><td>Flags</td><td>N,Z</td></tr> </table> | AND #CONSTANT | | Code | 0100 1001 zzzz zzzz | Operation | $Zzzz zzzz \wedge A \rightarrow A$ | Flags | N,Z |
| ADC label | | | | | | | | | | | | | | | | | | | |
| Code | 0100 0100 zzzz zzzz | | | | | | | | | | | | | | | | | | |
| Operation | $(zzzz zzzz) + A + C \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N, Z, C | | | | | | | | | | | | | | | | | | |
| AND #CONSTANT | | | | | | | | | | | | | | | | | | | |
| Code | 0100 1001 zzzz zzzz | | | | | | | | | | | | | | | | | | |
| Operation | $Zzzz zzzz \wedge A \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N,Z | | | | | | | | | | | | | | | | | | |
| 7. | <table border="1"> <tr><td colspan="2">ADC #CONSTANT</td></tr> <tr><td>Code</td><td>0100 0001 zzzz zzzz</td></tr> <tr><td>Operation</td><td>$Zzzz zzzz + A + C \rightarrow A$</td></tr> <tr><td>Flags</td><td>N, Z, C</td></tr> </table> | ADC #CONSTANT | | Code | 0100 0001 zzzz zzzz | Operation | $Zzzz zzzz + A + C \rightarrow A$ | Flags | N, Z, C | 15. | <table border="1"> <tr><td colspan="2">BRA ADDR8</td></tr> <tr><td>Code</td><td>0011 0010 zzzz zzzz</td></tr> <tr><td>Operation</td><td>$Zzzz zzzz \rightarrow PC0-7$</td></tr> <tr><td>Flags</td><td>-----</td></tr> </table> | BRA ADDR8 | | Code | 0011 0010 zzzz zzzz | Operation | $Zzzz zzzz \rightarrow PC0-7$ | Flags | ----- |
| ADC #CONSTANT | | | | | | | | | | | | | | | | | | | |
| Code | 0100 0001 zzzz zzzz | | | | | | | | | | | | | | | | | | |
| Operation | $Zzzz zzzz + A + C \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N, Z, C | | | | | | | | | | | | | | | | | | |
| BRA ADDR8 | | | | | | | | | | | | | | | | | | | |
| Code | 0011 0010 zzzz zzzz | | | | | | | | | | | | | | | | | | |
| Operation | $Zzzz zzzz \rightarrow PC0-7$ | | | | | | | | | | | | | | | | | | |
| Flags | ----- | | | | | | | | | | | | | | | | | | |
| 8. | <table border="1"> <tr><td colspan="2">AND A</td></tr> <tr><td>Code</td><td>0100 1000</td></tr> <tr><td>Operation</td><td>$A \wedge A \rightarrow A$</td></tr> <tr><td>Flags</td><td>N,Z</td></tr> </table> | AND A | | Code | 0100 1000 | Operation | $A \wedge A \rightarrow A$ | Flags | N,Z | | | | | | | | | | |
| AND A | | | | | | | | | | | | | | | | | | | |
| Code | 0100 1000 | | | | | | | | | | | | | | | | | | |
| Operation | $A \wedge A \rightarrow A$ | | | | | | | | | | | | | | | | | | |
| Flags | N,Z | | | | | | | | | | | | | | | | | | |



