HYNIX SEMICONDUCTOR INC.
8-BIT SINGLE-CHIP MICROCONTROLLERS

GMS81C5108

User's Manual (Ver. 1.0)

---

hynix Semiconductor

Version 1.0

Published by
MCU Application Team

Additional information of this manual may be served by Hynix Semiconductor offices in Korea or Distributors and Representatives listed at address directory.

Hynix Semiconductor reserves the right to make changes to any information here in at any time without notice.

The information, diagrams and other data in this manual are correct and reliable; however, Hynix Semiconductor is in no way responsible for any violations of patents or other rights of the third party generated by the use of this manual.

Table of Contents

GMS81C5108

CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH LCD CONTROLLER/DRIVER AND INFRARED REMOTE CONTROL TRANSMITTERS

1. OVERVIEW

1.1 Description

The GMS81C5108 is an advanced CMOS 8-bit microcontroller with 8K bytes of ROM. The device is one of GMS800 family. The Hynix GMS81C5108 is a powerful microcontroller which provides a high flexibility and cost effective solution to many LCD applications. The GMS81C5108 provides the following standard features: 8K bytes of ROM, 192 bytes of RAM, 37 Nibbles of Display RAM, 8/16-bit timer/counter, on-chip oscillator and clock circuitry. In addition, the GMS81C5108 supports power saving modes to reduce power consumption.
This document is only explained for the base of GMS81C5108, the eliminated functions are same as below.

Device name	ROM Size	OTP Size	RAM Size	I/O	Package
GMS81C5108	8K bytes	-	192 bytes	24	80QFP
GMS87C5108		8K bytes	192bytes	24	80QFP

1.2 Features

- **8K Bytes of On-chip Program Memory**
- **192 Bytes of On-chip Data RAM**
- **37 Nibbles of Display RAM**
- **Instruction Cycle Time:**
  - 1us at 4MHz (2 cycle NOP instruction)
- **24 Programmable I/O pins**
- **2V to 4V Operating Range**
- **Dual Clock Operation**
  - main : 400kHz ~ 4.2MHz
  - sub. : 32.768kHz
- **One 8-bit Basic Interval Timer/Counter**
- **Key Scan Interrupt**
- **Two 8-bit Timer/ Counter**
  (It can be used one 16-bit Timer/Counter)
- **Watch Timer (2Hz, 4Hz, 16Hz, 1/64Hz)**
- **8-bit Serial I/O (SIO)**
- **One 10-bit High Speed PWM Output**
- **Carrier Generator for Remote Controller**
- **11 Interrupt sources**
  - 3 External interrupts (INT0 ~ 2)
  - 8 Internal interrupts (BIT, Timer × 2, WT, A/DC, SIO, REM, Keyscan)
- **6-bit Buzzer Driving port**
  - 500Hz ~ 250kHz (@4MHz)
- **4-channel 8-bit On-chip A/D Converter**
- **Power Saving Mode**
  - STOP, SLEEP, Sub Active mode
- **LCD display/controller (LCDC)**
  - Static Mode (37Seg × 1Com, 1/3 Bias)
  - 1/2 Duty Mode (36Seg × 2Com, 1/3 Bias)
  - 1/3 Duty Mode (35Seg × 3Com, 1/3 Bias)
  - 1/4 Duty Mode (34Seg × 4Com, 1/3 Bias)
- **LCD Display Voltage Booster**
- **Supply Voltage Detector(SVD)**
  - 2 level detector (2.2V, 1.7V)

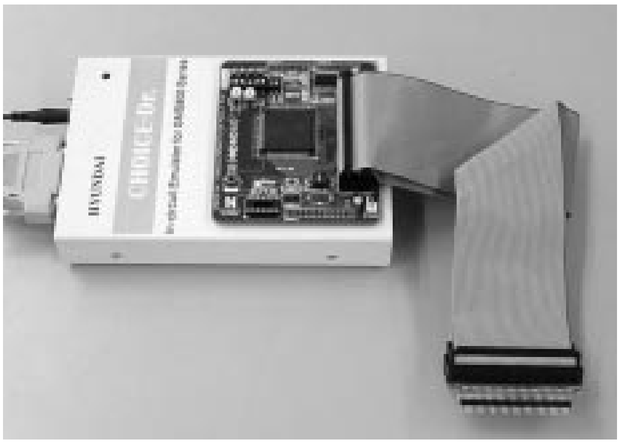
1.3 Development Tools

Note: *There are several setting switches in the Emulator. User should read carefully and do setting properly before developing the program refer to "24.2 Emulator S/W Setting" on page 84. Otherwise, the Emulator may not work properly.*

Software	- MS- Window base assembler - Linker / Editor / Debugger
Hardware (Emulator)	- CHOICE-Dr. - CHOICE-Dr. EVA 81C51 B/D
OTP Writer	- CHOICE - SIGMA (Single writer) - CHOICE - GANG4 (Gang writer)

The GMS81C5108 is supported by a full-featured macro assembler, an in-circuit emulator CHOICE-Dr.™ and OTP programmers. There are two different type programmers such as single type and gang type. For mode detail, refer to OTP Programming chapter. Macro assembler operates under the MS-Windows 95/98™.

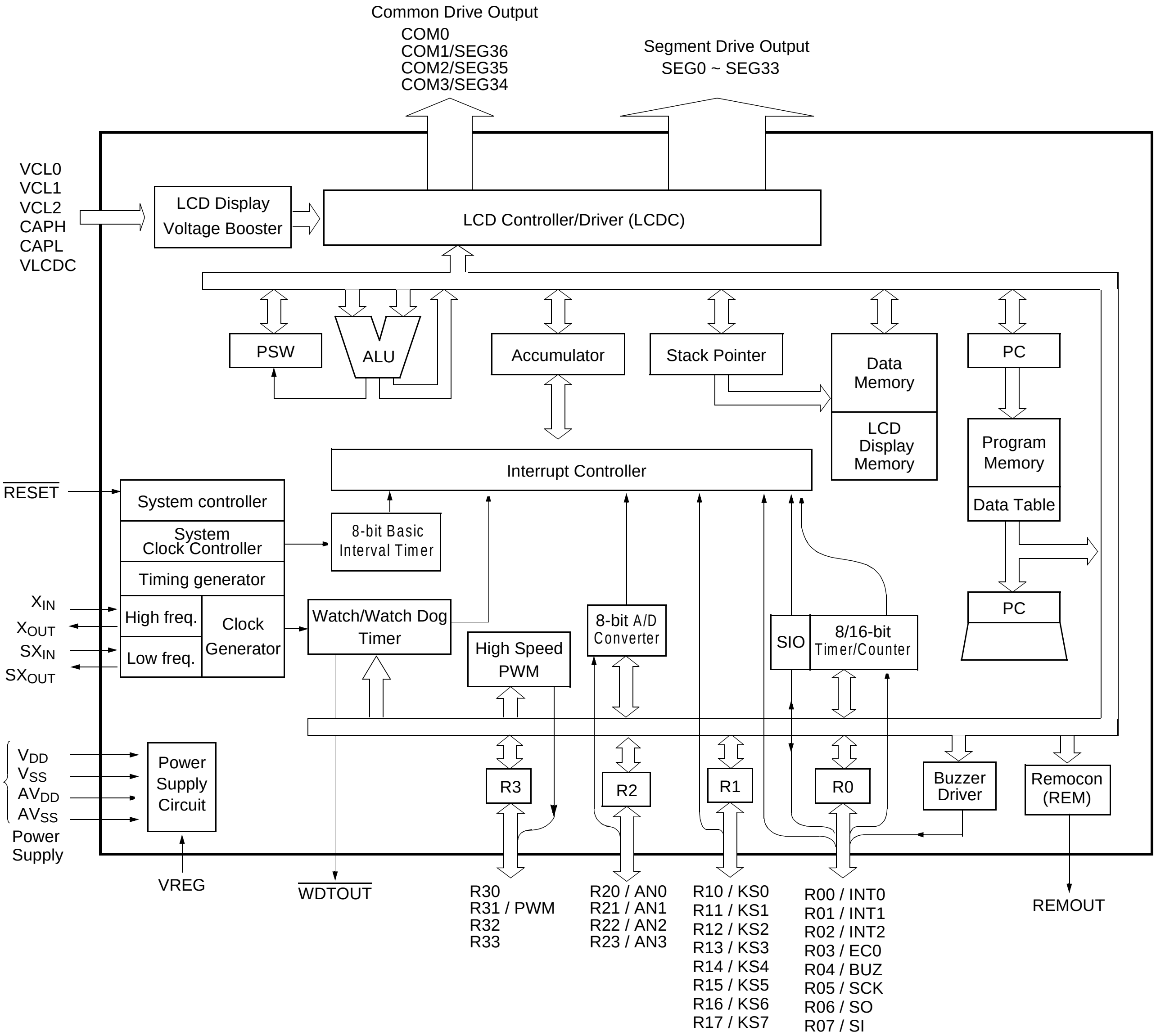
Please contact sales part of Hynix Semiconductor.



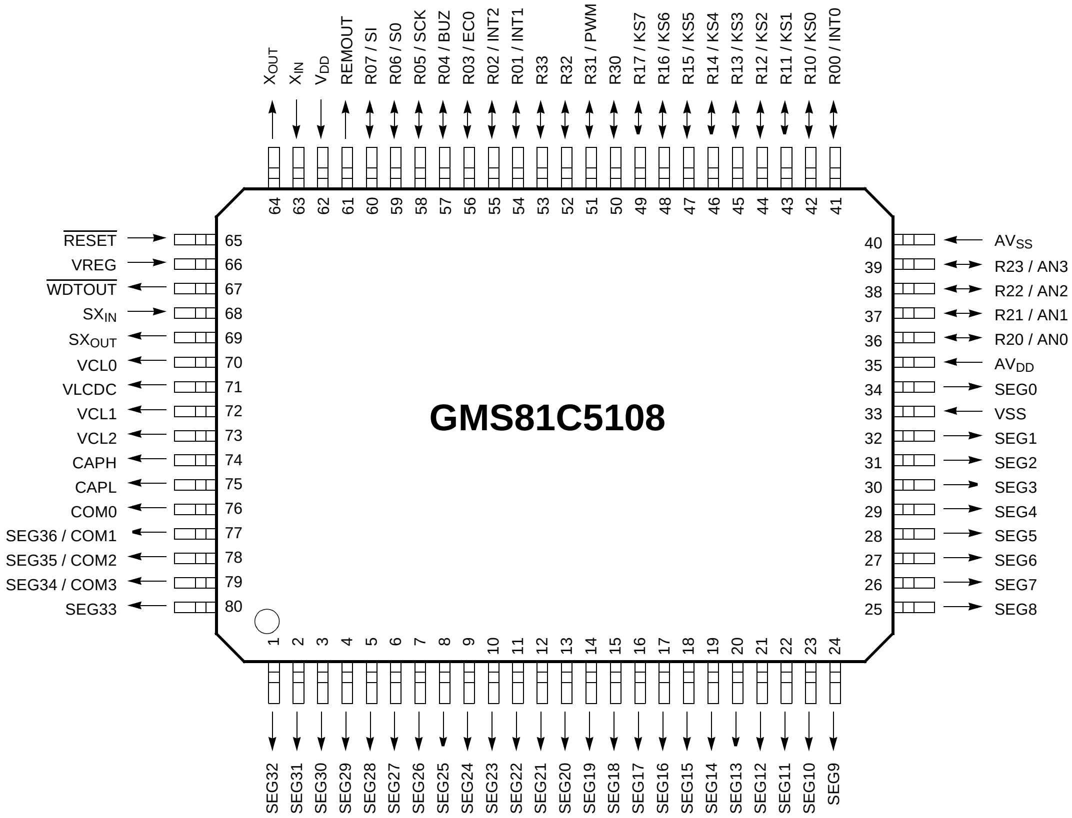
1.4 Ordering Information

	Device name	ROM Size (bytes)	RAM size	Package
Mask ROM version	GMS81C5108	8K bytes	192 bytes	80QFP
OTP ROM version	GMS87C5108	8K bytes OTP	192 bytes	80QFP

2. BLOCK DIAGRAM



3. PIN ASSIGNMENT



4. PACKAGE DIAGRAM

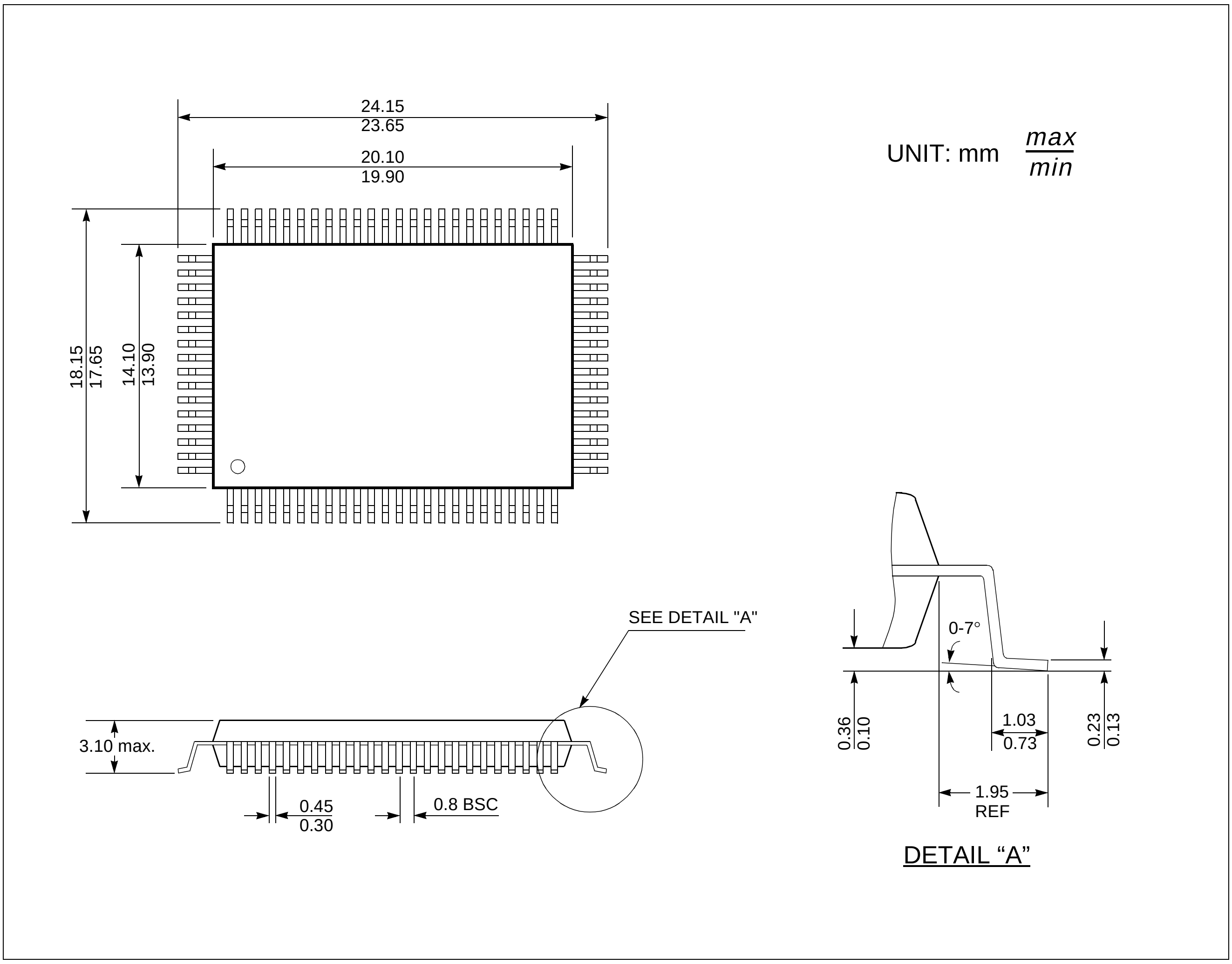


Figure 4-1 Package Diagram

5. PIN FUNCTION

V_{DD}: Supply voltage.

V_{SS}: Circuit ground.

AV_{DD}: Supply voltage to the ladder resistor of ADC circuit. To enhance the resolution of analog to digital converter, use independent power source as well as possible, other than digital power source.

AV_{SS}: ADC circuit ground

$\overline{\text{RESET}}$: Reset the MCU.

$\overline{\text{WDTOUT}}$: Output for detection of a program malfunction. If the user wants to use this pin, connect it to the $\overline{\text{RESET}}$ pin.

REMOUT: Signal output of an infrared remote controller.

X_{IN}: Input to the inverting oscillator amplifier and input to the internal main clock operating circuit.

X_{OUT}: Output from the inverting oscillator amplifier.

SX_{IN}: Input to the internal sub system clock operating circuit.

SX_{OUT}: Output from the inverting subsystem oscillator amplifier.

SEG0~SEG36: Segment signal output pins for the LCD display. See "19. LCD DRIVER" on page 71 for details.

COM0~COM3: Common signal output pins for the LCD display. See "19. LCD DRIVER" on page 71 for details.

SEG34~SEG36 and COM1~COM3 are selected by LCDD of the LCR register.

R00~R07: R0 is an 8-bit CMOS bidirectional I/O port. R0 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. Also, pull-up resistors and open-drain outputs can be assigned by software.

In addition, R0 serves the functions of the various following special features.

Port pin	Alternate function
R00	INT0 (External interrupt 0)
R01	INT1 (External interrupt 1)
R02	INT2 (External interrupt 2)
R03	Event counter input
R04	Buzzer Output
R05	SCK (SPI CLK Input/Output)
R06	SO (SPI Serial Data Output)
R07	SI (SPI Serial Data Input)

R10~R17: R1 is an 8-bit CMOS bidirectional I/O port. R1 pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs or schmitt trigger inputs. Also, pull-up resistors and open-drain outputs can be assigned by software.

In addition, R1 serves the functions of the various following special features.

Port pin	Alternate function
R10	KS0 (Key scan input 0)
R11	KS1 (Key scan input 1)
R12	KS2 (Key scan input 2)
R13	KS3 (Key scan input 3)
R14	KS4 (Key scan input 4)
R15	KS5 (Key scan input 5)
R16	KS6 (Key scan input 6)
R17	KS7 (Key scan input 7)

R20~R23: R2 is a 4-bit CMOS bidirectional I/O port. Each pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. Also, pull-up resistors and open-drain outputs can be assigned by software.

In addition, R2 serves the functions of the various following special features.

Port pin	Alternate function
R20	AN0 (Analog Input Port0)
R21	AN1 (Analog Input Port1)
R22	AN2 (Analog Input Port2)
R23	AN3 (Analog Input Port3)

R30~R33: R3 is a 4-bit CMOS bidirectional I/O port. Each pins 1 or 0 written to the Port Direction Register can be used as outputs or inputs. Also, pull-up resistors and open-drain outputs can be assigned by software.

In addition, R3 serves the functions of the various following special features.

Port pin	Alternate function
R31	PWM (PWM Output)

VCL0~VCL2: Power supply pins for the LCD driver. The voltage on each pin is VCL2> VCL1> VCL0. See "19. LCD DRIVER" on page 71 for details.

VLCDC: LCD drive voltage booster reference.

CAPH, CAPL: LCD drive voltage booster capacitor.

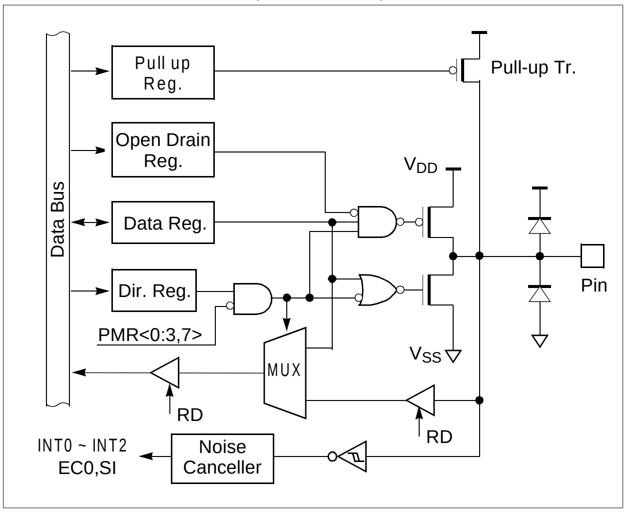
VREG: Output of the voltage regular for the sub clock oscillation circuit. Connect external 0.1uF capacitor to this pin when using the sub system clock.

<table>
<tr><th rowspan="2">PIN NAME</th><th rowspan="2">Pin No.</th><th colspan="2">Primary Function</th><th colspan="2">Secondary Function</th><th rowspan="2">State @ Reset</th><th rowspan="2">State @ STOP</th></tr>
<tr><th>I/O</th><th>Description</th><th>I/O</th><th>Description</th></tr>
<tr><td>V_{DD}</td><td>62</td><td>-</td><td>Supply Voltage</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>V_{SS}</td><td>33</td><td>-</td><td>Circuit Ground</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>AV_{DD}</td><td>35</td><td>-</td><td>Supply Voltage for ADC</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>AV_{SS}</td><td>40</td><td>-</td><td>Ground for ADC</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>$\overline{RESET}$</td><td>65</td><td>I</td><td>Reset (low active)</td><td>-</td><td>-</td><td>‘L’ input</td><td>‘H’ input</td></tr>
<tr><td>$\overline{WDTOUT}$</td><td>67</td><td>O</td><td>Watch dog output</td><td>-</td><td>-</td><td>Floating (To be connect Pull-up)</td><td>State of before STOP</td></tr>
<tr><td>REMOUT</td><td>61</td><td>O</td><td>Remocon output</td><td>-</td><td>-</td><td colspan="2">‘L’ output</td></tr>
<tr><td>X_{IN}, X_{OUT}</td><td>63, 64</td><td>I,O</td><td>Main clock oscillator</td><td>-</td><td>-</td><td>Oscillation</td><td>‘L’, ‘L’</td></tr>
<tr><td>SX_{IN}, SX_{OUT}</td><td>68, 69</td><td>I,O</td><td>Sub clock oscillator</td><td>-</td><td>-</td><td colspan="2">Oscillation</td></tr>
<tr><td>V_{REG}</td><td>66</td><td>-</td><td>Sub clock voltage</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>VCL0~VCL2</td><td>70,72,73</td><td>-</td><td>LCD drive voltage</td><td>-</td><td>-</td><td>Internal VCL0 Connected</td><td>State of before STOP</td></tr>
<tr><td>VLCDC</td><td>71</td><td>-</td><td>LCD drive voltage booster reference</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>CAPH,CAPL</td><td>74,75</td><td>-</td><td>LCD drive voltage booster capacitor</td><td>-</td><td>-</td><td>Internal VCL0 Connected</td><td>State of before STOP</td></tr>
<tr><td>SEG0 ~ SEG33</td><td>34, 32~1</td><td>O</td><td>LCD segment output</td><td>-</td><td>-</td><td colspan="2">Segment output</td></tr>
<tr><td>COM0</td><td>76</td><td>O</td><td>LCD common output</td><td>-</td><td>-</td><td colspan="2">Common output</td></tr>
<tr><td>SEG34/COM3
SEG35/COM2
SEG36/COM1</td><td>79~77</td><td>O</td><td>LCD common output.</td><td>-</td><td>LCD segment output</td><td>Common output</td><td rowspan="14">State of before STOP</td></tr>
<tr><td>R00/INT0</td><td>41</td><td>I/O</td><td rowspan="13">General I/O port</td><td>I</td><td>Interrupt Input</td><td rowspan="13">Input port</td></tr>
<tr><td>R01/INT1</td><td>54</td><td>I/O</td><td>I</td><td>Interrupt Input</td></tr>
<tr><td>R02/INT2</td><td>55</td><td>I/O</td><td>I</td><td>Interrupt Input</td></tr>
<tr><td>R03/EC0</td><td>56</td><td>I/O</td><td>I</td><td>Event counter input</td></tr>
<tr><td>R04/BUZ</td><td>57</td><td>I/O</td><td>O</td><td>Buzzer output</td></tr>
<tr><td>R05/SCK</td><td>58</td><td>I/O</td><td>I/O</td><td>Serial clock I/O</td></tr>
<tr><td>R06/SO</td><td>59</td><td>I/O</td><td>O</td><td>Serial Data Output</td></tr>
<tr><td>R07/SI</td><td>60</td><td>I/O</td><td>I</td><td>Serial Data Input</td></tr>
<tr><td>R10 ~ R17/
KS0 ~ KS7</td><td>42~49</td><td>I/O</td><td>I</td><td>Key wake-up input</td></tr>
<tr><td>R20 ~ R23/
AN0 ~ AN3</td><td>36~39</td><td>I/O</td><td>I</td><td>A/D converter analog input</td></tr>
<tr><td>R30,R32,R33</td><td>50,52,53</td><td>I/O</td><td>-</td><td>-</td></tr>
<tr><td>R31/PWM</td><td>51</td><td>I/O</td><td>O</td><td>PWM output</td></tr>
</table>

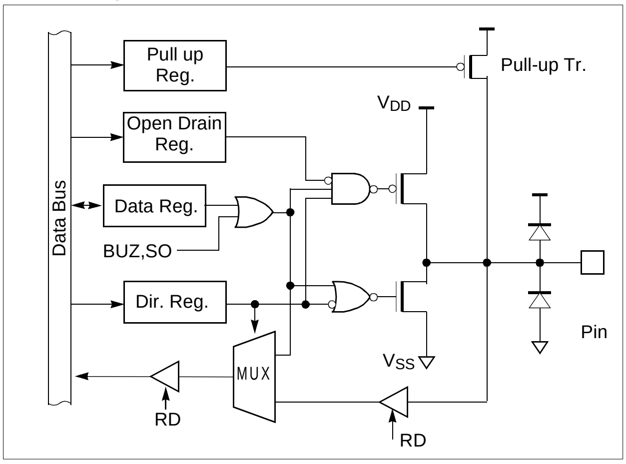
Table 5-1 Port Function Description

6. PORT STRUCTURES

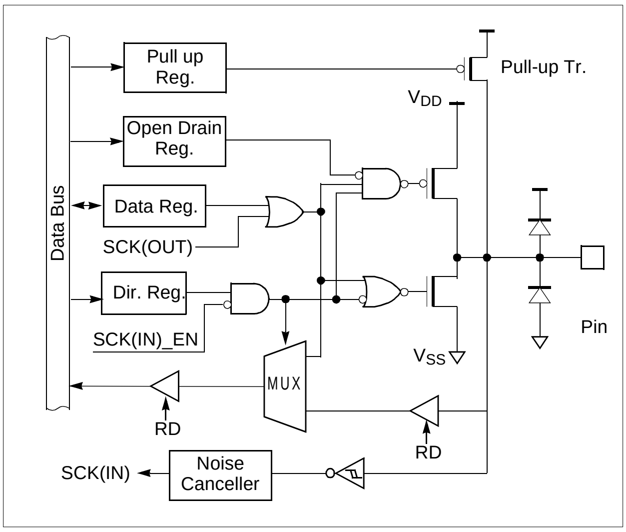
R00~R03/INT0~INT2, R03/EC0, R07/SI



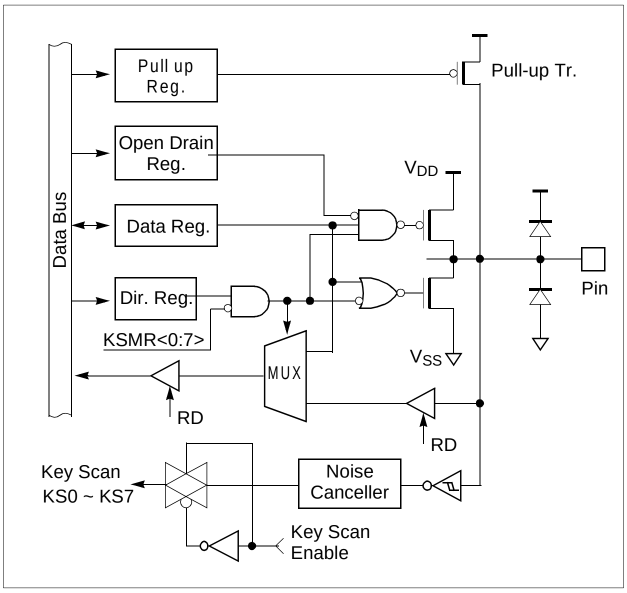
R04/BUZ, R06/SO



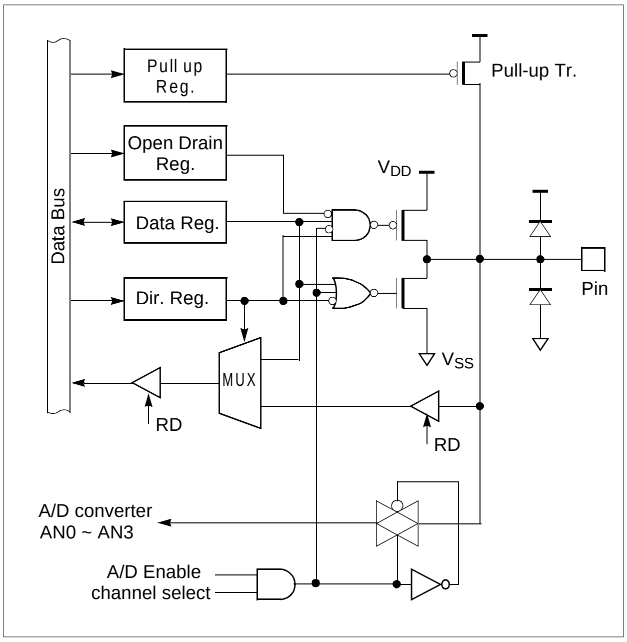
R05/SCK



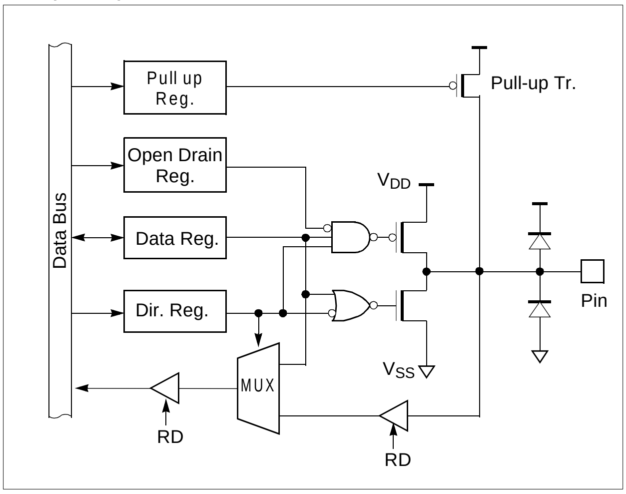
R10~R17/KS0~KS7



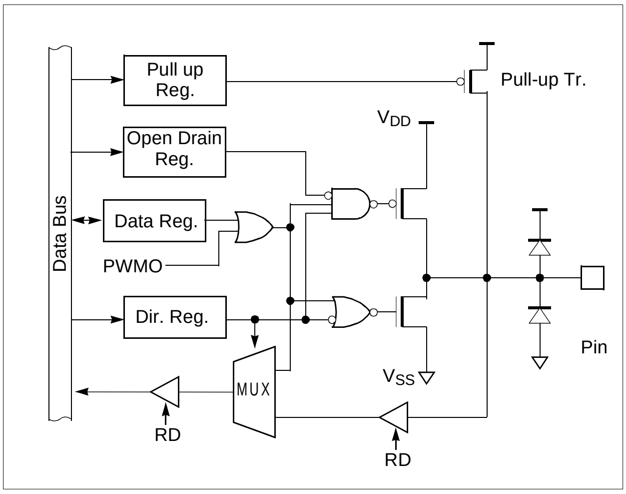
R20~R23/AN0~AN3



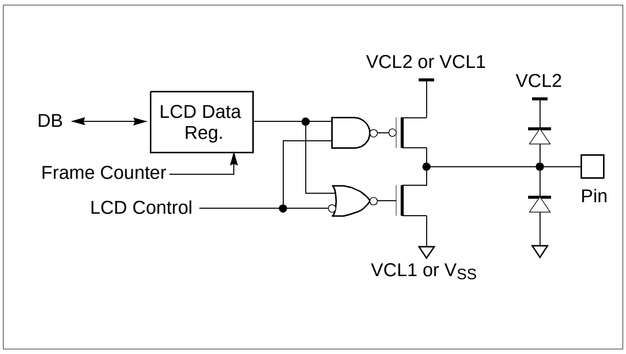
R30, R32, R33



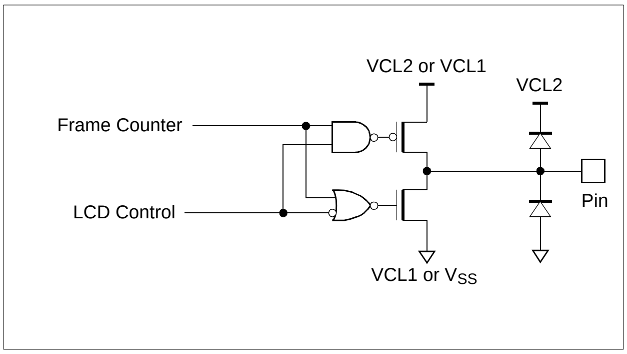
R31



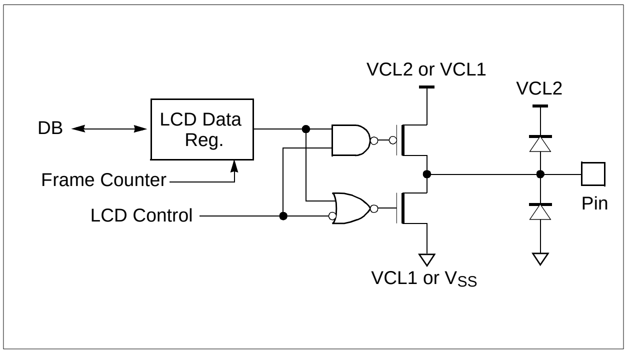
SEG0 ~ SEG33



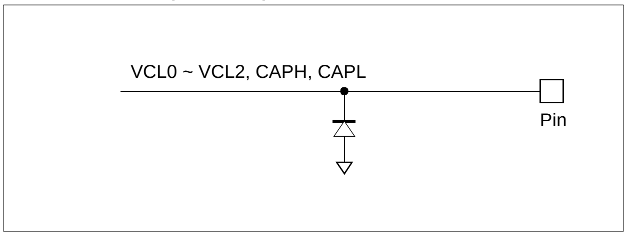
COM0



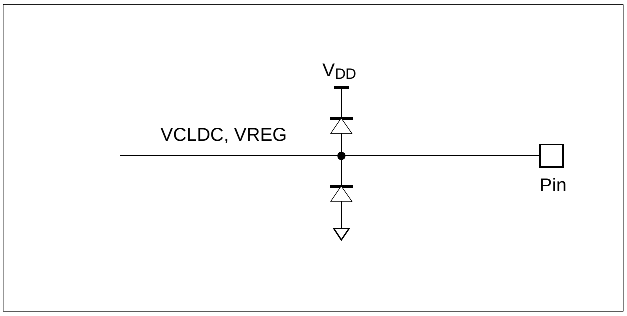
COM1/SEG36, COM2/SEG35, COM3/SEG34



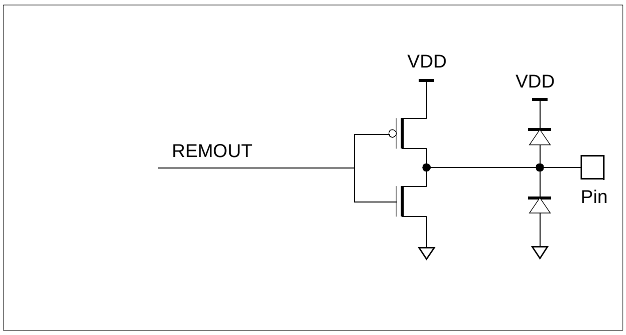
VCL0 ~ VCL2, CAPH, CAPL



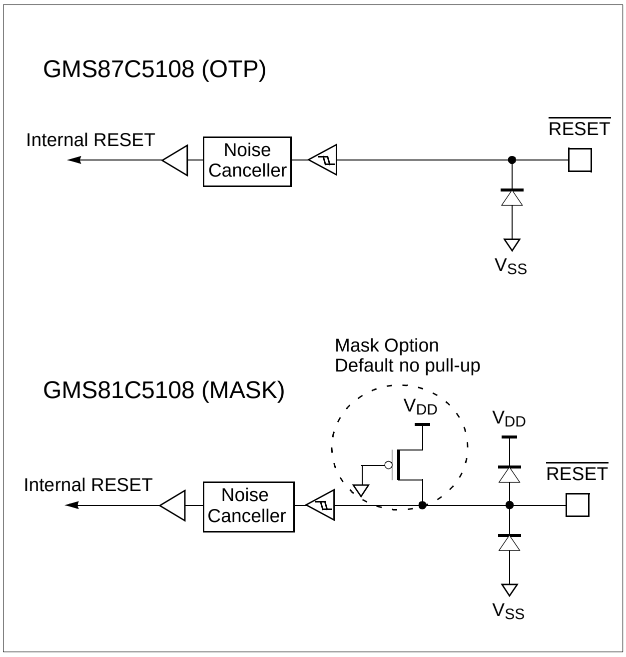
VLCDC, VREG



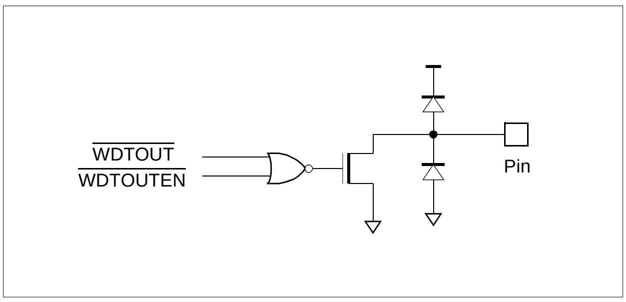
REMOUT



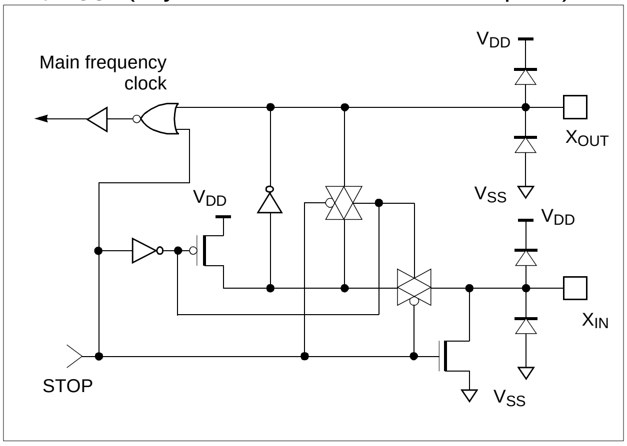
$\overline{\text{RESET}}$



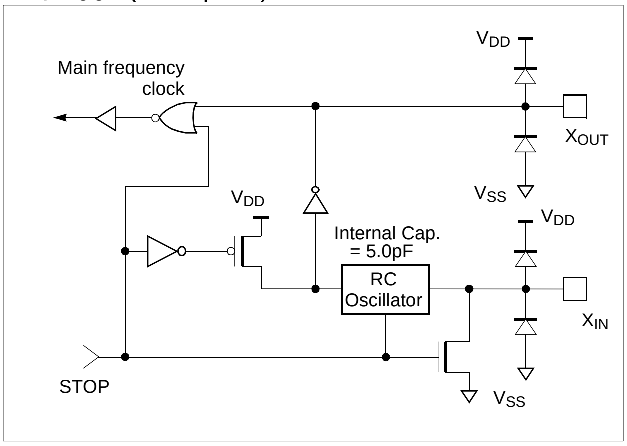
$\overline{\text{WDTOUT}}$



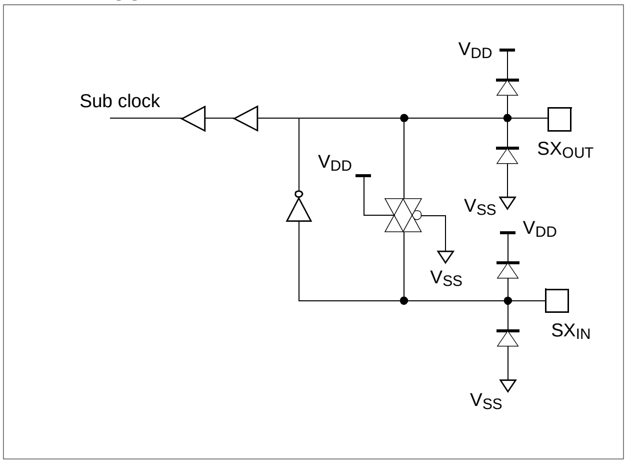
X_{IN}, X_{OUT} (Crystal or Ceramic resonator Option)



X_{IN}, X_{OUT} (RC Option)



SX_{IN}, SX_{OUT}



7. ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Ratings

Supply voltage .... -0.3 to +7.0 V

Storage Temperature ....-40 to +125 °C

Voltage on any pin with respect to Ground (V_{SS}) .... -0.3 to V_{DD}+0.3

Maximum current sunk by (I_{OL} per I/O Pin) ....20 mA

Maximum output current sourced by (I_{OH} per I/O Pin) ....15 mA

Maximum current (ΣI_{OL}) ....100 mA

Maximum current (ΣI_{OH})......................................60 mA

Note: *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

7.2 Recommended Operating Conditions

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Supply Voltage	V_{DD}	f_{MAIN}=4MHz f_{SUB}=32.768kHz	2.0	-	4.0	V
Main Operating Frequency	f_{MAIN}	V_{DD}=2~4V	0.4	-	4.2	MHz
Sub Operating Frequency	f_{SUB}	V_{DD}=2~4V	-	32.768	-	kHz
Operating Temperature	T_{OPR}		-20	-	70	°C

7.3 DC Electrical Characteristics

(TA=-20~70°C, V_{DD}=AV_{DD}=2~4V, V_{SS}=AV_{SS}=0V)

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Input High Voltage	V_{IH1}	R0~R3	0.7V_{DD}	-	V_{DD}	V
	V_{IH2}	$\overline{RESET}$, X_{IN}, INT0~INT2, EC0, SI, SCK	0.8V_{DD}	-	V_{DD}	
	V_{IH3}	SX_{IN}	0.8VREG	-	VREG	
Input Low Voltage	V_{IL1}	R0~R3	0	-	0.3 V_{DD}	V
	V_{IL2}	$\overline{RESET}$, X_{IN}, INT0~INT2, EC0, SI, SCK	0	-	0.2V_{DD}	
	V_{IL3}	SX_{IN}	0	-	0.2VREG	
Output High Voltage	V_{OH1}	R0~R3, I_{OH1}=-0.7mA	V_{DD}-0.3	-	-	V
	V_{OH2}	X_{OUT}, I_{OH2}=-50μA	V_{DD}-0.5	-	-	
	V_{OH3}	SX_{OUT}, I_{OH3}=-5μA	VREG-0.3	-	-	
Output Low Voltage	V_{OL1}	R0~R3, $\overline{WDTOUT}$, I_{OL1}=1mA	-	-	0.4	V
	V_{OL2}	X_{OUT}, I_{OL2}=50μA	-	-	0.5	
	V_{OL3}	SX_{OUT}, I_{OL3}=5μA	-	-	0.5	
Input High Leakage Current	I_{IH}	R0~R3, V_{IN}=V_{DD}	-	-	1	μA
Input Low Leakage Current	I_{IL}	R0~R3, V_{IN}=0V	-	-	-1	
Output High Leakage Current	I_{OH}	REMOUT, V_{DD}=3V, V_{OH}= V_{DD}-1.0V	-30	-	-5	mA
Output Low Leakage Current	I_{OL}	REMOUT, V_{DD}=3V, V_{OL}= 1.0V	0.5	-	3	
Pull-up Resister	R_{P1}	R0~R3, V_{DD}=3V	50	100	200	kΩ
	R_{P2}	$\overline{RESET}$, V_{DD}=3V (GMS81C5108 Mask Option)	30	60	120	
Feed Back Resister	R_{F1}	Main OSC Feedback Resister V_{DD}=3V	0.5	-	1.5	MΩ
	R_{F2}	Sub OSC Feedback Resister V_{DD}=3V	5.	-	15	
RC Oscillator Frequency	F_{RC}	R=30kΩ, V_{DD}=3V	1	2	3	MHz
VREG Voltage	VREG	VREG=0.2uF	2.0	2.2	2.4	V
Supply Current	I_{DD1}	**Main Active Mode** V_{DD}=4V±10%, X_{IN}=4MHz, SX_{IN}=0	-	2.7	4.0	mA
	I_{DD2}	**Main Sleep Mode** V_{DD}=4V±10%, X_{IN}=4MHz, SX_{IN}=0	-	0.47	1.2	
	I_{DD3}	**Stop Mode** V_{DD}=4V±10%, X_{IN}=0, SX_{IN}=0	-	2.0	10	μA
	I_{DD4}	**Sub Active mode[1]** V_{DD}=3V±10%, X_{IN}=0, S_{XIN}=32.768kHz	-	35(70)	80(150)	
	I_{DD5}	**Sub Sleep mode** V_{DD}=4V±10%, X_{IN}=0, S_{XIN}=32.768kHz	-	6.0	15	

1. I_{DD4} is tested by only nop operation. The value of ( ) is tested at OTP.

7.4 LCD Characteristics

(TA=-20~70°C, V_{DD}=AV_{DD}=2~4V, V_{SS}=AV_{SS}=0V)

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
VLCDC Output Voltage	VLCDC	V_{DD}=3V, TA=25°C, R1=1MΩ, R2=300kΩ	0.7	0.9	1.1	V
LCD Reference Output Voltage	VCL0	External Variable Resistance (0 to 1MΩ)	0.9	-	2.0	
Double Output Voltage	VCL1	C1~C4=0.47uF	1.9VCL0	2.0VCL0	-	V
Triple Output Voltage	VCL2	C1~C4=0.47uF	2.85VCL0	3.0VCL0	-	
LCD Common Output Current	I_{COM}	Output Voltage Deviation=0.2V	30	-	-	μA
LCD Segment Output Current	I_{SEG}	Output Voltage Deviation=0.2V	5	-	–	

7.5 A/D Converter Characteristics

(TA=25°C, V_{DD}=3V, AV_{DD}=3.072V, V_{SS}=AV_{SS}=0V)

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Analog Power Supply Input Voltage Range	AV_{DD}	-	AV_{SS}	-	AV_{DD}	V
Analog Input Voltage Range	V_{AN}	-	AV_{SS}-0.3	-	AV_{DD}+0.3	
Current Following Between AV_{DD} and AV_{SS}	IAV_{DD}	-	-	-	200	μA
Overall Accuracy	CAIN	-	-	±1.0	±2.0	LSB
Non Linearity Error	NNLE	-	-	±1.0	±2.0	
Differential Non Linearity Error	NDNLE	-	-	±1.0	±2.0	
Zero Offset Error	NZOE	-	-	±0.5	±1.5	
Full Scale Error	NFSE	-	-	±0.25	±0.5	
Gain Error	NGE	-	-	±1.0	±1.5	
Conversion Time	TCONV	f_{MAIN}=4MHz	-	-	30	μS

7.6 AC Characteristics

(TA=25°C, V_{DD}=4V, AV_{DD}=4V, V_{SS}=AV_{SS}=0V)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
Main Operating Frequency	f_{MCP}	X_{IN}	0.455	-	4.19	MHz
Sub Operating Frequency	f_{SCP}	SX_{IN}	30	32.768	35	kHz
System Clock Frequency[1]	t_{SYS}	-	0.477	-	4.395	µS
Main Oscillation Stabilization Time (4MHz)	t_{MST}	X_{IN}, X_{OUT}	-	-	20	mS
Main Oscillation Stabilization Time (910kHz)			-	-	60	
Main Oscillation Stabilization Time (455kHz)			-	-	100	
Sub Oscillation Stabilization Time	t_{SST}	SX_{IN}, SX_{OUT}	-	1	2	S
External Clock "H" or "L" Pulse Width	t_{MCPW}	X_{IN}	80	-	-	nS
	t_{SCPW}	SX_{IN}	5	-	-	µS
Interrupt Pulse Width	t_{IW}	INT0, INT1, INT2	2	-	-	t_{SYS}
$\overline{\text{RESET}}$ Input Pulse "L" Width	t_{RST}	$\overline{\text{RESET}}$	8	-	-	t_{SYS}
Event Counter Input "H" or "L" Pulse Width	t_{ECW}	EC0	2	-	-	t_{SYS}