16.	<table border="1"> <thead> <tr> <th colspan="2">CALL ADDR14</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>11zz zzzz zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>PC \rightarrow(SP);SP-2\rightarrowSP</td> </tr> <tr> <td></td> <td>ADDR14 \rightarrowPC[13:0]</td> </tr> <tr> <td></td> <td>If PAGE \neq iPAGE</td> </tr> <tr> <td></td> <td>IPAGE\rightarrow(SP);SP-1\rightarrowSP</td> </tr> <tr> <td></td> <td>PAGE \rightarrowiPAGE</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	CALL ADDR14		Code	11zz zzzz zzzz zzzz	Operation	PC \rightarrow (SP);SP-2 \rightarrow SP		ADDR14 \rightarrow PC[13:0]		If PAGE \neq iPAGE		IPAGE \rightarrow (SP);SP-1 \rightarrow SP		PAGE \rightarrow iPAGE	Flags	-----	24.	<table border="1"> <thead> <tr> <th colspan="2">CMP #CONSTANT</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0000 0001 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>A ∇ zzzz zzzz</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CMP #CONSTANT		Code	0000 0001 zzzz zzzz	Operation	A ∇ zzzz zzzz	Flags	Z
CALL ADDR14																											
Code	11zz zzzz zzzz zzzz																										
Operation	PC \rightarrow (SP);SP-2 \rightarrow SP																										
	ADDR14 \rightarrow PC[13:0]																										
	If PAGE \neq iPAGE																										
	IPAGE \rightarrow (SP);SP-1 \rightarrow SP																										
	PAGE \rightarrow iPAGE																										
Flags	-----																										
CMP #CONSTANT																											
Code	0000 0001 zzzz zzzz																										
Operation	A ∇ zzzz zzzz																										
Flags	Z																										
17.	<table border="1"> <thead> <tr> <th colspan="2">CLR A</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1000</td> </tr> <tr> <td>Operation</td> <td>0 \rightarrowA</td> </tr> <tr> <td>Flags</td> <td>N,Z</td> </tr> </tbody> </table>	CLR A		Code	0010 1000	Operation	0 \rightarrow A	Flags	N,Z	25.	<table border="1"> <thead> <tr> <th colspan="2">CMP label</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0000 0101 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>A ∇ (zzzz zzzz)</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CMP label		Code	0000 0101 zzzz zzzz	Operation	A ∇ (zzzz zzzz)	Flags	Z								
CLR A																											
Code	0010 1000																										
Operation	0 \rightarrow A																										
Flags	N,Z																										
CMP label																											
Code	0000 0101 zzzz zzzz																										
Operation	A ∇ (zzzz zzzz)																										
Flags	Z																										
18.	<table border="1"> <thead> <tr> <th colspan="2">CLR X</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1010</td> </tr> <tr> <td>Operation</td> <td>0 \rightarrowX</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	CLR X		Code	0010 1010	Operation	0 \rightarrow X	Flags	-----	26.	<table border="1"> <thead> <tr> <th colspan="2">CPMX #CONSTANT</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0000 1001 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>(X) ∇ zzzz zzzz</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CPMX #CONSTANT		Code	0000 1001 zzzz zzzz	Operation	(X) ∇ zzzz zzzz	Flags	Z								
CLR X																											
Code	0010 1010																										
Operation	0 \rightarrow X																										
Flags	-----																										
CPMX #CONSTANT																											
Code	0000 1001 zzzz zzzz																										
Operation	(X) ∇ zzzz zzzz																										
Flags	Z																										
19.	<table border="1"> <thead> <tr> <th colspan="2">CLR Y</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1011</td> </tr> <tr> <td>Operation</td> <td>0 \rightarrowY</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	CLR Y		Code	0010 1011	Operation	0 \rightarrow Y	Flags	-----	27.	<table border="1"> <thead> <tr> <th colspan="2">CPMX label</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0000 1101 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>(X) ∇ zzzz zzzz</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CPMX label		Code	0000 1101 zzzz zzzz	Operation	(X) ∇ zzzz zzzz	Flags	Z								
CLR Y																											