1.SCMR=XXXX000X that is $f_{MAIN}/2$

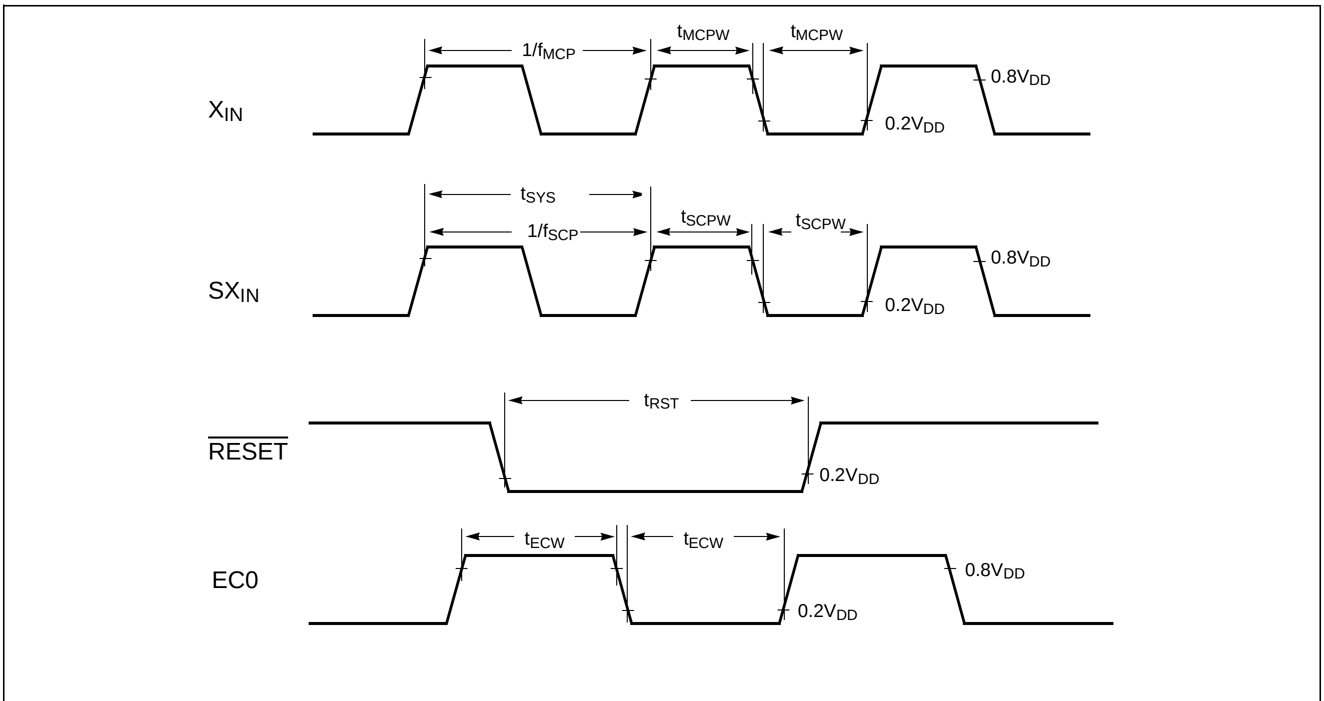


Figure 7-1 AC Timing Chart

7.7 Serial I/O Characteristics

(TA=25°C, V_{DD}=AV_{DD}=2~4V, V_{SS}=AV_{SS}=0V)

Parameter	Symbol	Pins	Specifications			Unit
			Min.	Typ.	Max.	
SCK Input Clock Pulse Period	t_{SCYC}	SCK	$2t_{SYS}$+200	-	-	nS
SCK Input Clock “H” or “L” Pulse Width	t_{SCKW}		t_{SYS}+70	-	-	
SCK Output Clock Cycle Time	t_{SCYC}		$4t_{SYS}$	-	$16t_{SYS}$	
SCK output Clock “H” or “L” Pulse Width	t_{SCKW}		$2t_{SYS}$-30	-	-	
SCK output Clock Delay Time	t_{DS}		-	-	100	
SI input Setup Time (External SCK)	t_{ESUS}	SI	100	-	-	
SI input Setup Time (Internal SCK)	t_{ISUS}		100	-	-	
SI input Hold Time	t_{HS}		t_{SYS}+100	-	-	

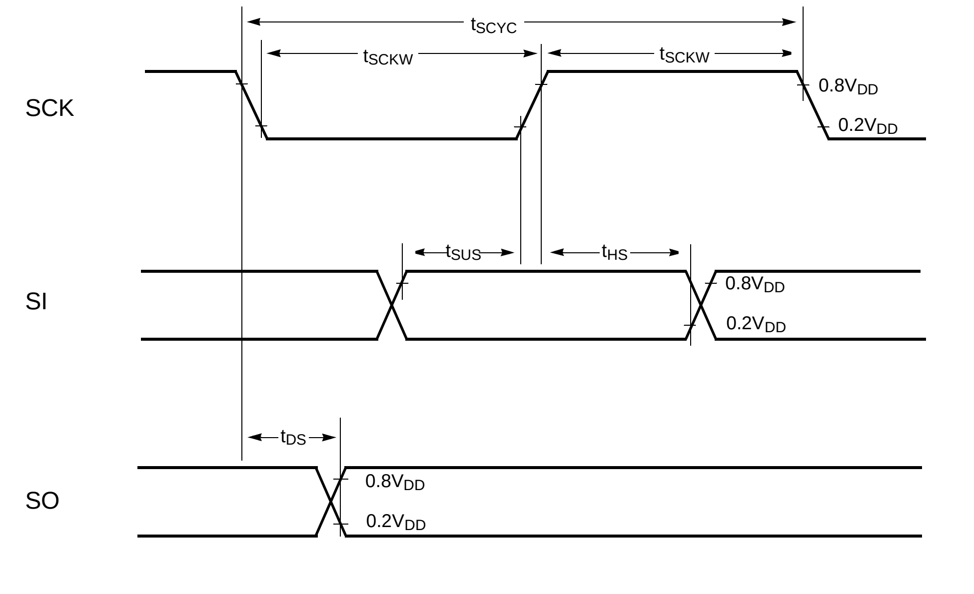


Figure 7-2 Serial I/O Timing Chart

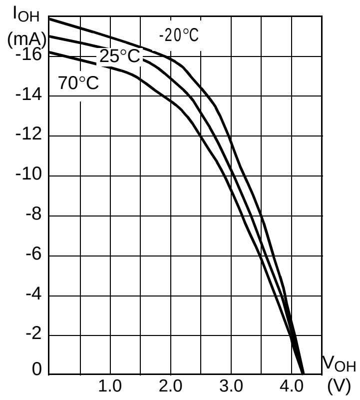
7.8 Typical Characteristics

These graphs and tables are for design guidance only and are not tested or guaranteed.

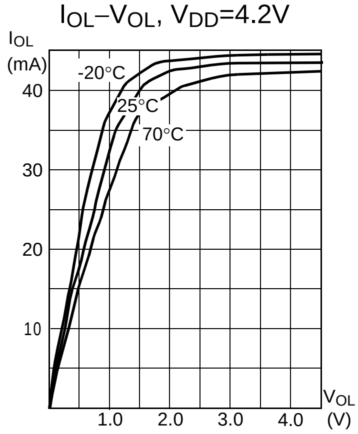
In some graphs or tables, the datas presented are outside specified operating range (e.g. outside specified V_{DD} range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean – 3σ) respectively where σ is standard deviation

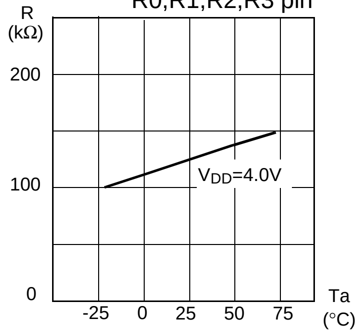
I_{OH}–V_{OH}, V_{DD}=4.2V



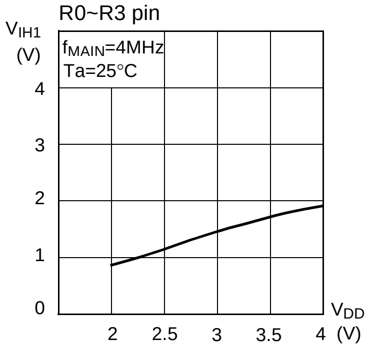
I_{OL}–V_{OL}, V_{DD}=4.2V



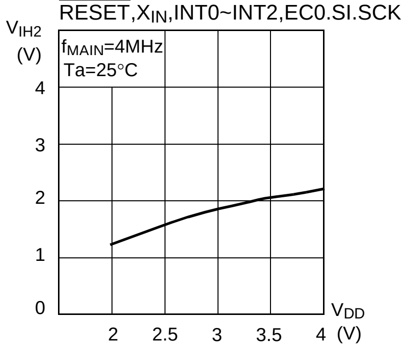
R–Ta R0,R1,R2,R3 pin



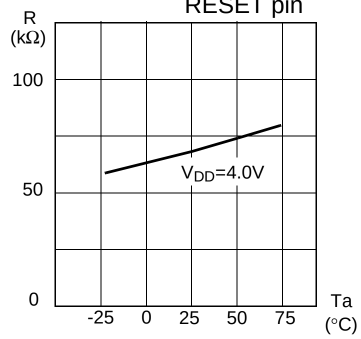
V_{DD}–V_{IH1}
R0~R3 pin



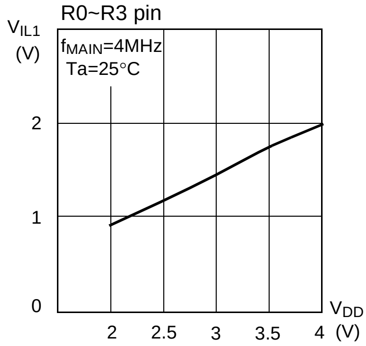
V_{DD}–V_{IH2}
$\overline{RESET}$,X_{IN},INT0~INT2,EC0.SI.SCK



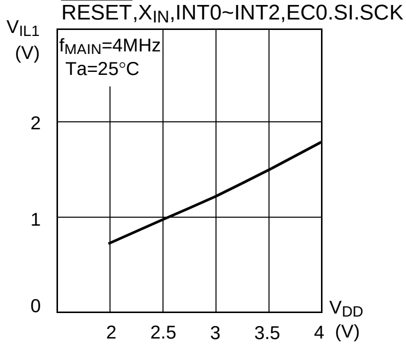
R–Ta $\overline{RESET}$ pin



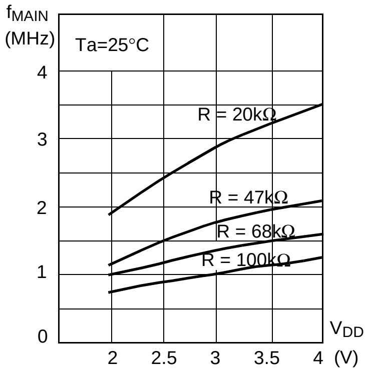
V_{DD}–V_{IL1}
R0~R3 pin



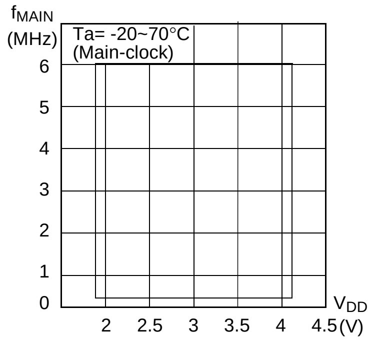
V_{DD}–V_{IL2}
$\overline{RESET}$,X_{IN},INT0~INT2,EC0.SI.SCK



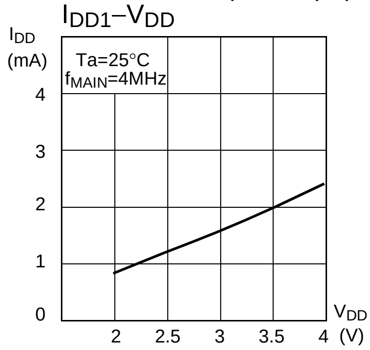
f_{MAIN}–V_{DD}



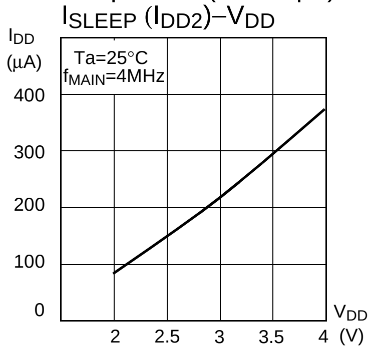
Operating Area



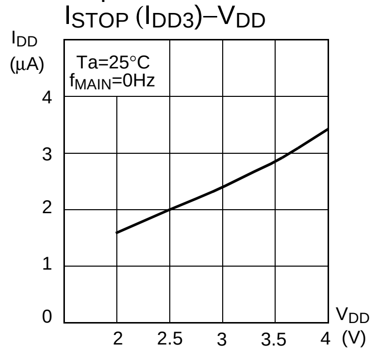
Normal Mode (Main opr.)
I_{DD1}–V_{DD}



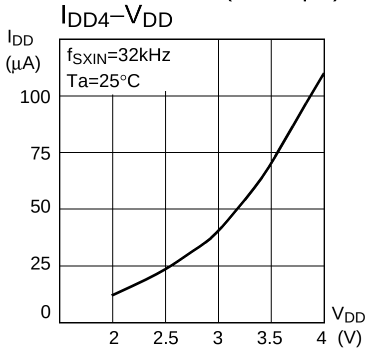
Sleep Mode (Main opr.)
I_{SLEEP} (I_{DD2})–V_{DD}



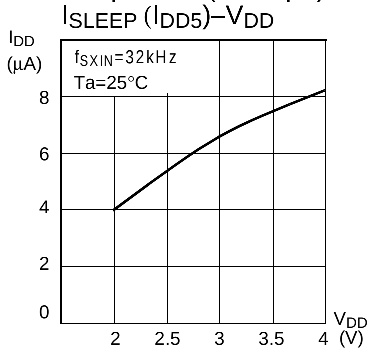
Stop Mode
I_{STOP} (I_{DD3})–V_{DD}



Normal Mode (Sub opr.)
I_{DD4}–V_{DD}



Sleep Mode (Sub opr.)
I_{SLEEP} (I_{DD5})–V_{DD}



8. MEMORY ORGANIZATION

The GMS81C5108 has separate address spaces for Program memory, Data Memory and Display memory. Program memory can only be read, not written to. It can be up to 8K bytes of Program memory. Data memory can be read and written to up to 192 bytes including the stack area. Display memory has prepared 37 bytes for LCD.

8.1 Registers

This device has six registers that are the Program Counter (PC), a Accumulator (A), two index registers (X, Y), the Stack Pointer (SP), and the Program Status Word (PSW). The Program Counter consists of 16-bit register.

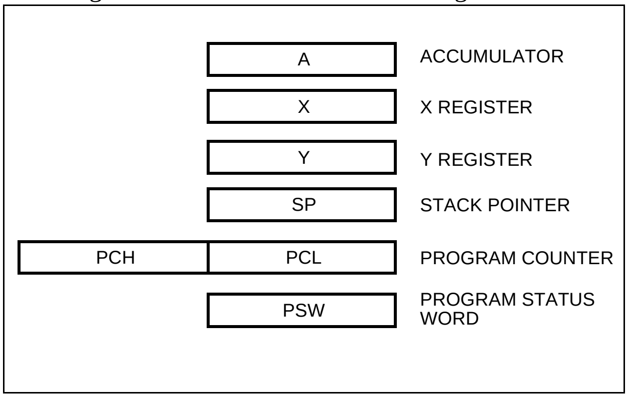


Figure 8-1 Configuration of Registers

Accumulator: The Accumulator is the 8-bit general purpose register, used for data operation such as transfer, temporary saving, and conditional judgement, etc.

The Accumulator can be used as a 16-bit register with Y Register as shown below.

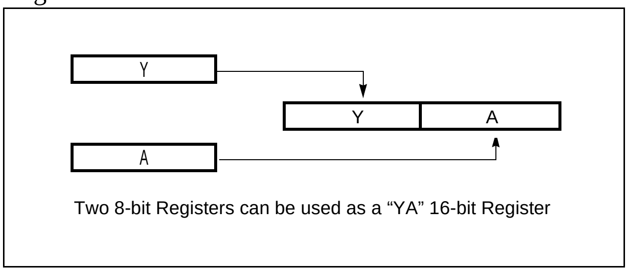


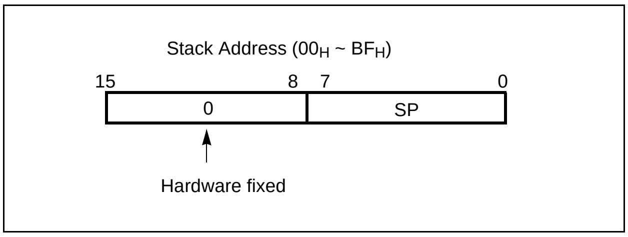
Figure 8-2 Configuration of YA 16-bit Register

X, Y Registers: In the addressing mode which uses these index registers, the register contents are added to the specified address, which becomes the actual address. These modes are extremely effective for referencing subroutine tables and memory tables. The index registers also have increment, decrement, comparison and data transfer functions, and they can be used as simple accumulators.

Stack Pointer: The Stack Pointer is an 8-bit register used for occurrence interrupts and calling out subroutines. Stack Pointer identifies the location in the stack to be accessed (save or restore).

Generally, SP is automatically updated when a subroutine call is executed or an interrupt is accepted. However, if it is used in excess of the stack area permitted by the data memory allocating configuration, the user-processed data may be lost.

The stack can be located at any position within 00_H to BF_H of the internal data memory. The SP is not initialized by hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of “BF_H” is used.



Caution:

The Stack Pointer must be initialized by software because its value is undefined after RESET.

Example: To initialize the SP

```
LDX    #0BFH
TXSP                  ; SP ← BFH
```

Program Counter: The Program Counter is a 16-bit wide which consists of two 8-bit registers, PCH and PCL. This counter indicates the address of the next instruction to be executed. In reset state, the program counter has reset routine address (PC_H:$0FF_H$, PC_L:$0FE_H$).

Program Status Word: The Program Status Word (PSW) contains several bits that reflect the current state of the CPU. The PSW is described in Figure 8-3. It contains the Negative flag, the Overflow flag, the Break flag the Half Carry (for BCD operation), the Interrupt enable flag, the Zero flag, and the Carry flag.

[Carry flag C]

This flag stores any carry or borrow from the ALU of CPU after an arithmetic operation and is also changed by the Shift Instruction or Rotate Instruction.

[Zero flag Z]

This flag is set when the result of an arithmetic operation or data transfer is “0” and is cleared by any other result.

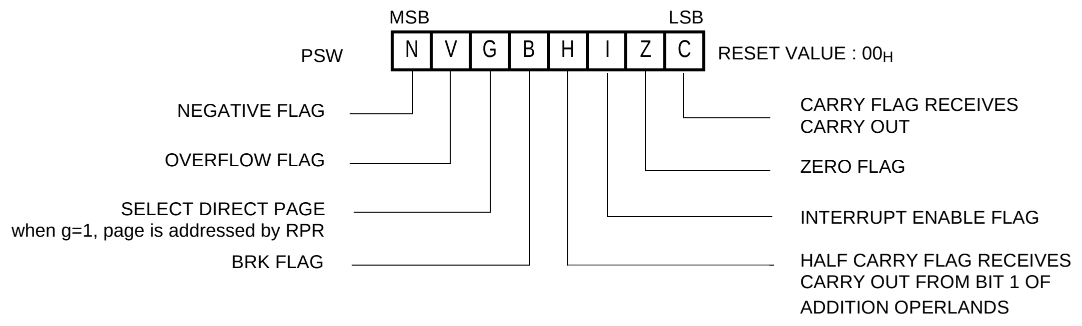


Figure 8-3 PSW (Program Status Word) Register

[Interrupt disable flag I]

This flag enables/disables all interrupts except interrupt caused by Reset or software BRK instruction. All interrupts are disabled when cleared to “0”. This flag immediately becomes “0” when an interrupt is served. It is set by the EI instruction and cleared by the DI instruction.

[Half carry flag H]

After operation, this is set when there is a carry from bit 3 of ALU or there is no borrow from bit 4 of ALU. This bit can not be set or cleared except CLRV instruction with Overflow flag (V).

[Break flag B]

This flag is set by software BRK instruction to distinguish BRK from TCALL instruction with the same vector address.

[Direct page flag G]

This flag assigns RAM page for direct addressing mode. In the direct addressing mode, addressing area is from zero page 00_H to $0FF_H$ when this flag is "0". If it is set to "1", addressing area is assigned by RPR register (address $0F3_H$). It is set by SETG instruction and cleared by CLRG.

[Overflow flag V]

This flag is set to “1” when an overflow occurs as the result of an arithmetic operation involving signs. An overflow occurs when the result of an addition or subtraction exceeds +127 ($7F_H$) or −128 (80_H). The CLRV instruction clears the overflow flag. There is no set instruction. When the BIT instruction is executed, bit 6 of memory is copied to this flag.

[Negative flag N]

This flag is set to match the sign bit (bit 7) status of the result of a data or arithmetic operation. When the BIT instruction is executed, bit 7 of memory is copied to this flag.

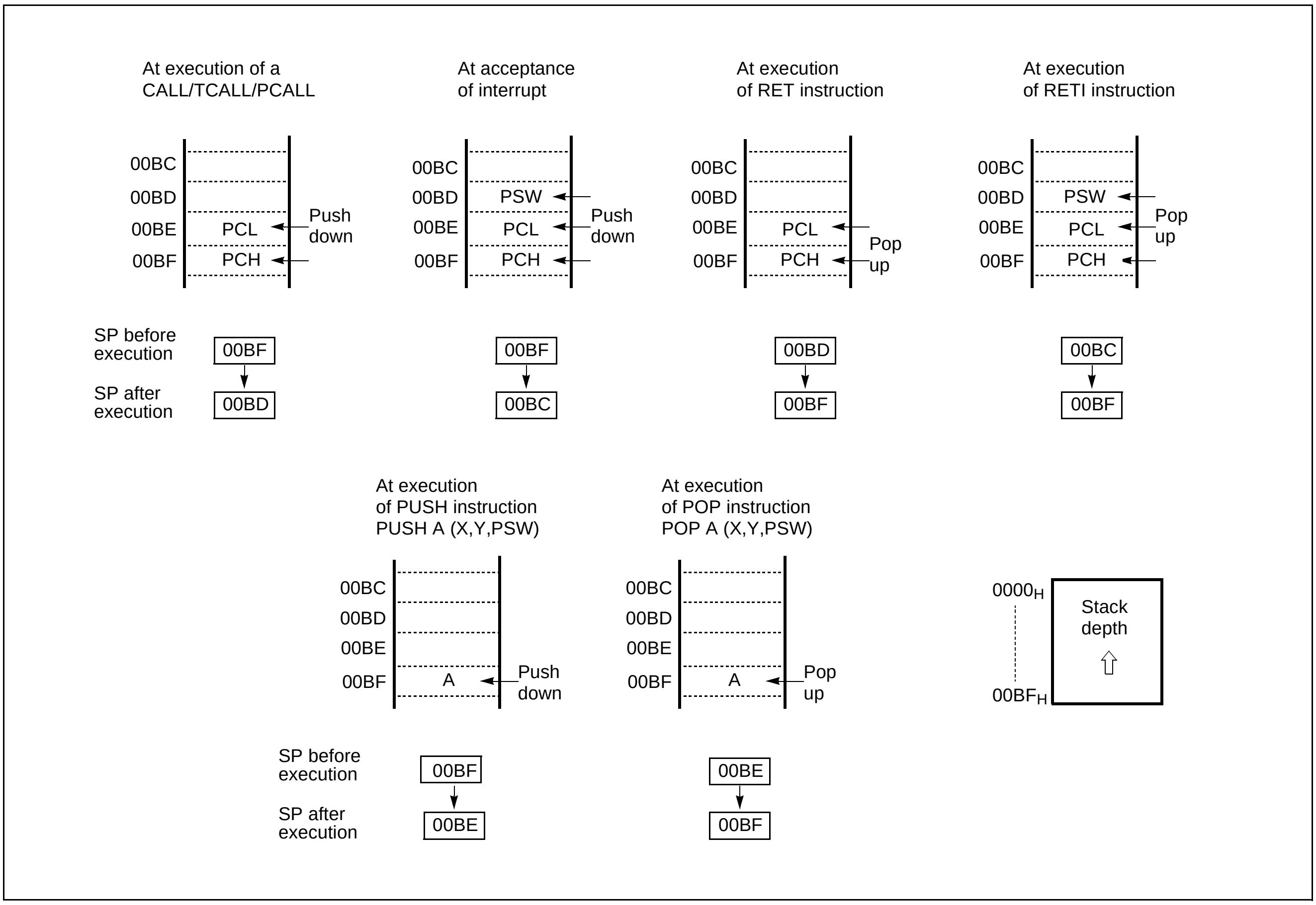


Figure 8-4 Stack Operation

8.2 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but this device has 8K bytes program memory space only physically implemented. Accessing a location above $FFFF_H$ will cause a wrap-around to 0000_H.

Figure 8-5 shows a map of Program Memory. After reset, the CPU begins execution from reset vector which is stored in address $FFFE_H$ and $FFFF_H$ as shown in Figure 8-6.

As shown in Figure 8-5, each area is assigned a fixed location in Program Memory. Program Memory area contains the user program.

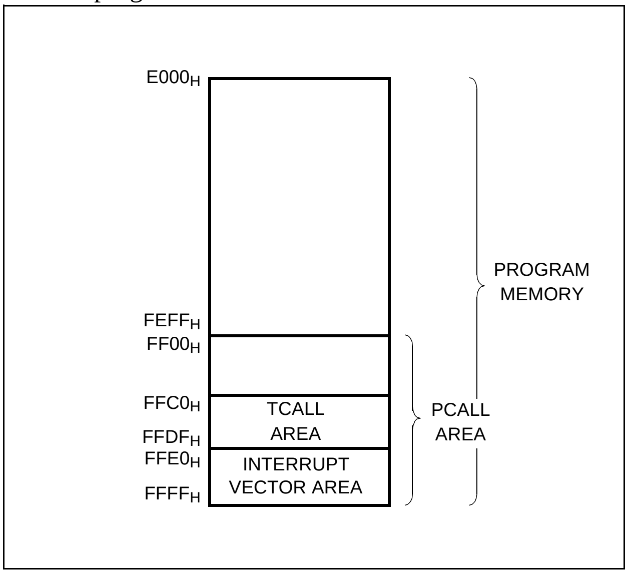


Figure 8-5 Program Memory Map

Page Call (PCALL) area contains subroutine program to reduce program byte length by using 2 bytes PCALL instead of 3 bytes CALL instruction. If it is frequently called, it is more useful to save program byte length.

Table Call (TCALL) causes the CPU to jump to each TCALL address, where it commences the execution of the service routine. The Table Call service area spaces 2-byte for every TCALL: $0FFC0_H$ for TCALL15, $0FFC2_H$ for TCALL14, etc., as shown in Figure 8-7.

Example: Usage of TCALL

```
LDA     #5
        TCALL  0FH          ;1BYTE INSTRUCTION
        :                   ;INSTEAD OF 2 BYTES
        :                   ;NORMAL CALL
;
;TABLE CALL ROUTINE
;
FUNC_A: LDA    LRG0
        RET
;
FUNC_B: LDA    LRG1       ②  ①
        RET
;
;TABLE CALL ADD. AREA
;
        ORG    0FFC0H       ;TCALL ADDRESS AREA
        DW     FUNC_A
        DW     FUNC_B
```

The interrupt causes the CPU to jump to specific location, where it commences the execution of the service routine. The External interrupt 0, for example, is assigned to location $0FFFA_H$. The interrupt service locations spaces 2-byte interval: $0FFF8_H$ and $0FFF9_H$ for External Interrupt 1, $0FFFA_H$ and $0FFFB_H$ for External Interrupt 0, etc.

Any area from $0FF00_H$ to $0FFFF_H$, if it is not going to be used, its service location is available as general purpose Program Memory.

Address	Vector Area Memory
$0FFE0_H$	-
E2	-
E4	-
E6	-
E8	Watch Timer Interrupt Vector Area
EA	Serial I/O Interrupt Vector Area
EC	AD Converter Interrupt Vector Area
EE	Remocon Interrupt Vector Area
F0	External Interrupt 2 Vector Area
F2	Timer/Counter 1 Interrupt Vector Area
F4	Timer/Counter 0 Interrupt Vector Area
F6	External Interrupt 1 Vector Area
F8	External Interrupt 0 Vector Area
FA	Basic Interval Timer Interrupt Vector Area
FC	Key Scan Interrupt Vector Area
FE	RESET Vector Area

NOTE:
"-" means reserved area.

Figure 8-6 Interrupt Vector Area

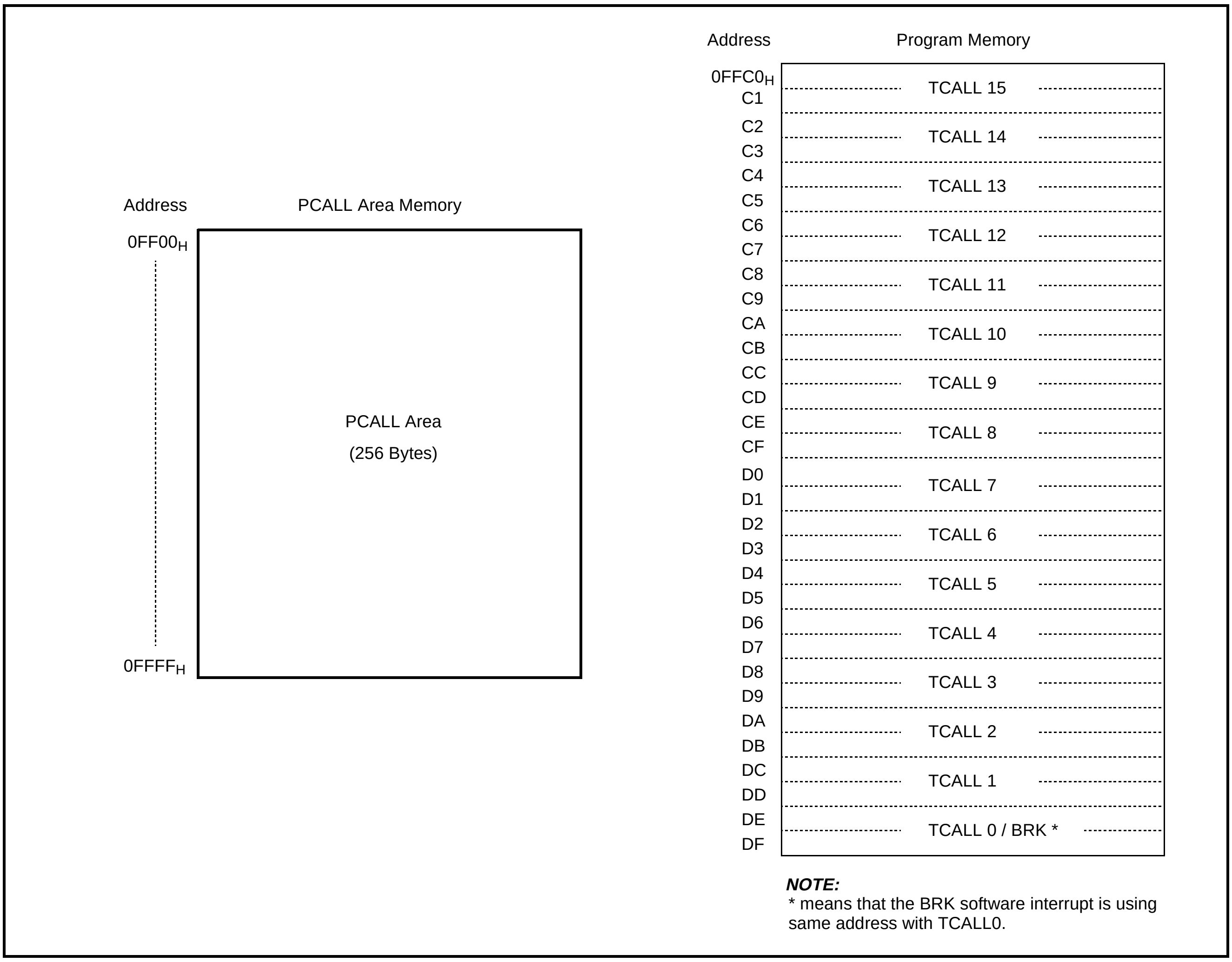
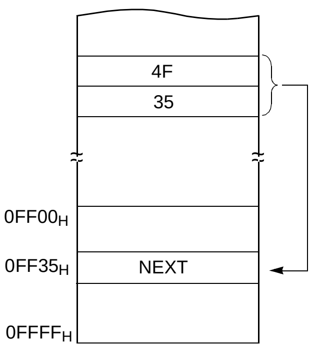


Figure 8-7 PCALL and TCALL Memory Area

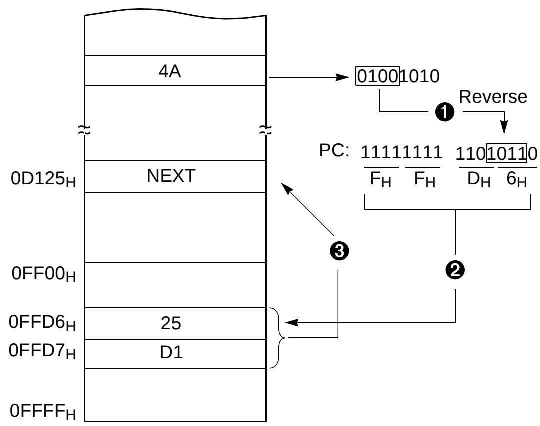
PCALL→ rel

```
4F35      PCALL  35H
```



TCALL→ n

```
4A        TCALL  4
```



Example: The usage software example of Vector address and the initialize part.

```
          ORG     0FFE0H

          DW      NOT_USED
          DW      NOT_USED
          DW      NOT_USED
          DW      NOT_USED
          DW      WT_INT                  ; Watch Timer
          DW      SIO                     ; Serial I/O
          DW      AD_Con                  ; AD converter
          DW      Carrier_INT             ; Carrier
          DW      INT2                    ; Int.2
          DW      TMR1_INT                ; Timer-1
          DW      TMR0_INT                ; Timer-0
          DW      INT1                    ; Int.1
          DW      INT0                    ; Int.0
          DW      BIT_INT                 ; BIT
          DW      KEY_INT                 ; Key Scan
          DW      RESET                   ; Reset

          ORG     0F000H

;*********************************************
;             MAIN     PROGRAM            *
;*********************************************
;
RESET:    DI                              ;Disable All Interrupts
          CLRG
          LDX     #0
RAM_CLR:  LDA     #0                      ;RAM Clear(!0000H->!00BFH)
          STA     {X}+
          CMPX    #0C0H
          BNE     RAM_CLR
;
          LDX     #0BFH                   ;Stack Pointer Initialize
          TXSP
;
          CALL    LCD_CLR                 ;Clear LCD display memory
;
          LDM     R0, #0                  ;Normal Port 0
          LDM     R0DR,#1000_0010B        ;Normal Port Direction
          LDM     R0PU,#1000_0010B        ;Pull Up Selection Set
          LDM     R0CR,#0000_0001B        ;R0 port Open Drain control
           :
           :
          LDM     SCMR,#1111_0000B        ;System clock control
           :
           :
```

8.3 Data Memory

Figure 8-8 shows the internal Data Memory space available. Data Memory is divided into four groups, a user RAM, control registers, Stack, and LCD memory.

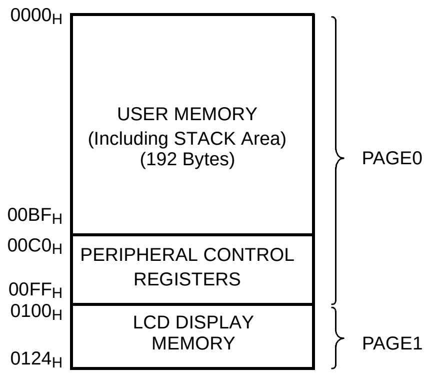


Figure 8-8 Data Memory Map

User Memory

The GMS81C5108 has 192 × 8 bits for the user memory (RAM).

There are two page internal RAM. Page is selected by G-flag and RAM page selection register RPR. When G-flag is cleared to “0”, always page 0 is selected regardless of RPR value. If G-flag is set to “1”, page will be selected according to RPR value.

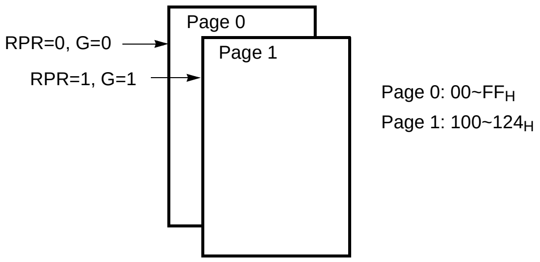


Figure 8-9 RAM page configuration

Control Registers

The control registers are used by the CPU and Peripheral function blocks for controlling the desired operation of the device. Therefore these registers contain control and status bits for the interrupt system, the timer/ counters, analog to digital converters and I/O ports. The control registers are in address range of $0C0_H$ to $0FF_H$.

Note that unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

More detailed informations of each register are explained in each peripheral section.

***Note:** Write only registers can not be accessed by bit manipulation instruction. Do not use read-modify-write instruction. Use byte manipulation instruction.*

Example; To write at CKCTLR

```
LDM    CKCTLR,#05H ;Divide ratio ÷8
```

Stack Area

The stack provides the area where the return address is saved before a jump is performed during the processing routine at the execution of a subroutine call instruction or the acceptance of an interrupt.

When returning from the processing routine, executing the subroutine return instruction [RET] restores the contents of the program counter from the stack; executing the interrupt return instruction [RETI] restores the contents of the program counter and flags.

The save/restore locations in the stack are determined by the stack pointed (SP). The SP is automatically decreased after the saving, and increased before the restoring. This means the value of the SP indicates the stack location number for the next save. Refer to Figure 8-4 on page 20.

LCD Display Memory

LCD display data area is handled in LCD section.

See "19.3 LCD Display Memory" on page 73.

Address	Register Name	Symbol	R/W	Initial Value								Addressing Mode	Page
				7	6	5	4	3	2	1	0		
00C0	R0 port data register	R0	R/W	0	0	0	0	0	0	0	0	byte, bit[1]	32
00C1	R1 port data register	R1	R/W	0	0	0	0	0	0	0	0	byte, bit	32
00C2	R2 port data register	R2	R/W	-	-	-	-	0	0	0	0	byte, bit	33
00C3	R3 port data register	R3	R/W	-	-	-	-	0	0	0	0	byte, bit	33
00C8	R0 port I/O direction register	R0DR	W	0	0	0	0	0	0	0	0	byte[2]	32
00C9	R1 port I/O direction register	R1DR	W	0	0	0	0	0	0	0	0	byte	32
00CA	R2 port I/O direction register	R2DR	W	-	-	-	-	0	0	0	0	byte	33
00CB	R3 port I/O direction register	R3DR	W	-	-	-	-	0	0	0	0	byte	33
00D0	R0 port pull-up register	R0PU	W	0	0	0	0	0	0	0	0	byte	32
00D1	R1 port pull-up register	R1PU	W	0	0	0	0	0	0	0	0	byte	32
00D2	R2 port pull-up register	R2PU	W	-	-	-	-	0	0	0	0	byte	33
00D3	R3 port pull-up register	R3PU	W	-	-	-	-	0	0	0	0	byte	33
00D4	R0 port open drain control register	R0CR	W	0	0	0	0	0	0	0	0	byte	32
00D5	R1 port open drain control register	R1CR	W	0	0	0	0	0	0	0	0	byte	32
00D6	R2 port open drain control register	R2CR	W	-	-	-	-	0	0	0	0	byte	33
00D7	R3 port open drain control register	R3CR	W	-	-	-	-	0	0	0	0	byte	33
00D8	Ext. interrupt edge selection register	IESR	R/W	-	-	0	0	0	0	0	0	byte, bit	69
00D9	Port selection register	PMR	R/W	-	0	-	0	0	0	0	0	byte, bit	32
00DA	Interrupt enable low register	IENL	R/W	-	0	0	0	0	-	-	-	byte, bit	65
00DB	Interrupt enable high register	IENH	R/W	-	0	0	0	0	0	0	0	byte, bit	65
00DC	Interrupt request flag low register	IRQL	R/W	-	0	0	0	0	-	-	-	byte, bit	65
00DD	Interrupt request flag high register	IRQH	R/W	-	0	0	0	0	0	0	0	byte, bit	65
00DE	Sleep mode register	SMR	R/W	-	-	-	-	-	-	-	0	byte, bit	39
00E0	Timer 0 mode register	TM0	R/W	-	-	0	0	0	0	0	0	byte, bit	45
00E1	Timer 0 counter register	T0	R	0	0	0	0	0	0	0	0	byte, bit	45
	Timer 0 data register	TDR0	W	1	1	1	1	1	1	1	1	byte	45
	Timer 0 input capture register	CDR0	R	0	0	0	0	0	0	0	0	byte, bit	45
00E2	Timer 1 mode register	TM1	R/W	0	0	0	0	0	0	0	0	byte, bit	45
00E3	Timer 1 data register	TDR1	W	1	1	1	1	1	1	1	1	byte	45
	PWM0 pulse period register	T1PPR	W	1	1	1	1	1	1	1	1	byte	45
00E4	Timer 1 counter register	T1	R	0	0	0	0	0	0	0	0	byte, bit	45
	Timer 1 input capture register	CDR1	R	0	0	0	0	0	0	0	0	byte, bit	45
	PWM0 pulse duty register	T1PDR	R/W	0	0	0	0	0	0	0	0	byte, bit	45
00E5	PWM0 high register	PWMHR	W	-	-	-	-	0	0	0	0	byte	45
00EC	A/D converter mode register	ADMR	R/W	-	0	-	-	0	0	0	1	byte, bit	58
00ED	A/D converter data register	ADDR	R	x	x	x	x	x	x	x	x	byte, bit	58

Table 8-1 Control Registers

Address	Register Name	Symbol	R/W	Initial Value 7	6	5	4	3	2	1	0	Addressing Mode	Page
00EF	Watch timer mode register	WTMR	R/W	-	0	0	0	0	0	0	0	byte, bit	56
00F0	Key scan mode register	KSMR	R/W	0	0	0	0	0	0	0	0	byte, bit	70
00F1	LCD control register	LCR	R/W	0	0	0	0	0	0	0	0	byte, bit	72
00F3	RAM paging register	RPR	R/W	-	-	-	-	-	-	0	0	byte, bit	73
00F4	Basic interval timer register	BITR	R	0	0	0	0	0	0	0	0	byte, bit	43
	Clock control register	CKCTLR	W	-	-	-	-	0	1	1	1	byte	43
00F5	System clock mode register	SCMR	R/W	0	0	0	0	0	0	0	0	byte, bit	34
00F6	Remocon mode register	RMR	R/W	-	0	0	0	0	0	0	0	byte, bit	76
00F7	Carrier frequency high selection	CFHS	W	-	-	1	1	1	1	1	1	byte	76
00F8	Carrier frequency low selection	CFLS	W	-	-	1	1	1	1	1	1	byte	76
00F9	Remocon data high register	RDHR	W	1	1	1	1	1	1	1	1	byte	76
00FA	Remocon data low register	RDLR	W	1	1	1	1	1	1	1	1	byte	76
	Remocon data counter	RDC	R	0	0	0	0	0	0	0	0	byte, bit	76
00FB	Remocon output data register	RODR	R/W	-	-	-	-	-	-	-	0	byte, bit	76
00FC	Remocon output buffer	ROB	R/W	-	-	-	-	-	-	-	0	byte, bit	76
00FD	Buzzer data register	BDR	W	0	0	0	0	0	0	0	0	byte	60
00FE	Serial I/O mode register	SIOM	R/W	0	0	0	0	0	0	0	1	byte, bit	62
00FF	Serial I/O data register	SIOD	R/W	x	x	x	x	x	x	x	x	byte, bit	62

Table 8-1 Control Registers

1. “byte”, “bit” means that register can be addressed by not only bit but byte manipulation instruction.
2. “byte” means that register can be addressed by only byte manipulation instruction. On the other hand, do not use any read-modify-write instruction such as bit manipulation.

8.4 Addressing Mode

The GMS81C5108 uses six addressing modes;

- **Register addressing**
- **Immediate addressing**
- **Direct page addressing**
- **Absolute addressing**
- **Indexed addressing**
- **Register-indirect addressing**

(1) Register Addressing

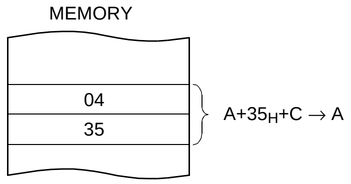
Register addressing accesses the A, X, Y, C and PSW.

(2) Immediate Addressing → #imm

In this mode, second byte (operand) is accessed as a data immediately.

Example:

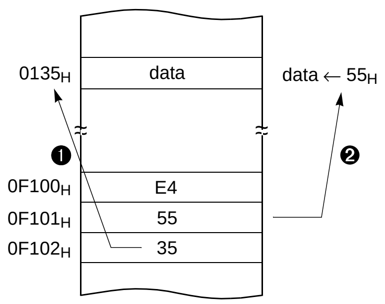
0435 ADC #35_H



When G-flag is 1, then RAM address is defined by 16-bit address which is composed of 8-bit RAM paging register (RPR) and 8-bit immediate data.

Example: G=1, RPR=01_H

E45535 LDM 35_H,#55_H

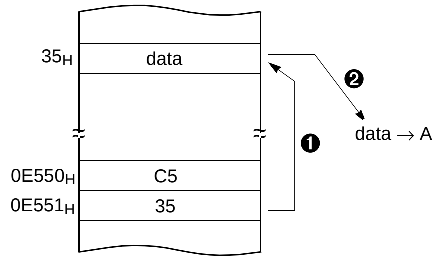


(3) Direct Page Addressing → dp

In this mode, a address is specified within direct page.

Example; G=0

C535 LDA 35_H ;A ←RAM[35_H]



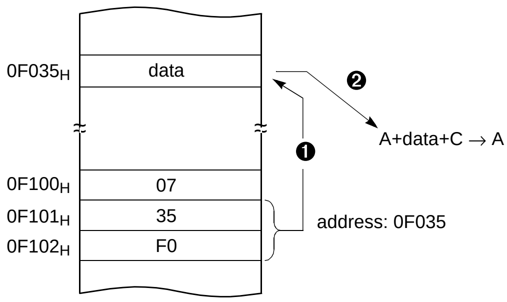
(4) Absolute Addressing → !abs

Absolute addressing sets corresponding memory data to Data, i.e. second byte (Operand I) of command becomes lower level address and third byte (Operand II) becomes upper level address.
With 3 bytes command, it is possible to access to whole memory area.

ADC, AND, CMP, CMPX, CMPY, EOR, LDA, LDX, LDY, OR, SBC, STA, STX, STY

Example;

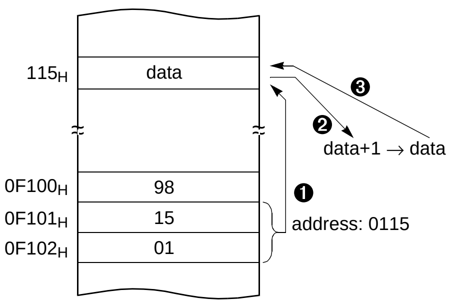
0735F0 ADC !$0F035_H$;A ←ROM[$0F035_H$]



The operation within data memory (RAM)
ASL, BIT, DEC, INC, LSR, ROL, ROR

Example; Addressing accesses the address 0135_H regardless of G-flag and RPR.

```
981501   INC   !0115H      ;A ←ROM[115H]
```



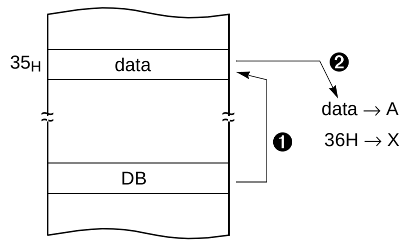
X indexed direct page, auto increment→ {X}+

In this mode, a address is specified within direct page by the X register and the content of X is increased by 1.

LDA, STA

Example; G=0, $X=35_H$

```
DB       LDA   {X}+
```



(5) Indexed Addressing

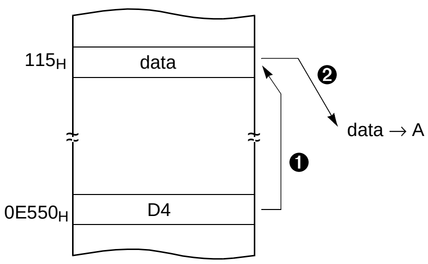
X indexed direct page (no offset) → {X}

In this mode, a address is specified by the X register.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA, XMA

Example; $X=15_H$, G=1, $RPR=01_H$

```
D4       LDA   {X}         ;ACC←RAM[X].
```



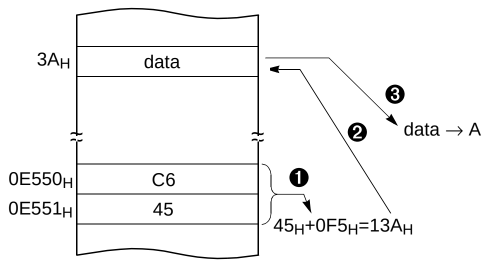
X indexed direct page (8 bit offset) → dp+X

This address value is the second byte (Operand) of command plus the data of X-register. And it assigns the memory in Direct page.

ADC, AND, CMP, EOR, LDA, LDY, OR, SBC, STA STY, XMA, ASL, DEC, INC, LSR, ROL, ROR

Example; G=0, $X=0F5_H$

```
C645     LDA   45H+X
```



Y indexed direct page (8 bit offset) → dp+Y

This address value is the second byte (Operand) of command plus the data of Y-register, which assigns Memory in Direct page.

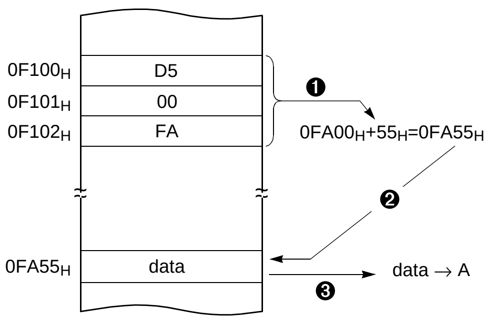
This is same with above (2). Use Y register instead of X.