Code	0010 1011																										
Operation	0 \rightarrow Y																										
Flags	-----																										
CPMX label																											
Code	0000 1101 zzzz zzzz																										
Operation	(X) ∇ zzzz zzzz																										
Flags	Z																										
20.	<table border="1"> <thead> <tr> <th colspan="2">CLR (X)</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1110</td> </tr> <tr> <td>Operation</td> <td>0 \rightarrow(X)</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	CLR (X)		Code	0010 1110	Operation	0 \rightarrow (X)	Flags	-----	28.	<table border="1"> <thead> <tr> <th colspan="2">CPX #CONSTANT</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0001 0001 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>X ∇ zzzz zzzz</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CPX #CONSTANT		Code	0001 0001 zzzz zzzz	Operation	X ∇ zzzz zzzz	Flags	Z								
CLR (X)																											
Code	0010 1110																										
Operation	0 \rightarrow (X)																										
Flags	-----																										
CPX #CONSTANT																											
Code	0001 0001 zzzz zzzz																										
Operation	X ∇ zzzz zzzz																										
Flags	Z																										
21.	<table border="1"> <thead> <tr> <th colspan="2">CLR (Y)</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1111</td> </tr> <tr> <td>Operation</td> <td>0 \rightarrow(Y)</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	CLR (Y)		Code	0010 1111	Operation	0 \rightarrow (Y)	Flags	-----	29.	<table border="1"> <thead> <tr> <th colspan="2">CPX label</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0001 0101 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>X ∇ (zzzz zzzz)</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CPX label		Code	0001 0101 zzzz zzzz	Operation	X ∇ (zzzz zzzz)	Flags	Z								
CLR (Y)																											
Code	0010 1111																										
Operation	0 \rightarrow (Y)																										
Flags	-----																										
CPX label																											
Code	0001 0101 zzzz zzzz																										
Operation	X ∇ (zzzz zzzz)																										
Flags	Z																										
22.	<table border="1"> <thead> <tr> <th colspan="2">CLR label</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1100</td> </tr> <tr> <td>Operation</td> <td>0 \rightarrow(zzzz zzzz)</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	CLR label		Code	0010 1100	Operation	0 \rightarrow (zzzz zzzz)	Flags	-----	30.	<table border="1"> <thead> <tr> <th colspan="2">CPY #CONSTANT</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0001 1001 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>Y ∇ zzzz zzzz</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CPY #CONSTANT		Code	0001 1001 zzzz zzzz	Operation	Y ∇ zzzz zzzz	Flags	Z								
CLR label																											
Code	0010 1100																										
Operation	0 \rightarrow (zzzz zzzz)																										
Flags	-----																										
CPY #CONSTANT																											
Code	0001 1001 zzzz zzzz																										
Operation	Y ∇ zzzz zzzz																										
Flags	Z																										
23.	<table border="1"> <thead> <tr> <th colspan="2">CLRC</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0110 0001</td> </tr> <tr> <td>Operation</td> <td>0 \rightarrowC</td> </tr> <tr> <td>Flags</td> <td>C</td> </tr> </tbody> </table>	CLRC		Code	0110 0001	Operation	0 \rightarrow C	Flags	C	31.	<table border="1"> <thead> <tr> <th colspan="2">CPY label</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0001 1101 zzzz zzzz</td> </tr> <tr> <td>Operation</td> <td>Y ∇ (zzzz zzzz)</td> </tr> <tr> <td>Flags</td> <td>Z</td> </tr> </tbody> </table>	CPY label		Code	0001 1101 zzzz zzzz	Operation	Y ∇ (zzzz zzzz)	Flags	Z								
CLRC																											
Code	0110 0001																										
Operation	0 \rightarrow C																										
Flags	C																										
CPY label																											
Code	0001 1101 zzzz zzzz																										
Operation	Y ∇ (zzzz zzzz)																										
Flags	Z																										



32.	DEC A
Code	0110 1000
Operation	$A - 1 \rightarrow A$
Flags	N, Z, C

33.	DEC X
Code	0110 1010
Operation	$X - 1 \rightarrow X$
Flags	N, Z, C

34.	DEC Y
Code	0110 1011
Operation	$Y - 1 \rightarrow Y$
Flags	N, Z, C

35.	DEC (X)
Code	0110 1110
Operation	$(X) - 1 \rightarrow (X)$
Flags	N, Z, C

36.	DEC (Y)
Code	0110 1111
Operation	$(Y) - 1 \rightarrow (Y)$
Flags	N, Z, C

37.	DEC label
Code	0110 1100 zzzz zzzz
Operation	$(zzzz\ zzzz) - 1 \rightarrow (zzzzzzzz)$
Flags	N, Z, C

38.	EOR X
Code	0101 1010
Operation	$X \vee A \rightarrow A$
Flags	N, Z

39.	EOR Y
Code	0101 1011
Operation	$Y \vee A \rightarrow A$
Flags	N, Z

40.	EOR (X)
Code	0101 1110
Operation	$(X) \vee A \rightarrow A$
Flags	N, Z

41.	EOR (Y)
Code	0101 1111
Operation	$(Y) \vee A \rightarrow A$
Flags	N, Z

42.	EOR label
Code	0101 1100 zzzz zzzz
Operation	$(zzzz\ zzzz) \vee A \rightarrow A$
Flags	N, Z

43.	EOR #CONSTANT
Code	0101 1001 zzzz zzzz
Operation	$zzzz\ zzzz \vee A \rightarrow A$
Flags	N, Z

44.	INC A
Code	0110 0000
Operation	$A + 1 \rightarrow A$
Flags	N, Z, C

45.	INC X
Code	0110 0010
Operation	$X + 1 \rightarrow X$
Flags	N, Z, C

46.	INC Y
Code	0110 0011
Operation	$Y + 1 \rightarrow Y$
Flags	N, Z, C

47.	INC (X)
Code	0110 0110
Operation	$(X) + 1 \rightarrow X$
Flags	N, Z, C

48.	INC (Y)
Code	0110 0111
Operation	$(Y) + 1 \rightarrow Y$
Flags	N, Z, C



49.	INC label
Code	0110 0100 zzzz zzzz
Operation	(zzzz zzzz) + 1 → (zzzz zzzz)
Flags	N, Z, C

50.	INT
Code	0011 0110
Operation	PC LOW → (SP), SP-1 → SP
	PCHIGH → (SP), SP-1 → SP
	IPAGE → (SP), SP-1 → SP
	PS → (SP), SP-1 → SP
	1 → I
	3FC0H → PC
	3 → iPAGE, PAGE
Flags	I