Y indexed absolute → !abs+Y

Sets the value of 16-bit absolute address plus Y-register data as Memory. This addressing mode can specify memory in whole area.

Example; Y=55_H

```
D500FA   LDA    !0FA00H+Y
```



(6) Indirect Addressing

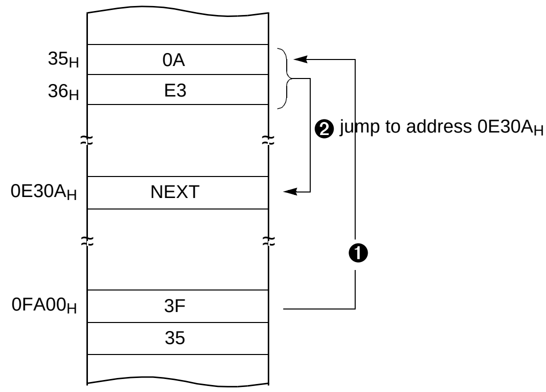
Direct page indirect → [dp]

Assigns data address to use for accomplishing command which sets memory data (or pair memory) by Operand. Also index can be used with Index register X,Y.

JMP, CALL

Example; G=0

```
3F35     JMP    [35H]
```



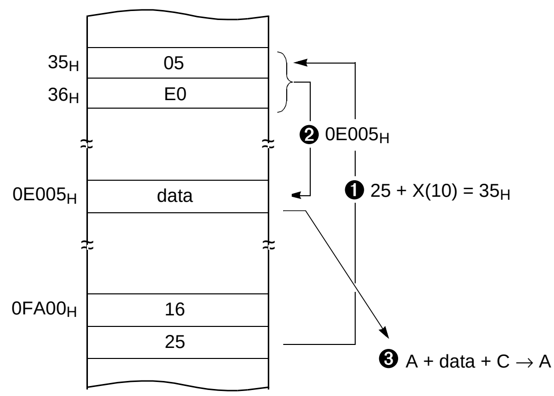
X indexed indirect → [dp+X]

Processes memory data as Data, assigned by 16-bit pair memory which is determined by pair data [dp+X+1][dp+X] Operand plus X-register data in Direct page.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA

Example; G=0, X=10_H

```
1625     ADC    [25H+X]
```



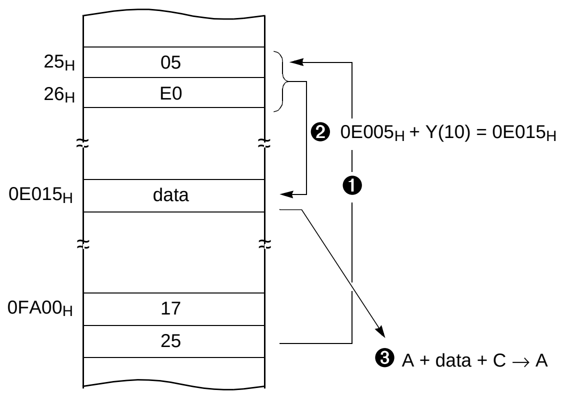
Y indexed indirect → [dp]+Y

Processes memory data as Data, assigned by the data [dp+1][dp] of 16-bit pair memory paired by Operand in Direct page plus Y-register data.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA

Example; G=0, Y=10_H

```
1725     ADC    [25H]+Y
```



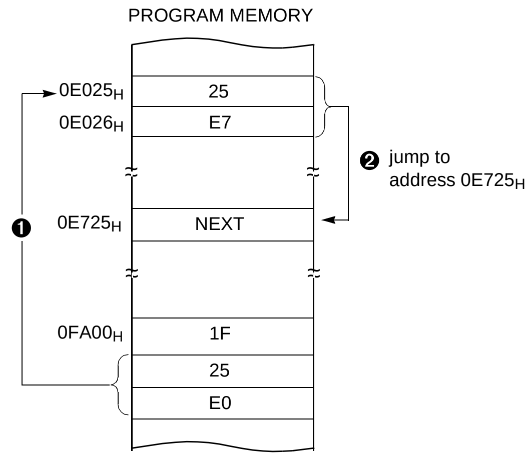
Absolute indirect → [!abs]

The program jumps to address specified by 16-bit absolute address.

JMP

Example; G=0

```
1F25E0   JMP    [!0E025H]
```



9. I/O PORTS

The GMS81C5108 has seven ports (R0, R1, R2 and R3), and LCD segment port (SEG0~SEG36), and LCD common port (COM0~COM3).

These ports pins may be multiplexed with an alternate function for the peripheral features on the device.

9.1 Registers for Port

Port Data Registers

The Port Data Registers (R0, R1, R2, R3) are represented as a D-Type flip-flop, which will clock in a value from the internal bus in response to a “write to data register” signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a “read data register” signal from the CPU. The level of the port pin itself is placed on the internal bus in response to “read data register” signal from the CPU. Some instructions that read a port activating the “read register” signal, and others activating the “read pin” signal.

Port Direction Registers

All pins have data direction registers which can define these ports as output or input. A “1” in the port direction register configure the corresponding port pin as output. Conversely, write “0” to the corresponding bit to specify it as input pin. For example, to use the even numbered bit of R0 as output ports and the odd numbered bits as input ports, write “55_H” to address $0C8_H$ (R0 port direction register) during initial setting as shown in Figure 9-1.

All the port direction registers in the GMS81C5108 have 0 written to them by reset function. On the other hand, its initial status is input.

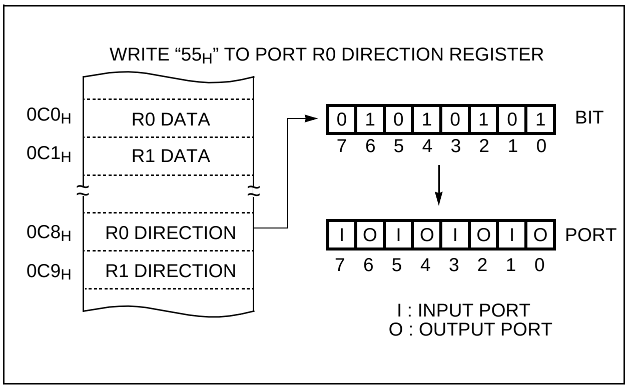


Figure 9-1 Example of port I/O assignment

Pull-up Control Registers

The R0, R1,R2 and R3 ports have internal pull-up resistors. Figure 9-2 shows a functional diagram of a typical pull-up port. It is connected or disconnected by Pull-up Control register (R*n*PU). The value of that resistor is typically 100kΩ. Refer to DC characteristics for more details.

When a port is used as key input, input logic is firmly either low or high, therefore external pull-down or pull-up resisters are required practically. The GMS81C5108 has internal pull-up, it can be logic high by pull-up that can be able to configure either connect or disconnect individually by pull-up control registers R*n*PU.

When ports are configured as inputs and pull-up resistor is selected by software, they are pulled to high.

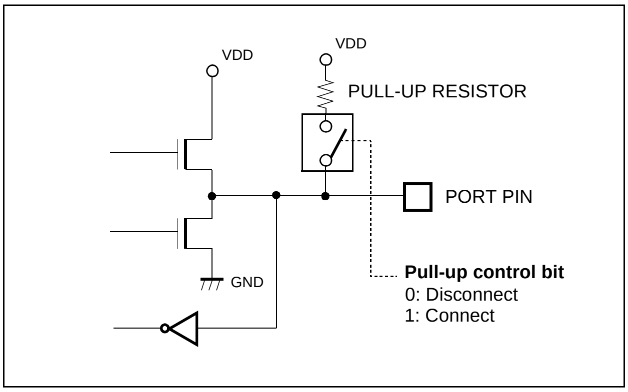


Figure 9-2 Pull-up Port Structure

Open drain port Registers

The R0, R1, R2 and R3 ports have open drain port resistors R0CR~R3CR.

Figure 9-3 shows an open drain port configuration by control register. It is selected as either push-pull port or open-drain port by R0CR, R1CR, R2CR and R3CR.

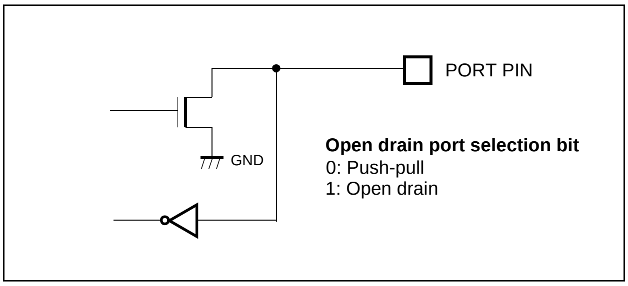


Figure 9-3 Open-drain Port Structure

9.2 I/O Ports Configuration

R0 Ports

R0 is an 8-bit CMOS bidirectional I/O port (address $0C0_H$). Each I/O pin can independently used as an input or an output through the R0DR register (address $0C8_H$).

R0 has internal pull-ups that is independently connected or disconnected by R0PU. The control registers for R0 are shown below.

R0 Data Register ADDRESS : $0C0_H$ RESET VALUE : 00_H

R0	R07	R06	R05	R04	R03	R02	R01	R00

R0 Direction Register ADDRESS : $0C8_H$ RESET VALUE : 00_H

R0DR

Port Direction
0: Input
1: Output

R0 Pull-up Selection Register ADDRESS :$0D0_H$ RESET VALUE : 00_H

R0PU

Pull-up select
0: Without pull-up
1: With pull-up

R0 Open Drain Selection Register ADDRESS :$0D4_H$ RESET VALUE : 00_H

R0CR

Open Drain select
0: No Open Drain
1: Open Drain

Port Mode Register ADDRESS :$0D9_H$ RESET VALUE : $-0-00000_B$

PMR	-	PWMO	-	BUZ	EC0	INT2	INT1	INT0

PWMO (PWM Output)
0: R31 Port
1: PWM

BUZ (Buzzer Output)
0: R04 Port
1: BUZ

EC0 (Timer0 Event Input)
0: R03 Port
1: EC0

INT2 (External Interrupt)
0: R02 Port
1: INT2

INT1 (External Interrupt)
0: R01 Port
1: INT1

INT0 (External Interrupt)
0: R00 Port
1: INT0

In addition, Port R0 and R3 are multiplexed with various special features. The control register PMR (address 0D9H) controls the selection of alternate function. After reset, this value is “0”, port may be used as normal I/O port.
To use alternate function such as External Interrupt rather than normal I/O, write “1” in the corresponding bit of PMR0.

.

Port Pin	Alternate Function
R00	INT0 (External Interrupt 0)
R01	INT1 (External Interrupt 1)
R02	INT2 (External Interrupt 2)
R03	EC0 (Timer0 Event Input)
R04	BUZ (Buzzer Output)
R31	PWM (PWM Output)

R1 Ports

R1 is an 8-bit CMOS bidirectional I/O port (address $0C1_H$). Each I/O pin can independently used as an input or an output through the R1DR register (address $0C9_H$).

R1 has internal pull-ups that is independently connected or disconnected by register R1PU. If the key scan function is used, these pin can input the key switch signal without external pull-up registers. For more details refer to "18. KEY SCAN" on page 70.

The control registers for R1 are shown below.

R1 Data Register ADDRESS : $0C1_H$ RESET VALUE : 00_H

R1	R17	R16	R15	R14	R13	R12	R11	R10

R1 Direction Register ADDRESS : $0C9_H$ RESET VALUE : 00_H

R1DR

Port Direction
0: Input
1: Output

R1 Pull-up Selection Register ADDRESS : $0D1_H$ RESET VALUE : 00_H

R1PU

Pull-up select
0: Without pull-up
1: With pull-up

R1 Open Drain Selection Register ADDRESS :$0D5_H$ RESET VALUE : 00_H

R1CR

Open Drain select
0: No Open Drain
1: Open Drain

KEY SCAN Mode Register ADDRESS :$0F0_H$ RESET VALUE : 00_H

KSMR

KEY Input select
0: Port selection
1: KS selection

Port R1 is multiplexed with various special features.The control registers controls the selection of alternate function. After reset, this value is “0”, port may be used as normal I/O port. The way to select alternate function such as comparator input or buzzer will be shown in each peripheral section.

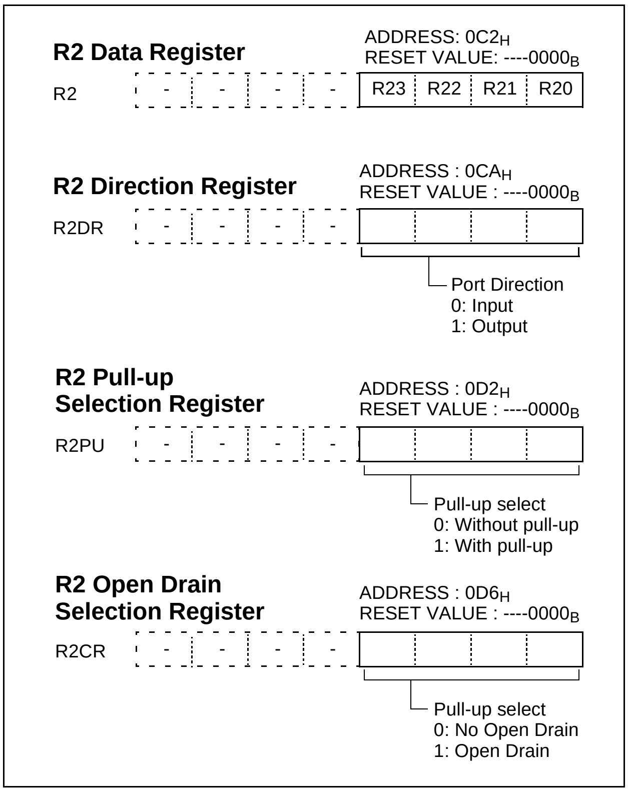
In addition, R1 port is used as key scan function which operate with normal input port.

Input or output is configured automatically by each function register (KSMR) regardless of R1DR.

R2 Port

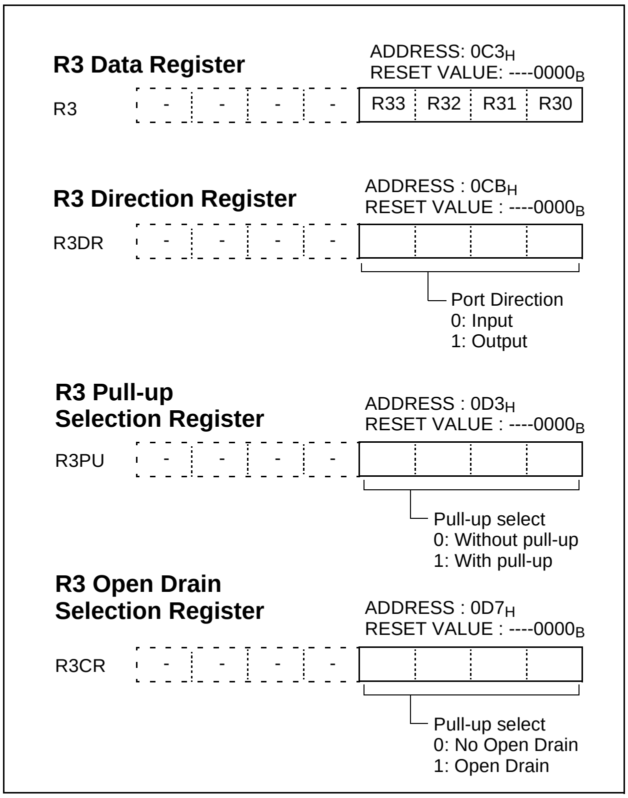
R2 is a 4-bit CMOS bidirectional I/O port (address $0C2_H$). Each I/O pin can independently used as an input or an output through the R2DR register (address $0CA_H$).

R2 has internal pull-ups that is independently connected or disconnected by R2PU (address $0D2_H$). The control registers for R2 are shown as below.



R3 Port

R3 is a 4-bit CMOS bidirectional I/O port (address $0C3_H$). Each I/O pin can independently used as an input or an output through the R3DR register (address $0CB_H$).



SEG0~SEG36

Segment signal output pins for the LCD display. See "19. LCD DRIVER" on page 71 for details.

COM0~COM3

Common signal output pins for the LCD display. See "19. LCD DRIVER" on page 71 for details.

SEG34~SEG36 and COM1~COM3 are selected by LCDD of the LCR register.

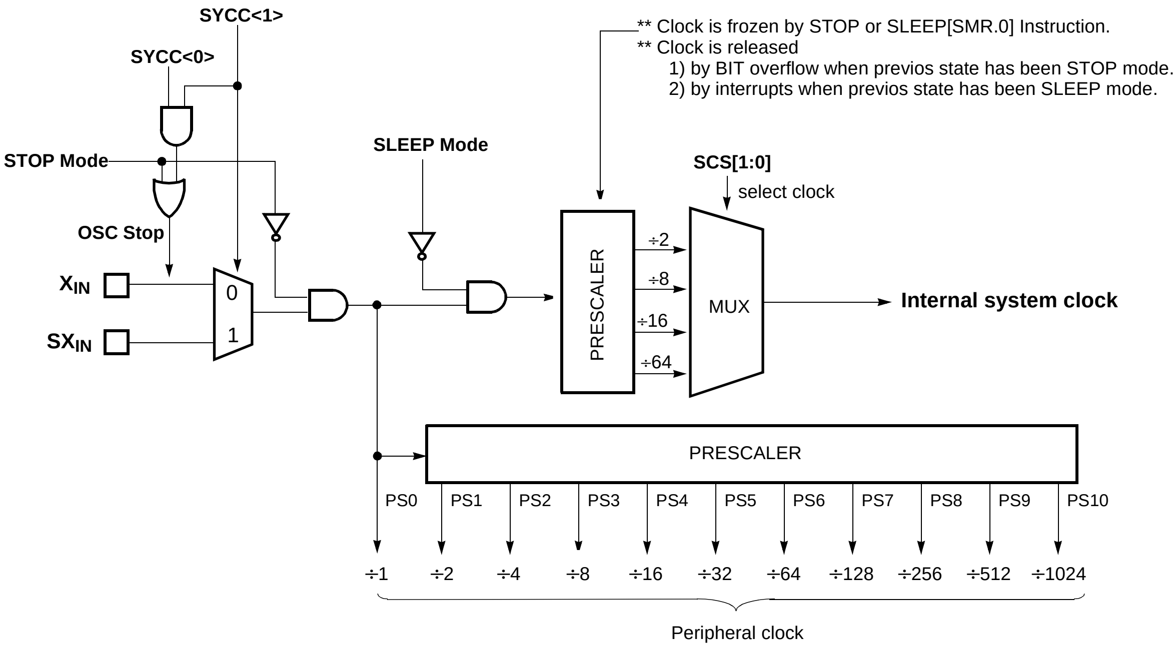
10. CLOCK GENERATOR

As shown in Figure 10-1, the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. It contains two oscillators: a main-frequency clock oscillator and a sub-frequency clock oscillator. Power consumption can be reduced by switching them to the low power operation frequency clock can be easily obtained by attaching a resonator between the X_{IN} and X_{OUT} pin and the SX_{IN} and SX_{OUT} pin, respectively. The system clock can also be obtained from the external oscillator.

The clock generator produces the system clocks forming clock pulse, which are supplied to the CPU and the peripheral hardware. The internal system clock can be selected by bit2, and bit3 of the system clock mode register (SCMR). The registers are shown in Figure 10-2.

CPU clock	Instruction cycle time	
	f_{MAIN} = 4MHz	f_{SUB} = 32.768kHz
÷ 2	0.5 us	61 us
÷ 8	2.0 us	244 us
÷ 16	4.0 us	488 us
÷ 64	16.0 us	1953 us

To the peripheral block, the clock among the not-divided original clocks, divided by 2, 4,..., up to 1024 can be provided. Peripheral clock is enabled or disabled by STOP instruction. The peripheral clock is controlled by clock control register (CKCTLR). See "11. BASIC INTERVAL TIMER" on page 43 for details.



f_{EX} (Hz)		PS0	PS1	PS2	PS3	PS4	PS5	PS6	PS7	PS8	PS9	PS10
4M	Frequency	4M	2M	1M	500K	250K	125K	62.5K	31.25K	15.63K	7.183K	3.906K
	period	250n	500n	1u	2u	4u	8u	16u	32u	64u	128u	256u

Figure 10-1 Block Diagram of Clock Generator

The system clock is decided by bit1 of the system clock mode register, SCMR. In selection Sub clock, to oscillate or stop the Main clock is decided by bit0 of SCMR.

On the initial reset, internal system clock is PS1 which is the fastest and other clock can be provided by bit2 and bit3 of SCMR.

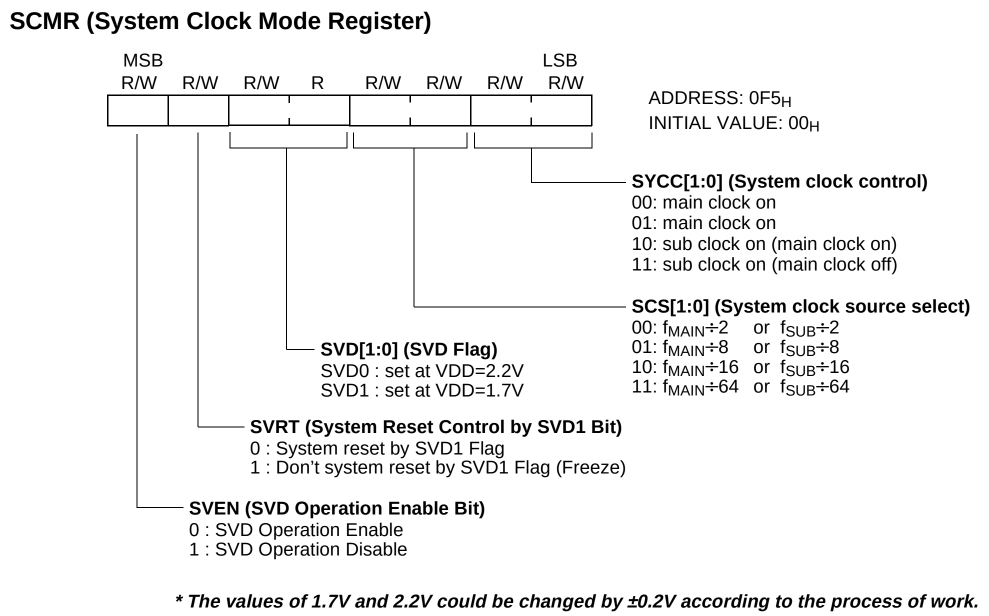


Figure 10-2 SCMR : System Clock Control Registers

10.1 Operation Mode

The system clock controller starts or stops the main-frequency clock oscillator and switches between the sub frequency clock. The operating mode is generally divided into the main active mode and the sub active mode, which are controlled by System clock mode register (SCMR). Figure 10-3 shows the operating mode transition diagram.

System clock control is performed by the system clock mode register, SCMR. During reset, this register is initialized to “0” so that the main-clock operating mode is selected.

Main Active mode

This mode is fast-frequency operating mode.
The CPU and the peripheral hardwares are operated on the high-frequency clock. At reset release, this mode is invoked.

Sub Active mode

This mode is low-frequency operating mode
In this mode, the CPU and the peripheral hardware clock are provided by low-frequency clock oscillation, so power consumption can be reduced.

SLEEP mode

In this mode, the CPU clock stops while peripherals and the oscillation source continue to operate normally.

STOP mode

In this mode, the system operations are all stopped, holding the internal states valid immediately before the stop at the low power consumption level.