51.	JA
Code	0111 0001
Operation	A → PC[7:0]
Flags	-----

52.	JUMP ADDR14
Code	10zz zzzz zzzz zzzz
Operation	ADDR14 → PC[13:0]
	If PAGE ≠ iPAGE PAGE → iPAGE
Flags	-----

53.	MOV A, X
Code	0000 0010
Operation	A → X
Flags	-----

54.	MOV A, Y
Code	0000 0011
Operation	A → Y
Flags	-----

55.	MOV A, (X)
Code	0000 0110
Operation	A → (X)
Flags	-----

56.	MOV A, (Y)
Code	0000 0111
Operation	A → (Y)
Flags	-----

57.	MOV A label
Code	0000 0100 zzzz zzzz
Operation	A → (zzzz zzzz)
Flags	-----

58.	MOV X, A
Code	0001 0000
Operation	X → A
Flags	N, Z

59.	MOV X, Y
Code	0001 0011
Operation	X → Y
Flags	-----

60.	MOV X, (Y)
Code	0001 0111
Operation	X → (Y)
Flags	-----

61.	MOV X, label
Code	0001 0100 zzzz zzzz
Operation	X → (zzzz zzzz)
Flags	-----

62.	MOV Y, A
Code	0001 1000
Operation	Y → A
Flags	N, Z



63.	MOV Y, X
Code	0001 1010
Operation	Y →X
Flags	-----

64.	MOV Y, (X)
Code	0001 1110
Operation	Y →(X)
Flags	-----

65.	MOV Y, label
Code	0001 1100 zzzz zzzz
Operation	Y →(zzzz zzzz)
Flags	-----

66.	MOV (X), A
Code	0011 0000
Operation	(X) →A
Flags	N, Z

67.	MOV (X), Y
Code	0011 0011
Operation	(X) →Y
Flags	-----

68.	MOV (X), (Y)
Code	0011 0111
Operation	(X) →(Y)
Flags	-----

69.	MOV (X), label
Code	0011 0100 zzzz zzzz
Operation	(X) →(zzzz zzzz)
Flags	-----

70.	MOV (Y), A
Code	0011 1000
Operation	(Y) →A
Flags	N, Z

71.	MOV (Y), X
Code	0011 1010
Operation	(Y) →X
Flags	-----

72.	MOV (Y), (X)
Code	0011 1110
Operation	(Y) →(X)
Flags	-----

73.	MOV (Y), label
Code	0011 1100 zzzz zzzz
Operation	(Y) →(zzzz zzzz)
Flags	-----

74.	MOV label, A
Code	0010 0000 zzzz zzzz
Operation	(zzzz zzzz) →A
Flags	N, Z

75.	MOV label, X
Code	0010 0010 zzzz zzzz
Operation	(zzzz zzzz) →X
Flags	-----

76.	MOV label, Y
Code	0010 0011 zzzz zzzz
Operation	(zzzz zzzz) →Y
Flags	-----

77.	MOV label, (X)
Code	0010 0110 zzzz zzzz
Operation	(zzzz zzzz) →(X)
Flags	-----

78.	MOV label, (Y)
Code	0010 0111 zzzz zzzz
Operation	(zzzz zzzz) →(Y)
Flags	-----

79.	MOV #CONSTANT, A
Code	0000 1000 zzzz zzzz
Operation	Zzzz zzzz →A
Flags	N, Z



80.	MOV #CONSTANT, X		
	Code	0000 1010 zzzz zzzz	
	Operation	Zzzz zzzz →X	
	Flags	-----	
81.	MOV #CONSTANT, Y		
	Code	0000 1011 zzzz zzzz	
	Operation	Zzzz zzzz →Y	
	Flags	-----	
82.	MOV #CONSTANT, (X)		
	Code	0000 1110 zzzz zzzz	
	Operation	Zzzz zzzz →(X)	
	Flags	-----	
83.	MOV #CONSTANT, (Y)		
	Code	0000 1111 zzzz zzzz	
	Operation	zzzz zzzz →(Y)	
	Flags	-----	
84.	NOP		
	Code	0000 0000	
	Operation	No operation	
	Flags	-----	
85.	OR X		
	Code	0101 0010	
	Operation	X V A→A	
	Flags	N, Z	
86.	OR Y		
	Code	0101 0011	
	Operation	Y V A→A	
	Flags	N, Z	
87.	OR (X)		
	Code	0101 0110	
	Operation	(X) V A→A	
	Flags	N, Z	
88.	OR (Y)		
	Code	0101 0111	
	Operation	(Y) V A→A	
	Flags	N, Z	
89.	OR label		
	Code	0101 0100 zzzz zzzz	
	Operation	(zzzz zzzz)V A→A	
	Flags	N, Z	
90.	OR #CONSTANT		
	Code	0101 0001 zzzz zzzz	
	Operation	Zzzz zzzzV A→A	
	Flags	N, Z	
91.	POP		
	Code	0011 1111	
	Operation	SP+1→SP	
		(SP)→A	
	Flags	-----	
92.	PSHPAGE		
	Code	0110 1101	
	Operation	PAGE→(SP); SP-1→SP	
	Flags	-----	
93.	PUSH		
	Code	0110 1001	
	Operation	A→(SP),SP-1→SP	
	Flags	-----	
94.	ROL A		
	Code	0111 0000	
	Operation	C←A←C	
	Flags	N, Z, C	
95.	ROL X		
	Code	0111 0010	
	Operation	C←X←C	
	Flags	N, Z, C	
96.	ROL Y		
	Code	0111 0011	
	Operation	C←Y←C	
	Flags	N, Z, C	



97.	ROL (X)	Code	0111 0110	Operation	$C \leftarrow (X) \leftarrow C$	Flags	N, Z, C
98.	ROL (Y)	Code	0111 0111	Operation	$C \leftarrow (Y) \leftarrow C$	Flags	N, Z, C
99.	ROL label	Code	0111 0100 zzzz zzzz	Operation	$C \leftarrow (zzzz\ zzzz) \leftarrow C$	Flags	N, Z, C
100.	ROM A	Code	0111 1101	Operation	ROM (BANK,X,A)→A	Flags	-----
101.	ROM Y	Code	0111 1001	Operation	ROM (BANK,X,Y)→A	Flags	-----
102.	ROMDBL	Code	0001 0010	Operation	ROM (BANK,X,Y)→B ROM (BANK,X,Y+1)→A	Flags	-----
103.	ROR A	Code	0111 1000	Operation	$C \rightarrow A \rightarrow C$	Flags	N, Z, C
104.	ROR X	Code	0111 1010	Operation	$C \rightarrow X \rightarrow C$	Flags	N, Z, C
105.	ROR Y	Code	0111 1011	Operation	$C \rightarrow Y \rightarrow C$	Flags	N, Z, C
106.	ROR (X)	Code	0111 1110	Operation	$C \rightarrow (X) \rightarrow C$	Flags	N, Z, C
107.	ROR (Y)	Code	0111 1111	Operation	$C \rightarrow (Y) \rightarrow C$	Flags	N, Z, C
108.	ROR label	Code	0111 1100 zzzz zzzz	Operation	$C \rightarrow (zzzz\ zzzz) \rightarrow C$	Flags	N, Z, C
109.	RTI	Code	0100 0101	Operation	SP+1→SP (SP)→PS; SP+1→1 (SP)→IPAGE,PAGE; SP+1→SP (SP)→PCHIGH; SP+1→SP (SP)→PCLOW	Flags	-----
110.	RTL	Code	0100 1101	Operation	SP+1→SP (SP)→iPAGE,PAGE; SP+1→SP (SP)→PCHIGH SP+1→SP (SP)→PCLOW	Flags	-----
111.	RTS	Code	0101 0101	Operation	SP+1→SP (SP)→PCHIGH; SP+1→SP (SP)→PCLOW	Flags	-----