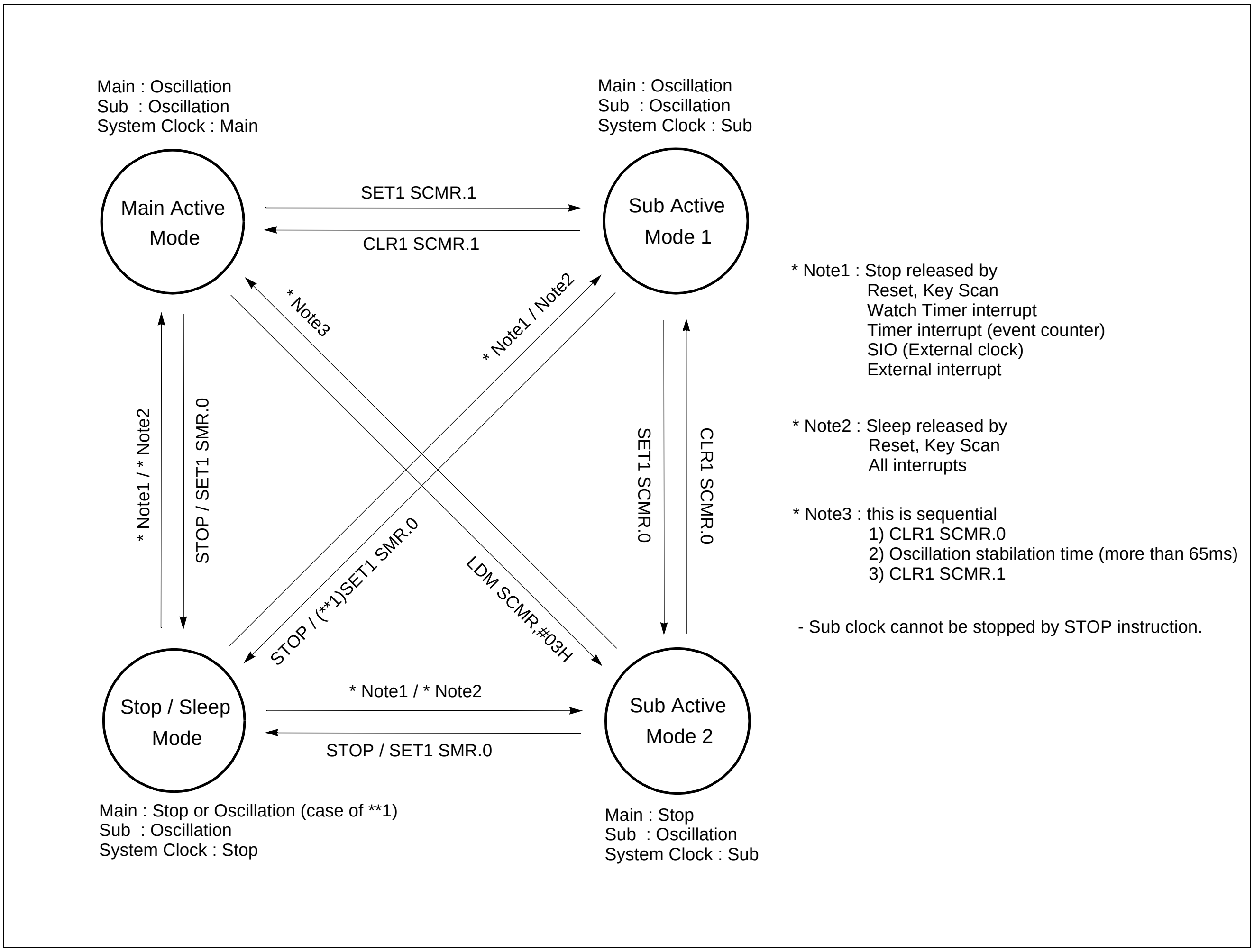


Figure 10-3 Operating Mode

10.2 Operation Mode Switching

In the Main active mode, only the high-frequency clock oscillator is used.

In the Sub active mode, the low-frequency clock oscillation is used, so the low power voltage operation or the low power consumption operation can be enabled. Instruction execution does not stop during the change of operation mode. In this case, some peripheral hardware capabilities may be affected. For details, refer to the description of the relevant operation.

The following describes the switching between the Main active mode and the Sub active mode. During reset, the system clock mode register is initialized at the Main active mode. It must be set to the Sub active mode for reducing the power consumption.

Switching from Main active to Sub active

First, write "02_H" into lower 2 bits of SCMR to switch the main system clock to the sub-frequency clock.
Next, write "03_H" to turn off main frequency oscillation.

Example:

```
  :
  :
  :
LDM  SCMR,#02H  ;Switch to sub active
LDM  SCMR,#03H  ;Turn off main clock
  :
  :
```

Returning from Sub active to Main active

First, write "02_H" into lower 2 bits of the SCMR to turn on the main-frequency oscillation. This time, the stabilization (warm-up) time needs to be taken by the software delay routine. Sub active mode can also be released by setting the $\overline{\text{RESET}}$ pin to low, which immediately performs the reset operation. After reset, the GMS81C5108 is placed in Main active mode.

Example:

```
  :
  :
  :
LDM  SCMR,#02H  ;Turn on main-clock
CALL DELAY      ;Wait until stable
LDM  SCMR,#0    ;Move to main active
         :
         :
         :

;about 65ms software delay
DELAY:   LDA   #0
DELAY0:  INC   A
         CMP   #85H
         BCC   DELAY0
         RET
```

Shifting from the Normal operation to the SLEEP mode

By setting bit 0 of SMR, the CPU clock stops and the SLEEP mode is invoked. The CPU stops while other peripherals are operate normally.

The way of release from this mode is RESET and all available interrupts.

For more detail, See " SLEEP Mode" on page 39

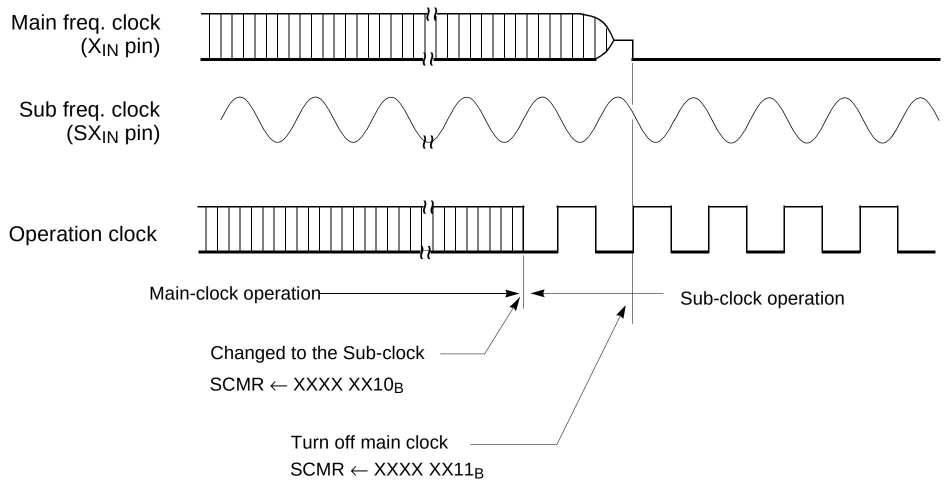
Shifting from the Normal operation to the STOP mode

By executing STOP instruction, the main-frequency clock oscillation stops and the STOP mode is invoked. But sub-frequency clock oscillation is operated continuously.
After the STOP operation is released by reset, the operation mode is changed to Main active mode.

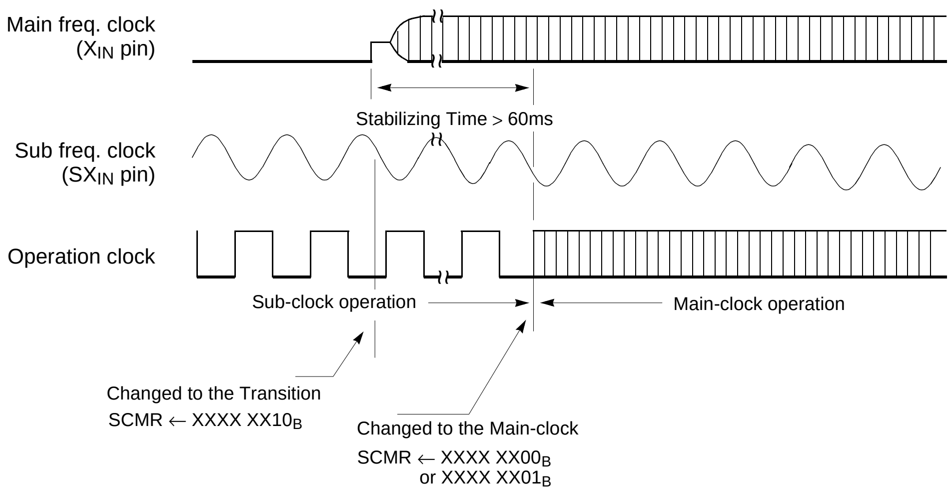
The methods of release are RESET, Key scan interrupt, Watch Timer interrupt, Timer/Event counter1 (EC0 pin), SIO (External clock) and External Interrupt.

For more details, see " STOP Mode" on page 40.

Note: *In the STOP and SLOW operating modes, the power consumption by the oscillator and the internal hardware is reduced. However, the power for the pin interface (depending on external circuitry and program) is not directly associated with the low-power consumption operation. This must be considered in system design as well as interface circuit design.*



(a) Main active mode → Sub active mode



(b) Sub active mode → Main active mode

Figure 10-4 System Clock Switching Timing

10.3 Power Saving Operation

GMS81C5108 has 2 power-saving mode. In power-saving mode, power consumption is reduced considerably that in Battery operation Battery life can be extended a lot.

Sleep mode is entered by setting bit 0 of Sleep Mode Register (SMR), and STOP Mode is entered by STOP instruction.

SLEEP Mode

In this mode, the internal oscillation circuits remain active.

Oscillation continues and peripherals are operate normally but CPU stops. Movement of all Peripherals is shown in Table 10-1. Sleep mode is entered by setting bit 0 of SMR (address $0DE_H$).

It is released by RESET or interrupt. To be released by interrupt, interrupt should be enabled before Sleep mode.

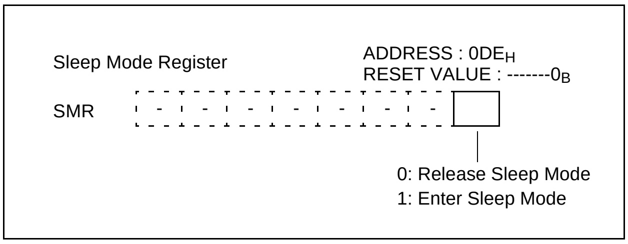


Figure 10-5 SLEEP Mode Register

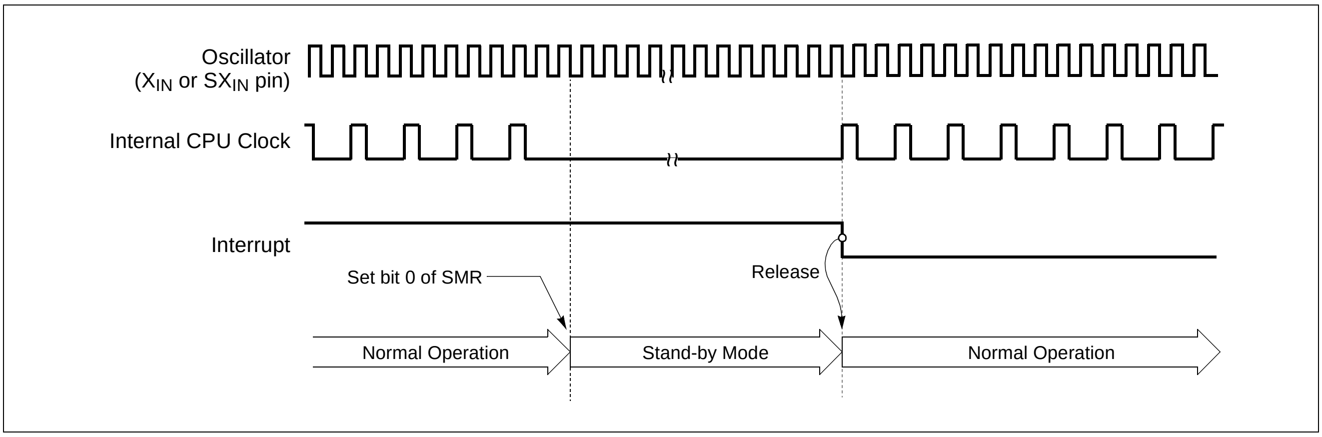


Figure 10-6 Sleep Mode Release Timing by External Interrupt

.

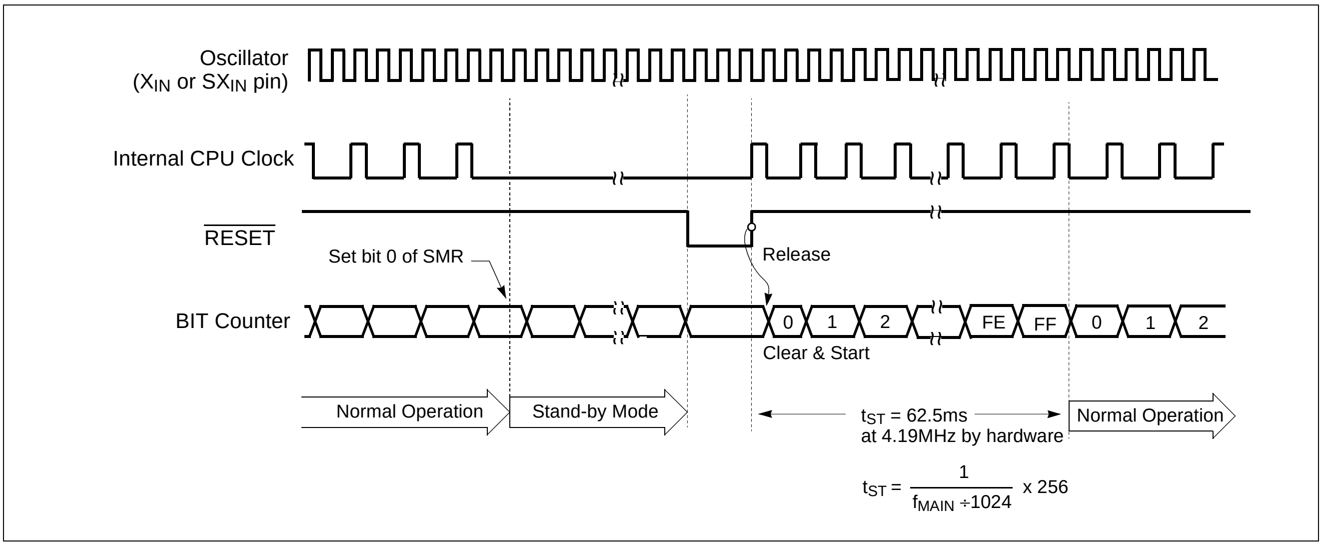


Figure 10-7 SLEEP Mode Release Timing by RESET pin

STOP Mode

For applications where power consumption is a critical factor, device provides STOP mode for reducing power consumption.

Start The Stop Operation

The STOP mode can be entered by STOP instruction during program execution. In Stop mode, the on-chip main-frequency oscillator, system clock, and peripheral clock are stopped (Watch timer clock is oscillating continuously:. With the clock frozen, all functions are stopped, but the on-chip RAM and Control registers are held. The port pins output the values held by their respective port data register, the port direction registers. The status of peripherals during Stop mode is shown below.

Peripheral	STOP Mode	Sleep Mode
CPU	All CPU operations are disabled	All CPU operations are disabled
RAM	Retain	Retain
LCD driver	Operates continuously	Operates continuously
Basic Interval Timer	Halted	Operates continuously
Timer/Event counter 0,1	Halted (Only when the Event counter mode is enabled, Timer 0,1 operates normally)	Timer/Event counter 0,1 operates continuously
Watch Timer	Operates continuously	Operates continuously
Key Scan	Active	Active
Main-oscillation	Stop (X_{IN}=L, X_{OUT}=L)	Oscillation[1]
Sub-oscillation	Oscillation	Oscillation
I/O ports	Retain	Retain
Control Registers	Retain	Retain
Release method	by RESET, Key Scan interrupt, SIO interrupt, Watch Timer interrupt, Timer interrupt (EC0), and External interrupt	by RESET, All interrupts

Table 10-1 Peripheral Operation during Power Saving Mode

1. refer to the Table 10-2

Operating Clock source	Main Operating Mode	Main Sleep Mode	Sub Operating Mode	Sub Sleep Mode	Stop Mode
Main Clock	Oscillation	Oscillation	SCMR<1:0> 00,01,10 → Oscillation 11 → Stop	SCMR<1:0> 00,01,10 → Oscillation 11 → Stop	Stop
Sub Clock	Oscillation	Oscillation	Oscillation	Oscillation	Oscillation
System Clock	Active	Stop	Active	Stop	Stop
Peri. Clock	Active	Active	Active	Active	Stop

Table 10-2 Clock Operation of STOP and SLEEP mode

Note: *Since the X_{IN} pin is connected internally to GND to avoid current leakage due to the crystal oscillator in STOP mode, do not use STOP instruction when an external clock is used as the main system clock.*

In the Stop mode of operation, V_{DD} can be reduced to minimize power consumption. Be careful, however, that V_{DD} is not reduced before the Stop mode is invoked, and that V_{DD} is restored to its normal operating level before the Stop mode is terminated.

The reset should not be activated before V_{DD} is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize.
And after STOP instruction, at least two or more NOP instruction should be written as shown in example below.

Example)

```
        :
        LDM    CKCTLR,#0000_1111B
        STOP
        NOP
        NOP
        :
```

The Interval Timer Register CKCTLR should be initialized by software in order that oscillation stabilization time should be longer than 20ms before STOP mode.

Release the STOP mode

The exit from STOP mode is using hardware reset or external interrupt, watch timer, SIO interrupt, key scan or timer interrupt (EC0).

To release STOP mode, corresponding interrupt should be enabled before STOP mode.
Specially as a clock source of Timer/Event counter, EC0 pin can release it by Timer/Event counter Interrupt request.

Reset redefines all the control registers but does not change the on-chip RAM. External interrupts allow both on-chip RAM and Control registers to retain their values.

Start-up is performed to acquire the time for stabilizing oscillation. During the start-up, the internal operations are all stopped.

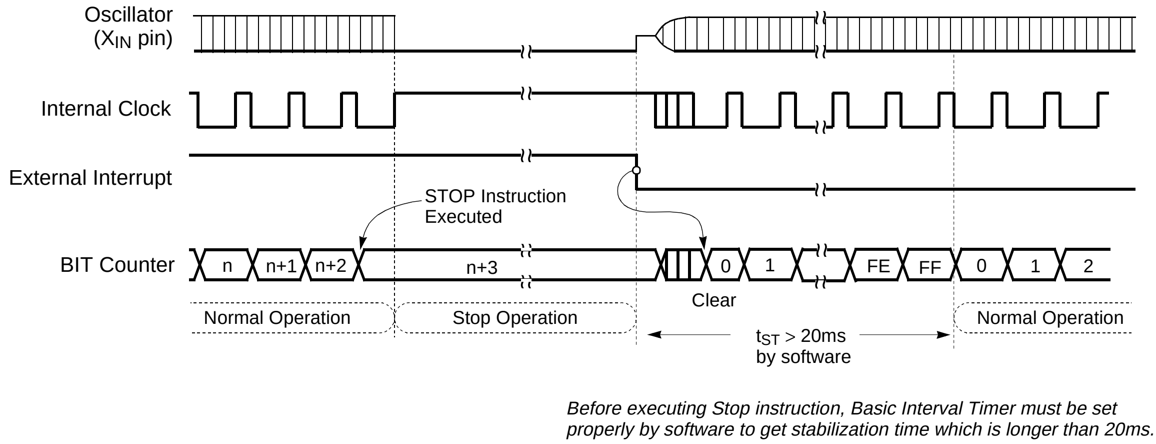


Figure 10-8 STOP Mode Release Timing by External Interrupt

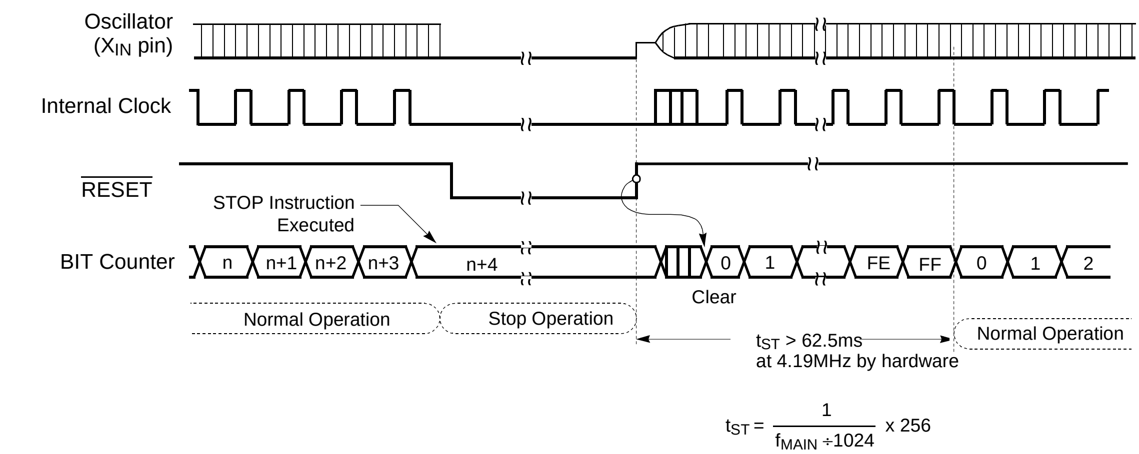


Figure 10-9 STOP Mode Release Timing by RESET

Minimizing Current Consumption

The Stop mode is designed to reduce power consumption. To minimize current drawn during Stop mode, the user should turn-off output drivers that are sourcing or sinking current, if it is practical.

Note: *In the STOP operation, the power dissipation associated with the oscillator and the internal hardware is lowered; however, the power dissipation associated with the pin interface (depending on the external circuitry and program) is not directly determined by the hardware operation of the STOP feature. This point should be little current flows when the input level is stable at the power voltage level (V_{DD}/V_{SS}); however, when the input level becomes higher than the power voltage level (by approximately 0.3V), a current begins to flow. Therefore, if cutting off the output transistor at an I/O port puts the pin signal into the high-impedance state, a current flow across the ports input transistor, requiring it to fix the level by pull-up or other means.*

It should be set properly that current flow through port doesn't exist.

First consider the setting to input mode. Be sure that there is no current flow after considering its relationship with external circuit. In input mode, the pin impedance viewing from external MCU is very high that the current doesn’t flow.

But input voltage level should be V_{SS} or V_{DD}. Be careful that if unspecified voltage, i.e. if unfirmed voltage level (not V_{SS}or V_{DD}) is applied to input pin, there can be little current (max. 1mA at around 2V) flow.

If it is not appropriate to set as an input mode, then set to output mode considering there is no current flow. Setting to High or Low is decided considering its relationship with external circuit. For example, if there is external pull-up resistor then it is set to output mode, i.e. to High, and if there is external pull-saving register, it is set to low.

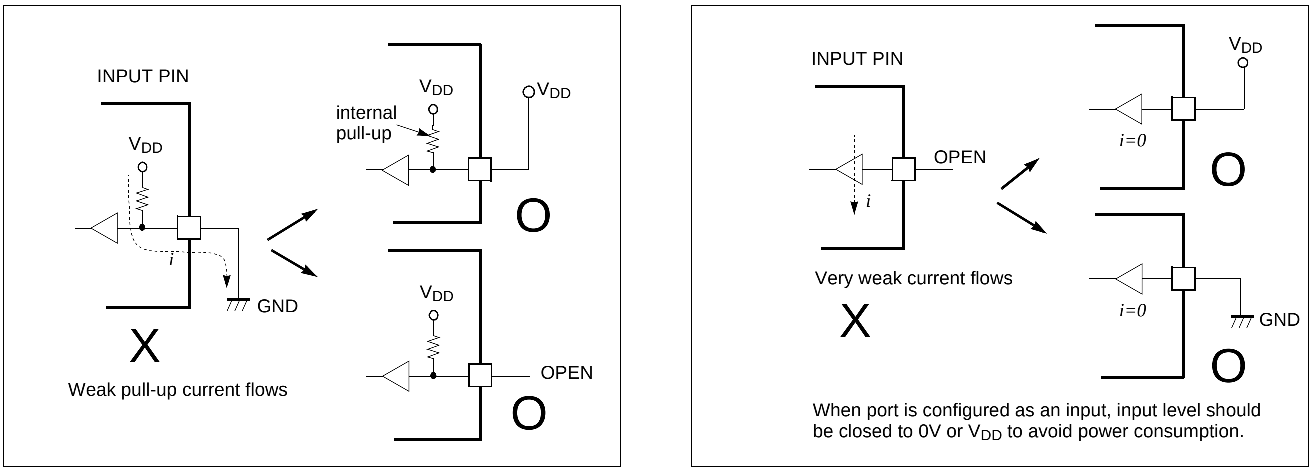


Figure 10-10 Application Example of Unused Input Port

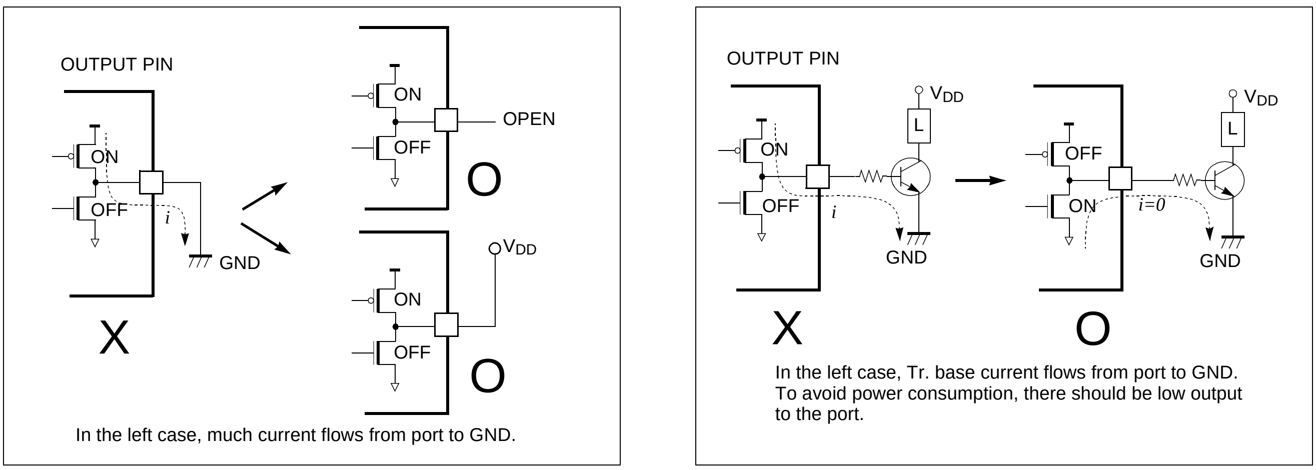


Figure 10-11 Application Example of Unused Output Port

11. BASIC INTERVAL TIMER

The GMS81C5108 has one 8-bit Basic Interval Timer that is free-run and can not stop. Block diagram is shown in Figure 11-1.

The Basic Interval Timer Register (BITR) is increased every internal count pulse which is divided by prescaler. Since prescaler has divided ratio by 8 to 1024, the count rate is 1/8 to 1/1024 of the oscillator frequency. After reset, the BCK bits are all set, so the longest oscillation stabilization time is obtained.

It also provides a Basic interval timer interrupt (BITF). The count overflow of BITR from FF_H to 00_H causes the interrupt to be generated. The Basic Interval Timer is controlled by the clock control register (CKCTLR) shown in Figure 11-2.

Source clock can be selected by lower 3 bits of CKCTLR. When write “1” to bit BCL of CKCTLR, BITR register is cleared to “0” and restart to count up. The bit BCL becomes “0” automatically after one machine cycle by hardware.

BITR and CKCTLR are located at same address, and address $0F4_H$ is read as a BITR, and written to CKCTLR.

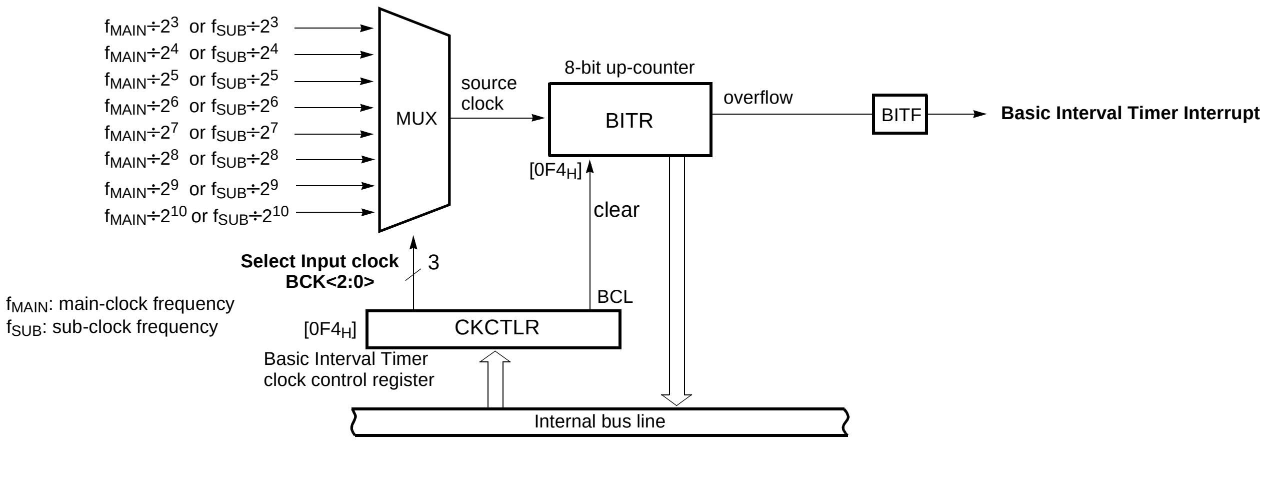


Figure 11-1 Block Diagram of Basic Interval Timer

BCK <2:0>	Source clock		Interrupt (overflow) Period	
	SCMR[1:0]= 00 or 01	SCMR[1:0]= 10 or 11	At f_{MAIN}=4MHz	At f_{SUB}=32.768kHz
000	$f_{MAIN}÷2^3$	$f_{SUB}÷2^3$	0.512 ms	62.5 ms
001	$f_{MAIN}÷2^4$	$f_{SUB}÷2^4$	1.024	125.0
010	$f_{MAIN}÷2^5$	$f_{SUB}÷2^5$	2.048	250.0
011	$f_{MAIN}÷2^6$	$f_{SUB}÷2^6$	4.096	500.0
100	$f_{MAIN}÷2^7$	$f_{SUB}÷2^7$	8.192	1000.0
101	$f_{MAIN}÷2^8$	$f_{SUB}÷2^8$	16.384	2000.0
110	$f_{MAIN}÷2^9$	$f_{SUB}÷2^9$	32.768	4000.0
111	$f_{MAIN}÷2^{10}$	$f_{SUB}÷2^{10}$	65.536	8000.0

Table 11-1 Basic Interval Timer Interrupt Time

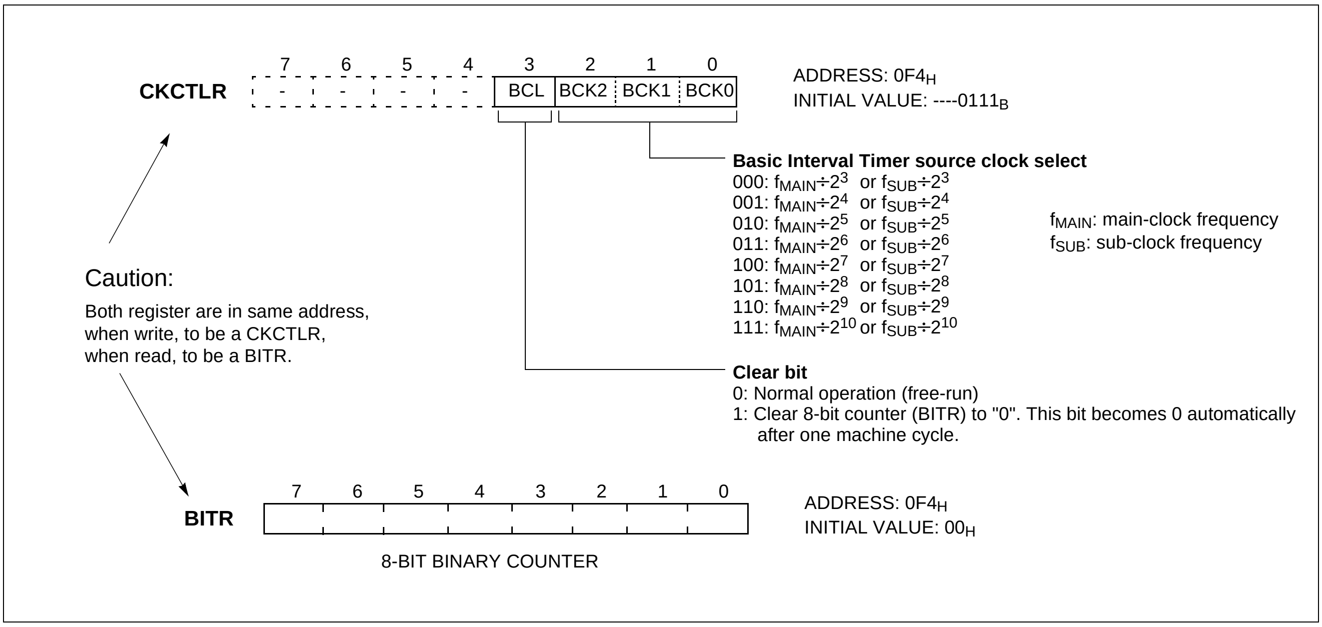


Figure 11-2 BITR: Basic Interval Timer Mode Register

Example 1:

Interrupt request flag is generated every 8.192ms at 4MHz.

```
:
LDM    CKCTLR,#0CH
SET1   BITE
EI
:
```

12. Timer / Counter

Timer/Event Counter consists of prescaler, multiplexer, 8-bit timer data register, 8-bit counter register, mode register, input capture register and Comparator as shown in Figure 12-3. And the PWM high register for PWM is consisted separately.

The timer/counter has seven operating modes.
- 8 Bit Timer/Counter Mode
- 8 Bit Capture Mode
- 8 Bit Compare Output Mode
- 16 Bit Timer/Counter Mode
- 16 Bit Capture Mode
- 16 Bit Compare Output Mode
- PWM Mode

In the "timer" function, the register is increased every internal clock input. Thus, one can think of it as counting internal clock input. Since a least clock consists of 2 and most clock consists of 1024 oscillator periods, the count rate is 1/2 to 1/1024 of the oscillator frequency in Timer0. And Timer1 can use the same clock source too. In addition, Timer1 has more fast clock source (1/1 to 1/8).

In the "counter" function, the register is increased in response to a 0-to-1 (rising edge) transition at its corresponding external input pin EC0 (Timer 0).

In addition the "capture" function, the register is increased in response external interrupt same with timer function. When external interrupt edge input, the count register is captured into capture data register CDRx.

Timer1 is shared with "PWM" function and "Compare output" function.

Example 1:

Timer 0 = 8-bit timer mode, 8ms interval at 4MHz
Timer 1 = 8-bit timer mode, 4ms interval at 4MHz

```
LDM   SCMR,#0        ;Main clock mode
LDM   TDR0,#249
LDM   TM0,#0001_0011B
LDM   TDR1,#124
LDM   TM1,#0000_1111B

SET1  T0E
SET1  T1E
EI
 :
 :
 :
```

Example 2:

Timer0 = 16-bit timer mode, 0.5s at 4MHz

```
LDM   SCMR,#0        ;Main clock mode
LDM   TDR0,#23H
LDM   TDR1,#0F4H
LDM   TM0,#0FH       ;FMAIN/32, 8us
LDM   TM1,#4CH

SET1  T0E
EI
 :
 :
 :
```

Example 3:

Timer0 = 8-bit event counter, 2ms interval at 4MHz
Timer1 = 8-bit capture mode, 2us sampling count.

```
LDM   TDR0,#99          ;99+1, 100 count
LDM   TM0,#01FH         ;event counter
LDM   R0DR,#XXXX_1XXXB  ;R03input

LDM   IESR,#XXXX_01XXB  ;FALLING
LDM   PMR,#XXXX_1X1XB   ;EC0,INT1
LDM   TDR1,#0FFH
LDM   TM1,#0001_1011B   ;2us

SET1  T0E;ENABLE TIMER 0
SET1  T1E;ENABLE TIMER 1
SET1  INT1E;ENABLE EXT. INT1
EI
 :
```

X: don't care.

Example 4:

Timer0 = 16-bit capture mode, 8us sampling count. at 4MHz

```
LDM   TDR0,#0FFH
LDM   TDR1,#0FFH
LDM   TM0,#02FH
LDM   TM1,#04FH

LDM   IESR,#XXXX_XX01B
LDM   PMR,#XXXX_XXX1B     ;AS INT0

SET1  T0E;ENABLE TIMER 0
SET1  INT0E;ENABLE EXT. INT0
EI
 :
```

X: don't care.

TM0 (Timer0 Mode Register)

Bit :	7	6	5	4	3	2	1	0
			R/W	R/W	R/W	R/W	R/W	R/W
	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST

Bits 7–6: Reserved

ADDRESS: $0E0_H$
INITIAL VALUE: $--000000_B$

CAP0 (Capture Mode Selection Bit)
0: Capture Disable
1: Capture Enable

T0CN (Timer 0 Continue Start)
0: Stop Counting
1: Start Counting

T0ST (Timer 0 Start Control)
0: stop counting
1: clear the counter and start count again

T0CK[2:0] (Timer 0 Input Clock Selection)
000: $f_{MAIN}\div 2$ or $f_{SUB}\div 2$
001: $f_{MAIN}\div 2^2$ or $f_{SUB}\div 2^2$
010: $f_{MAIN}\div 2^3$ or $f_{SUB}\div 2^3$
011: $f_{MAIN}\div 2^5$ or $f_{SUB}\div 2^5$
100: $f_{MAIN}\div 2^7$ or $f_{SUB}\div 2^7$
101: $f_{MAIN}\div 2^9$ or $f_{SUB}\div 2^9$
110: $f_{MAIN}\div 2^{10}$ or $f_{SUB}\div 2^{10}$
111: External Event clock (EC0)

f_{MAIN}: main-clock frequency
f_{SUB}: sub-clock frequency

TM1 (Timer1 Mode Register)

Bit :	7	6	5	4	3	2	1	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	POL	16BIT	PWME	CAP1	T1CK1	T1CK0	T1CN	T1ST

ADDRESS: $0E2_H$
INITIAL VALUE: 00000000_B

POL (PWM Output Polarity Selection)
0: Duty Active Low
1: Duty Active High

16BIT (16 Bit Mode Selection)
0: 8-Bit Mode
1: 16-Bit Mode

PWME (PWM Enable Bit)
0: PWM Disable
1: PWM Enable

CAP1 (Capture Mode Selection Bit)
0: Capture Disable
1: Capture Enable

T1CK[1:0] (Timer 1 Input Clock Selection)
00: f_{MAIN} or f_{SUB}
01: $f_{MAIN}\div 2$ or $f_{SUB}\div 2$
10: $f_{MAIN}\div 2^3$ or $f_{SUB}\div 2^3$
11: Timer 0 Clock

T1CN (Timer 1 Continue Start)
0: Stop Counting
1: Start Counting

T1ST (Timer 1 Start Control)
0: stop counting
1: clear the counter and start count again

****The counter will be cleared and restarted only when the TxST bit cleared and set again. If TxST bit set again when TxST bit is set, the counter can't be cleared but only start again.***

Figure 12-1 Timer0,1 Registers

CDR0 (Input Capture Register)
T0 (Timer 0 Counter Register)

Bit :	7	6	5	4	3	2	1	0	
	R	R	R	R	R	R	R	R	
	CDR07	CDR06	CDR05	CDR04	CDR03	CDR02	CDR01	CDR00	ADDRESS: $E1_H$ INITIAL VALUE:00H

In Timer mode, this register is the value of Timer 0 counter and in Capture mode, this register is the value of input capture.

TDR0 (Timer 0 Data Register)

Bit :	7	6	5	4	3	2	1	0	
	W	W	W	W	W	W	W	W	
	TDR07	TDR06	TDR05	TDR04	TDR03	TDR02	TDR01	TDR00	ADDRESS: $0E1_H$ INITIAL VALUE:FF_H

If the counter of Timer 0 and the data of TDR0 is equal, interrupt is occurred.

CDR1 (Input Capture Register)
T1 (Timer 1 Counter Register)

Bit :	7	6	5	4	3	2	1	0	
	R	R	R	R	R	R	R	R	
	CDR17	CDR16	CDR15	CDR14	CDR13	CDR12	CDR11	CDR10	ADDRESS: $0E4_H$ INITIAL VALUE:00_H

In Timer mode, this register is the value of Timer 1 counter and in Capture mode, this register is the value of input capture.

TDR1 (Timer 1 Data Register)

Bit :	7	6	5	4	3	2	1	0	
	W	W	W	W	W	W	W	W	
	TDR17	TDR16	TDR15	TDR14	TDR13	TDR12	TDR11	TDR10	ADDRESS: $0E3_H$ INITIAL VALUE:FF_H

If the counter of Timer 1 and the data of TDR1 is equal, interrupt is occurred.

T1PPR (Timer 1 Pulse Period Register)

Bit :	7	6	5	4	3	2	1	0	
	W	W	W	W	W	W	W	W	
	T1PPR7	T1PPR6	T1PPR5	T1PPR4	T1PPR3	T1PPR2	T1PPR1	T1PPR0	ADDRESS: $0E3_H$ INITIAL VALUE:FF_H

The period is decided by PWM.

T1PDR (Timer 1 Pulse Duty Register)

Bit :	7	6	5	4	3	2	1	0	
	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R	
	T1PDR7	T1PDR6	T1PDR5	T1PDR4	T1PDR3	T1PDR2	T1PDR1	T1PDR0	ADDRESS: $0E4_H$ INITIAL VALUE:00_H

In PWM mode, decide the pulse duty.

PWMHR (PWM High Register)

Bit :	7	6	5	4	3	2	1	0	
					W	W	W	W	
	-	-	-	-	PWM03	PWM02	PWM01	PWM00	ADDRESS: $0E5_H$ INITIAL VALUE:----0000_B
	Reserved								

PWM Period = [PWMHR[3:2] + T1PPR] x Source Clock
PWM Duty = [PWMHR[1:0] + T1PDR] x Source Clock

Figure 12-2 Related Registers with Timer/Counter

16BIT	CAP0	CAP1	PWME	T0CK[2:0]	T1CK[1:0]	PWMO	Timer 0	Timer 1
0	0	0	0	XXX	XX	0	8 Bit Timer	8 Bit Timer
0	0	1	0	111	XX	0	8 Bit Event Counter	8 Bit Capture
0	1	0	0	XXX	XX	1	8 Bit Capture	8 Bit Compare Output
0	0	0	1	XXX	XX	1	8 Bit Timer/Counter	10 Bit PWM
1	0	0	0	XXX	11	0	16 Bit Timer	
1	0	0	0	111	11	0	16 Bit Event Counter	
1	1	X[1]	0	XXX	11	0	16 Bit Capture	
1	0	0	0	XXX	11	1	16 Bit Compare Output	

1. *X: The value “0” or “1” corresponding your operation.*

Table 12-1 Operating Modes of Timer 0 and Timer 1

12.1 8-Bit Timer/Counter Mode

The GMS81C5108 has two 8-bit Timer/Counters, Timer 0, Timer 1, as shown in Figure 12-3.

The “timer” or “counter” function is selected by mode registers TMx as shown in Figure 12-1 and Table 12-1. To use as an 8-bit timer/counter mode, bit CAP0 of TM0 is cleared to “0” and bits 16BIT of TM1 should be cleared to “0” (Table 12-1).

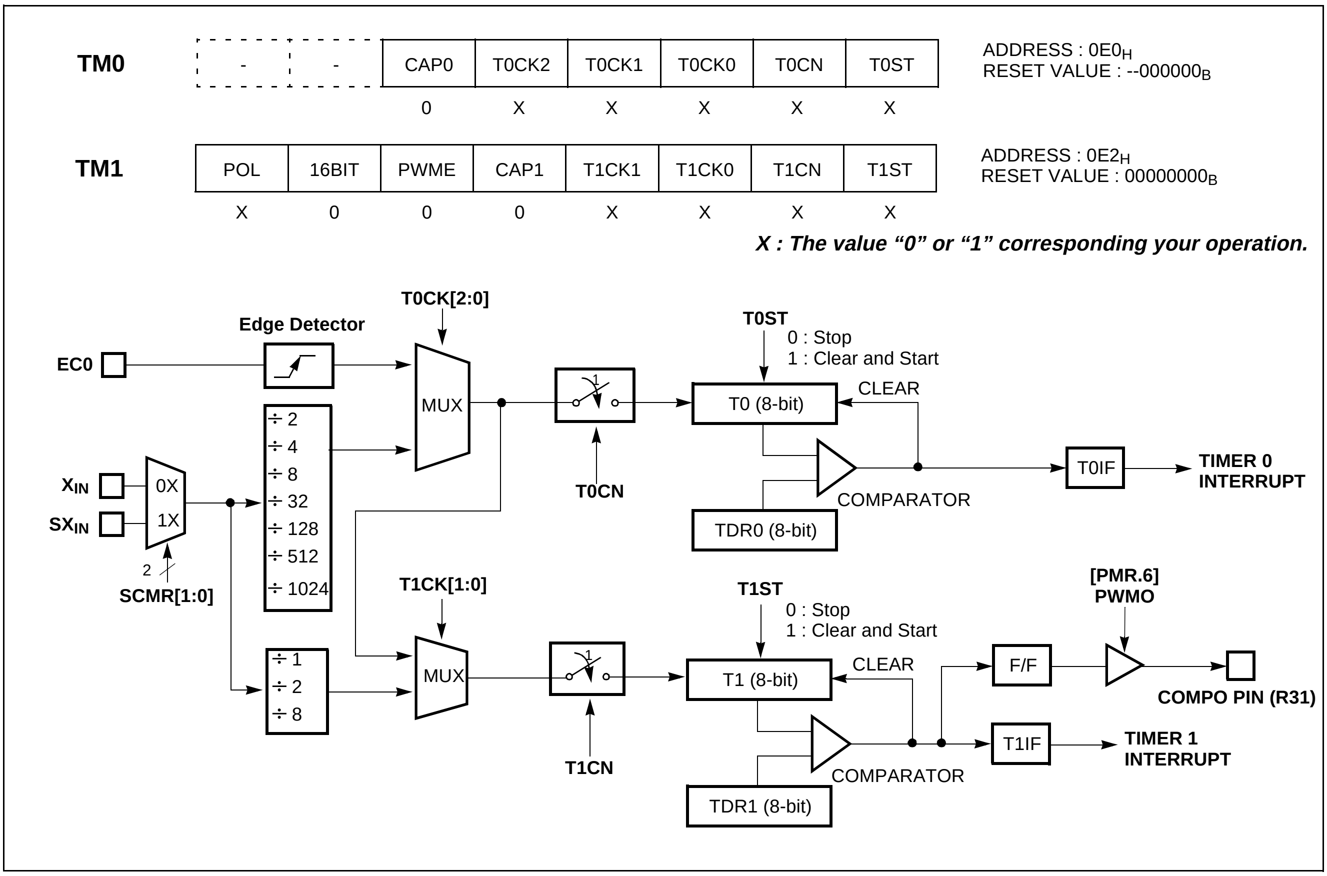


Figure 12-3 Block Diagram of Timer/Event Counter

These timers have each 8-bit count register and data register. The count register is increased by every internal or external clock input. The internal clock has a prescaler divide ratio option of 2, 4, 8, 32,128, 512, 1024 (selected by control bits T0CK2, T0CK1 and T0CK0 of register TM0) and 1, 2, 8 (selected by control bits T1CK1 and T1CK0 of register TM1).

In the Timer, timer register T_X increases from 00_H until it matches TDR_X and then reset to 00_H. If the value of T_X is equal with TDR_X, Timer $_X$ interrupt is occurred (latched in T_XIF bit). TDR0 and T0 register are in same address, so this register is read from T0 and written to TDR0.

In counter function, the counter is increased every 0-to 1 (rising edge) transition of EC0 pin. In order to use counter function, the bit R03 of the R0 Direction Register (R0DR) should be set to "0" and the bit EC0 of Port Mode Register (PMR) should set to "1". The Timer 0 can be used as a counter by pin EC0 input, but Timer 1 can not used as a counter.