112.	<table border="1"> <thead> <tr> <th colspan="2">SETC</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0110 0101</td> </tr> <tr> <td>Operation</td> <td>1→C</td> </tr> <tr> <td>Flags</td> <td>C</td> </tr> </tbody> </table>	SETC		Code	0110 0101	Operation	1→C	Flags	C	120.	<table border="1"> <thead> <tr> <th colspan="2">STOP</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 0100</td> </tr> <tr> <td>Operation</td> <td>Stop</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	STOP		Code	0010 0100	Operation	Stop	Flags	-----		
SETC																					
Code	0110 0101																				
Operation	1→C																				
Flags	C																				
STOP																					
Code	0010 0100																				
Operation	Stop																				
Flags	-----																				
113.	<table border="1"> <thead> <tr> <th colspan="2">SETZ</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0101 1000</td> </tr> <tr> <td>Operation</td> <td>1→Z,0→A,</td> </tr> <tr> <td>Flags</td> <td>N, Z</td> </tr> </tbody> </table>	SETZ		Code	0101 1000	Operation	1→Z,0→A,	Flags	N, Z	121.	<table border="1"> <thead> <tr> <th colspan="2">TAPAGE</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0111 0101</td> </tr> <tr> <td>Operation</td> <td>A→PAGE</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	TAPAGE		Code	0111 0101	Operation	A→PAGE	Flags	-----		
SETZ																					
Code	0101 1000																				
Operation	1→Z,0→A,																				
Flags	N, Z																				
TAPAGE																					
Code	0111 0101																				
Operation	A→PAGE																				
Flags	-----																				
114.	<table border="1"> <thead> <tr> <th colspan="2">SKIPC</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0011 0001</td> </tr> <tr> <td>Operation</td> <td>PC+2→PC if C = 1</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	SKIPC		Code	0011 0001	Operation	PC+2→PC if C = 1	Flags	-----	122.	<table border="1"> <thead> <tr> <th colspan="2">TZ</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0101 0000</td> </tr> <tr> <td>Operation</td> <td>Z=1 if A = 0</td> </tr> <tr> <td></td> <td>Z=0 if A ≠ 0</td> </tr> <tr> <td>Flags</td> <td>N, Z</td> </tr> </tbody> </table>	TZ		Code	0101 0000	Operation	Z=1 if A = 0		Z=0 if A ≠ 0	Flags	N, Z
SKIPC																					
Code	0011 0001																				
Operation	PC+2→PC if C = 1																				
Flags	-----																				
TZ																					
Code	0101 0000																				
Operation	Z=1 if A = 0																				
	Z=0 if A ≠ 0																				
Flags	N, Z																				
115.	<table border="1"> <thead> <tr> <th colspan="2">SKIPMI</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0001 0110</td> </tr> <tr> <td>Operation</td> <td>PC+2→PC if N = 1</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	SKIPMI		Code	0001 0110	Operation	PC+2→PC if N = 1	Flags	-----	123.	<table border="1"> <thead> <tr> <th colspan="2">XB</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 0101</td> </tr> <tr> <td>Operation</td> <td>B↔A</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	XB		Code	0010 0101	Operation	B↔A	Flags	-----		
SKIPMI																					
Code	0001 0110																				
Operation	PC+2→PC if N = 1																				
Flags	-----																				
XB																					
Code	0010 0101																				
Operation	B↔A																				
Flags	-----																				
116.	<table border="1"> <thead> <tr> <th colspan="2">SKIPNC</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0011 1001</td> </tr> <tr> <td>Operation</td> <td>PC+2→PC if C = 0</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	SKIPNC		Code	0011 1001	Operation	PC+2→PC if C = 0	Flags	-----	124.	<table border="1"> <thead> <tr> <th colspan="2">XBANK</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0011 0101</td> </tr> <tr> <td>Operation</td> <td>BANK↔A</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	XBANK		Code	0011 0101	Operation	BANK↔A	Flags	-----		
SKIPNC																					
Code	0011 1001																				
Operation	PC+2→PC if C = 0																				
Flags	-----																				
XBANK																					
Code	0011 0101																				
Operation	BANK↔A																				
Flags	-----																				
117.	<table border="1"> <thead> <tr> <th colspan="2">SKIPNZ</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1001</td> </tr> <tr> <td>Operation</td> <td>PC+2→PC if Z = 0</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	SKIPNZ		Code	0010 1001	Operation	PC+2→PC if Z = 0	Flags	-----	125.	<table border="1"> <thead> <tr> <th colspan="2">XSP</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0011 1101</td> </tr> <tr> <td>Operation</td> <td>SP↔A</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	XSP		Code	0011 1101	Operation	SP↔A	Flags	-----		
SKIPNZ																					
Code	0010 1001																				
Operation	PC+2→PC if Z = 0																				
Flags	-----																				
XSP																					
Code	0011 1101																				
Operation	SP↔A																				
Flags	-----																				
118.	<table border="1"> <thead> <tr> <th colspan="2">SKIPPL</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0001 1111</td> </tr> <tr> <td>Operation</td> <td>PC+2→PC if N = 0</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	SKIPPL		Code	0001 1111	Operation	PC+2→PC if N = 0	Flags	-----	126.	<table border="1"> <thead> <tr> <th colspan="2">XST</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 1101</td> </tr> <tr> <td>Operation</td> <td>PS↔A</td> </tr> <tr> <td>Flags</td> <td>N, Z, C, I</td> </tr> </tbody> </table>	XST		Code	0010 1101	Operation	PS↔A	Flags	N, Z, C, I		
SKIPPL																					
Code	0001 1111																				
Operation	PC+2→PC if N = 0																				
Flags	-----																				
XST																					
Code	0010 1101																				
Operation	PS↔A																				
Flags	N, Z, C, I																				
119.	<table border="1"> <thead> <tr> <th colspan="2">SKIPZ</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>0010 0001</td> </tr> <tr> <td>Operation</td> <td>PC+2→PC if Z = 1</td> </tr> <tr> <td>Flags</td> <td>-----</td> </tr> </tbody> </table>	SKIPZ		Code	0010 0001	Operation	PC+2→PC if Z = 1	Flags	-----												
SKIPZ																					
Code	0010 0001																				
Operation	PC+2→PC if Z = 1																				
Flags	-----																				

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings:

1. VDD	7.0V
2. VIH	VDD + 0.3V
3. VIL	GND - 0.3V
4. Operating Temperature	0°C to + 60°C
5. Storage Temperature	-50°C to + 150°C

Recommended Operating:

			MIN	TYP	MAX	Unit
1. Operating Voltage (VDD)			2.4	3.0	5.5	V
2. Operating Frequency (Fosc)			-	4.0	-	MHz
3. Input Voltage						
3.1 VDD = 2.4V	VIH		1.6	2.0	2.4	V
	VIL		0	0.4	0.8	V
3.2 VDD = 3.0V	VIH		2.0	2.4	3.0	V
	VIL		0	0.6	1.0	V
3.3 VDD = 5.5V	VIH		3.6	4.0	5.5V	V
	VIL		0	1.5	1.8V	V
4. Output Voltage						
4.1 VDD = 2.4V	VOH		2.0	-	-	V
@1mA	VOL		-	-	0.4	V
4.2 VDD = 3.0V	VOH		2.4	-	-	V
@2mA	VOL		-	-	0.6	V
4.3 VDD = 5.5V	VOH		4.0	-	-	V
@10mA	VOL		-	-	1.5	V
5. Input Current						
VIH = VDD	IH		-	-	150	µA (through pull-low resistor)
VIL = OV	IL		-	-	0.3	µA (leakage)
6. Output Current						
6.1 VDD = 2.4V,	IOH@VOH =2.0V		1.0	1.5	-	mA
	IOL@VOL =0.4V		2.0	3.0	-	mA
6.2 VDD = 3.0V,	IOH@VOH =2.4V		1.5	3.5	-	mA
	IOL@VOL =0.6V		3.0	7.0	-	mA
6.3 VDD = 5.5V,	IOH@VOH =4.0V		8	10	-	mA
	IOL@VOL =1.5V		16	20	-	mA
7. Current Dissipation						
7.1 Operating Current @4MHz, 3.0V			1.0	1.5	4.5	mA
& Output Pad Load = 50pF						
7.2 Operating Current @4MHz, 5.0V			2.0	3.0	9.0	mA
& Output Pad Load = 50pF						
7.3 Standby Current (OFF mode)			-	5.0	30.0	µA



PAD Co-ordinates

	EXT_INT 3 9	RESET- 38	TEST 37	VDD 36											
1	P_CLK											35	VSS		
2	CLK32O											34	PWMM		
3	CLK32I											33	PWMP		
4	VSS											32	P2W6		
5	P0RW0											31	P2W5		
6	P0RW1											30	P2W4		
7	P0RW2											29	VDD		
8	P0RW3											28	P2W3		
9	VDD											27	P2W2		
10	P0RW4														
11	P0RW5														
12	P0RW6														
13	P0RW7														
	1 4 VSS	15 PIRW0	16 PIRW1	17 PIRW2	18 PIRW3	19 VDD	20 PIRW4	21 PIRW5	22 PIRW6	23 PIRW7	24 VSS	25 P2W0	26 P2W1		

The substrate of IC should be connected to VSS

Pad No	Pad Name	X	Y	Pad No	Pad Name	X	Y
1	P_CLK	-1193.600	955.000	20	P1RW4	-24.000	-1174.600
2	CLK32O	-1193.600	795.000	21	P1RW5	136.000	-1174.600
3	CLK32I	-1193.600	635.000	22	P1RW6	296.000	-1174.600
4	VSS	-1193.600	475.000	23	P1RW7	456.000	-1174.600
5	P0RW0	-1193.600	315.000	24	VSS	616.000	-1174.600
6	P0RW1	-1193.600	155.000	25	P2W0	776.000	-1174.600
7	P0RW2	-1193.600	-5.000	26	P2W1	936.000	-1174.600
8	P0RW3	-1193.600	-165.000	27	P2W2	1193.600	-965.000
9	VDD	-1193.600	-325.000	28	P2W3	1193.600	-805.000
10	P0RW4	-1193.600	-485.000	29	VDD	1193.600	-645.000
11	P0RW5	-1193.600	-645.000	30	P2W4	1193.600	-485.000
12	P0RW6	-1193.600	-805.000	31	P2W5	1193.600	-325.000
13	P0RW7	-1193.600	-965.000	32	P2W6	1193.600	-165.000
14	VSS	-984.000	-1174.600	33	PWMP	1193.600	-5.000
15	P1RW0	-824.000	-1174.600	34	PWMM	1193.600	155.000
16	P1RW1	-664.000	-1174.600	35	VSS	1193.600	315.000
17	P1RW2	-504.000	-1174.600	36	VDD	-504.000	1174.600
18	P1RW3	-344.000	-1174.600	37	TEST	-664.000	1174.600
19	VDD	-184.000	-1174.600	38	$\overline{\text{RESET}}$	-824.000	1174.600
				39	EXT_INT	-984.000	1174.600



Application Circuit