Note: *The contents of TDR0 and TDR1 must be initialized (by software) with the value between 1_H and $0FF_H$, not 0_H.*

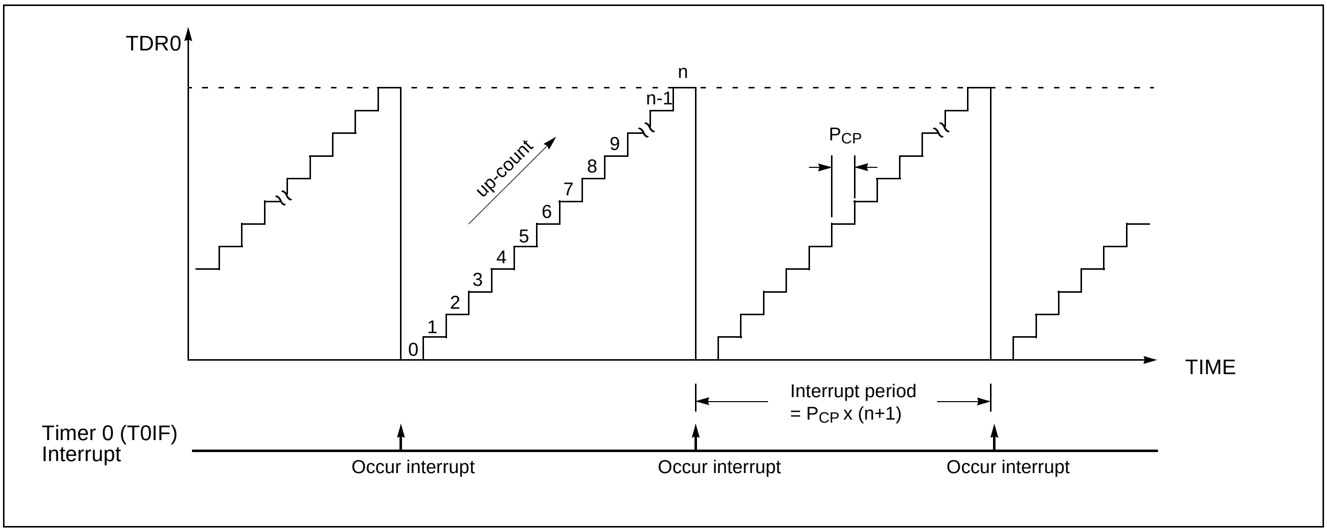


Figure 12-4 Counting Example of Timer Data Registers

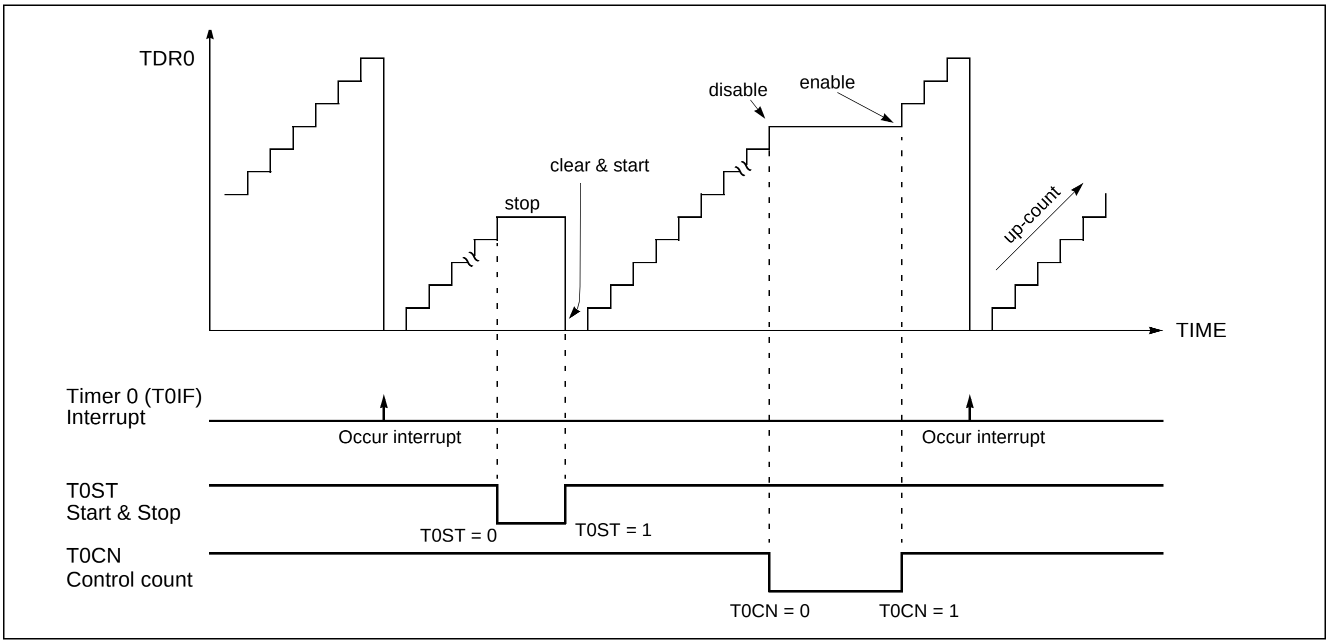


Figure 12-5 Timer Count Operation

12.2 16 Bit Timer/Counter Mode

The Timer register is running with 16 bits. A 16-bit timer/counter register T0, T1 are increased from 0000_H until it matches TDR0, TDR1 and then resets to 0000_H. The match output generates Timer 0 interrupt not Timer 1 interrupt.

The clock source of the Timer 0 is selected either internal or external clock by bit T0CK2, T0CK1 and T0CK0.

In 16-bit mode, the bits T1CK1,T1CK0 and 16BIT of TM1 should be set to “1” respectively.

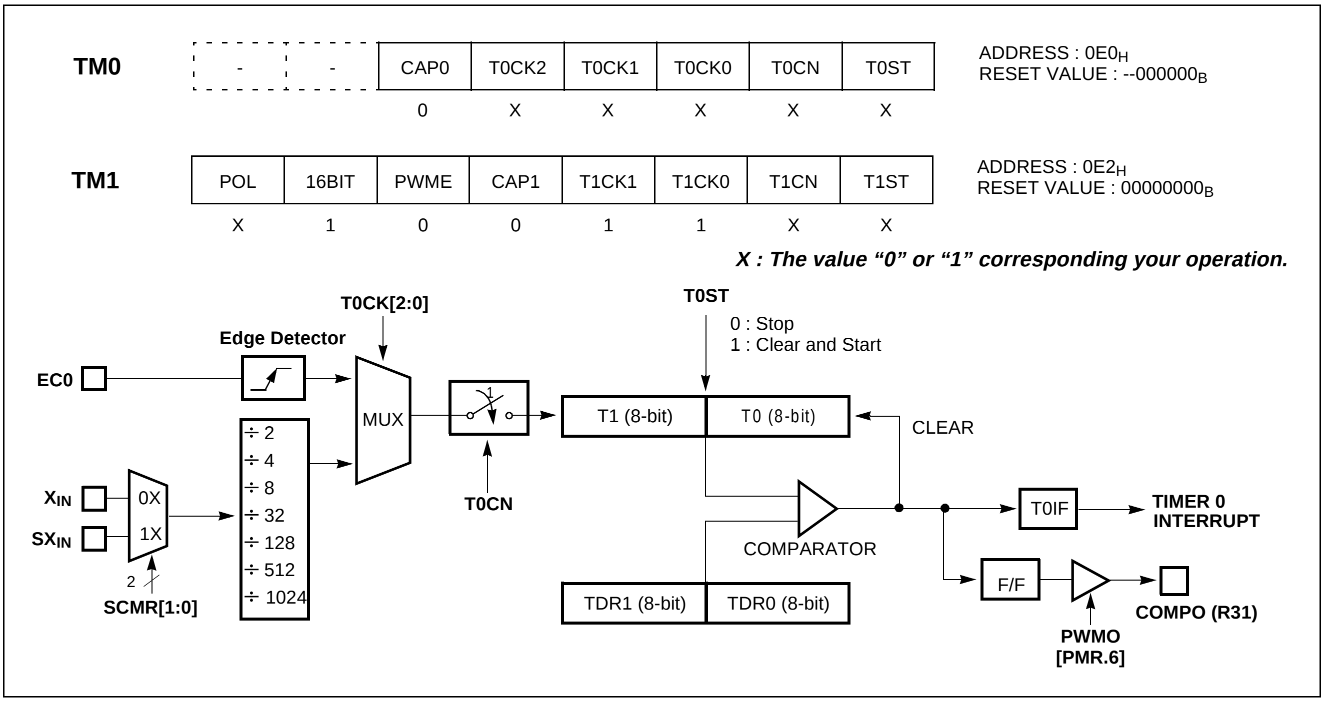


Figure 12-6 16-bit Timer / Counter Mode

12.3 8-Bit Capture Mode

The Timer 0 capture mode is set by bit CAP0 of timer mode register TM0 (bit CAP1 of timer mode register TM1 for Timer 1) as shown in Figure 12-7.

As mentioned above, not only Timer 0 but Timer 1 can also be used as a capture mode.

The Timer/Counter register is increased in response internal or external input. This counting function is same with normal timer mode, and Timer interrupt is generated when timer register T0 (T1) increases and matches TDR0 (TDR1).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is more wider than the maximum period of Timer.

For example, in Figure 12-9, the pulse width of captured signal is wider than the timer data value (FF_H) over 2 times. When external interrupt is occurred, the captured value (13_H) is more little than wanted value. It can be obtained correct value by counting the number of timer overflow occurrence.

Timer/Counter still does the above, but with the added feature that a edge transition at external input INTx pin causes the current value in the Timer x register (T0,T1), to be captured into registers CDRx (CDR0, CDR1), respectively. After captured, Timer x register is cleared and restarts by hardware.

It has three transition modes: “falling edge”, “rising edge”, “both edge” which are selected by interrupt edge selection register IESR (Refer to External interrupt section). In addition, the transition at INTx pin generate an interrupt.

Note: *The CDR0, TDR0 and T0 are in same address. In the capture mode, reading operation is read the CDR0 and in timer mode, reading operation is read the T0. TDR0 is only for writing operation.*
The CDR1, T1 are in same address, the TDR1 is located in different address. In the capture mode, reading operation is read the CDR1

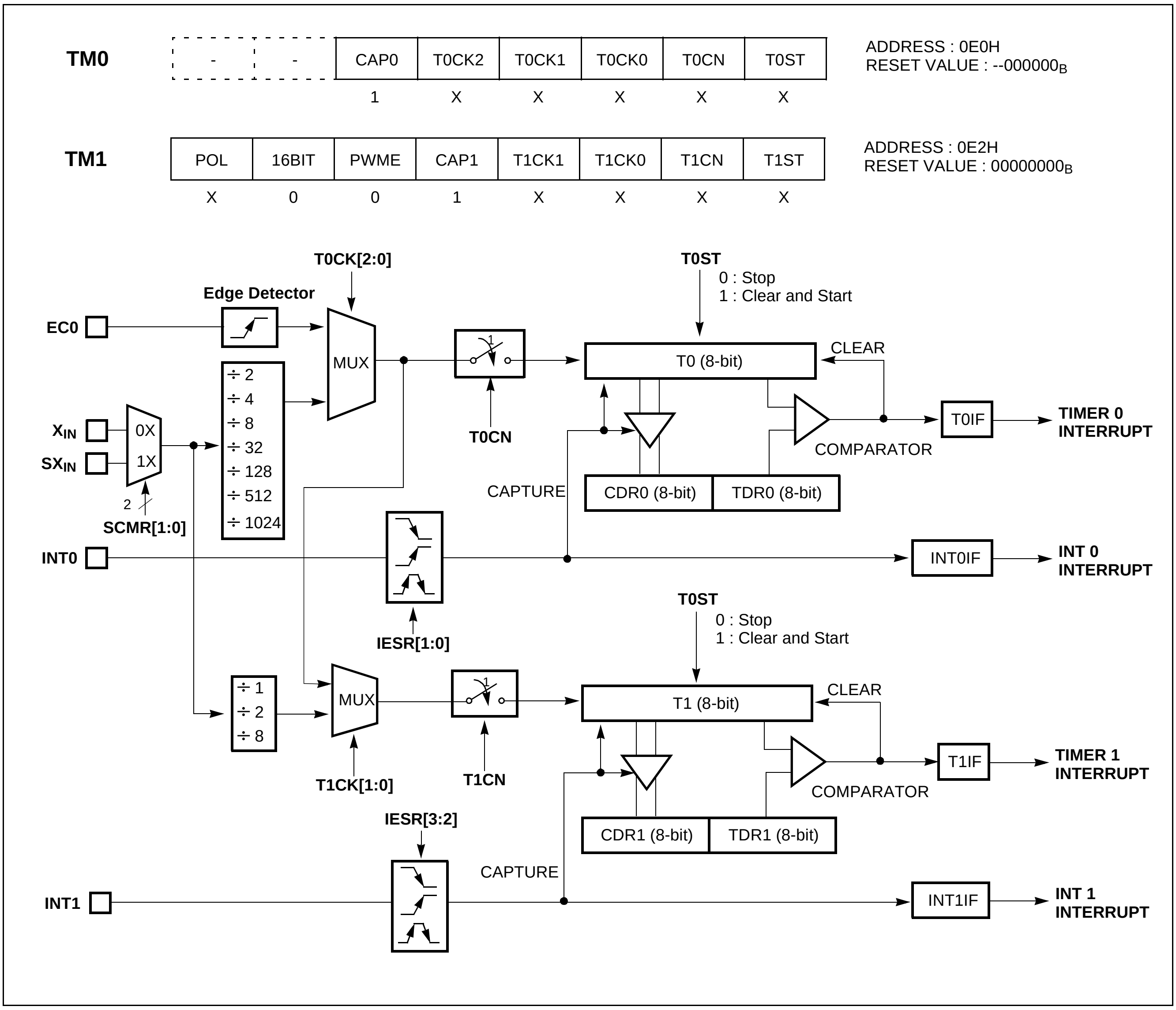


Figure 12-7 8-bit Capture Mode

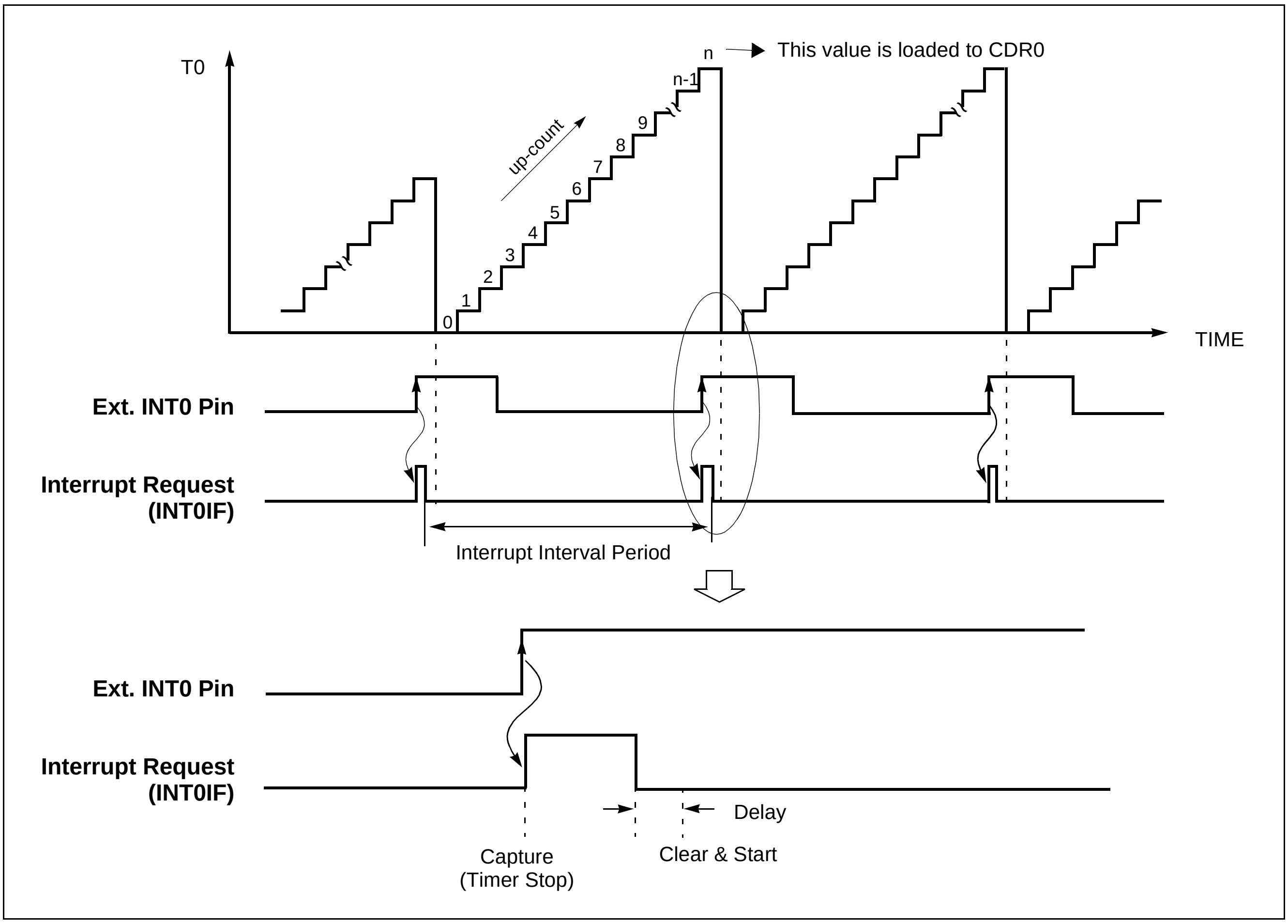


Figure 12-8 Input Capture Operation

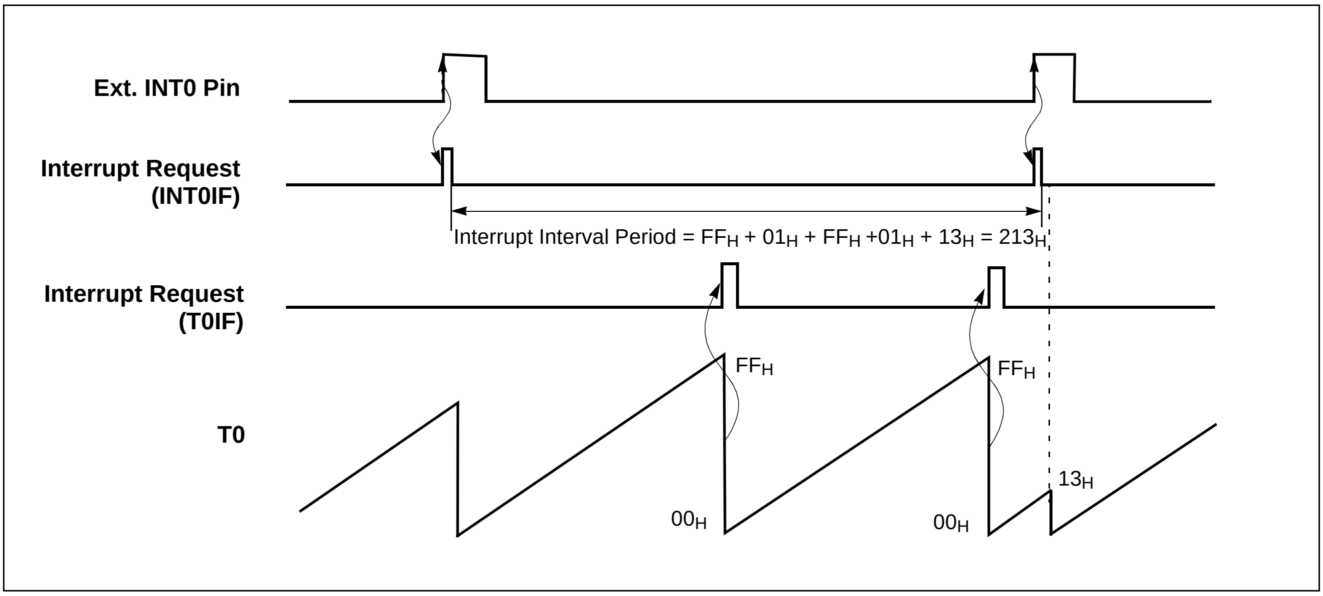


Figure 12-9 Excess Timer Overflow in Capture Mode

12.4 16-bit Capture Mode

16-bit capture mode is the same as 8-bit capture, except that the Timer register is running with 16 bits.

The clock source of the Timer 0 is selected either internal or external clock by bit T0CK2, T0CK1 and T0CK0.

In 16-bit mode, the bits T1CK1,T1CK0 and 16BIT of TM1 should be set to “1” respectively.

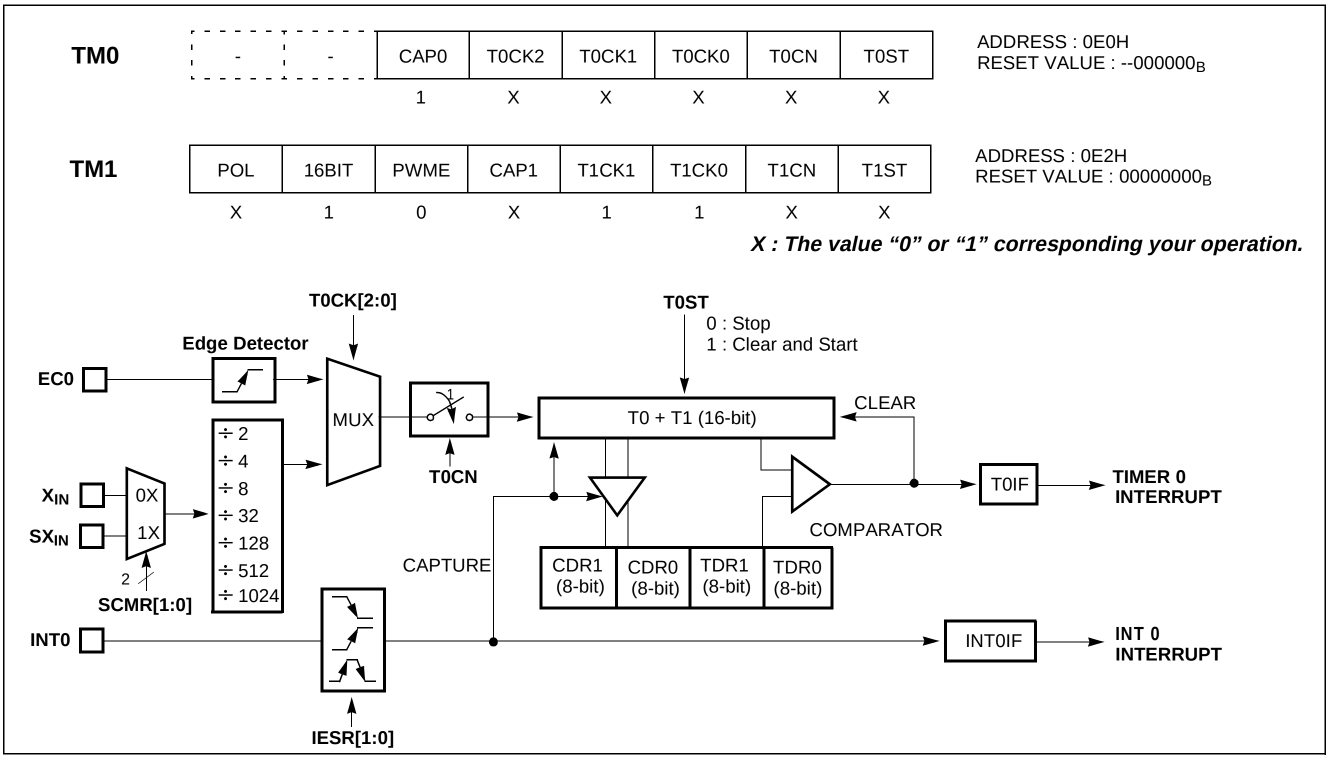


Figure 12-10 16-bit Capture Mode

12.5 8-Bit (16-Bit) Compare OutPut Mode

The GMS81C5108 has a function of Timer Compare Output. To pulse out, the timer match can goes to port pin (R31) as shown in Figure 12-3 and Figure 12-6. Thus, pulse out is generated by the timer match. These operation is implemented to pin, R31/PWM.

In this mode, the bit PWMO of Port Mode Register (PMR) should be set to “1”, and the bit PWME of Timer1 Mode Register (TM1) should be cleared to “0”.

In addition, 16-bit Compare output mode is available, also.

This pin output the signal having a 50 : 50 duty square wave, and output frequency is same as below equation.

$$f_{COMP} = \frac{\text{Oscillation Frequency}}{2 \times \text{Prescaler Value} \times (TDR + 1)}$$

12.6 PWM Mode

The GMS81C5108 has one high speed PWM (Pulse Width Modulation) function which shared with Timer1.

In PWM mode, the R31/PWM pin operates as a 10-bit resolution PWM output port. For this mode, the bit PWM of Port Mode Register (PMR) and the bit PWME of timer1 mode register (TM1) should be set to “1” respectively.

The period of the PWM output is determined by the T1PPR (PWM Period Register) and PWMHR[3:2] (bit3,2 of PWM High Register) and the duty of the PWM output is determined by the T1PDR (PWM Duty Register) and PWMHR[1:0] (bit1,0 of PWM High Register).

The user can use PWM data by writing the lower 8-bit period value to the T1PPR and the higher 2-bit period value to the PWMHR[3:2]. And the duty value can be used with the T1PDR and the PWMHR[1:0] in the same way.

The T1PDR is configured as a double buffering for glitchless PWM output. In Figure 12-11, the duty data is transferred from the master to the slave when the period data matched to the counted value. (i.e. at the beginning of next duty cycle).

The bit POL0 of TM1 decides the polarity of duty cycle.

The duty value can be changed when the PWM outputs. However the changed duty value is output after the current period is over. And it can be maintained the duty value at present output when changed only period value shown as Figure 12-13. As it were, the absolute duty time is not changed in varying frequency.

Note: *If the user need to change mode from the Timer1 mode to the PWM mode, the Timer1 should be stopped firstly, and then set period and duty register value. If user writes register values and changes mode to PWM mode while Timer1 is in operation, the PWM data would be different from expected data in the beginning.*

The relation of frequency and resolution is in inverse proportion. Table 12-2 shows the relation of PWM frequency vs. resolution.

PWM Period = [PWMHR[3:2]T1PPR+1] X Source Clock

PWM Duty = [PWMHR[1:0]T1PDR+1] X Source Clock

If it needed more higher frequency of PWM, it should be reduced resolution.

Note: *If the duty value and the period value are same, the PWM output is determined by the bit POL0 (1: High, 0: Low). And if the duty value is set to "00$_H$", the PWM output is determined by the bit POL0(1: Low, 0: High). The period value must be same or more than the duty value, and 00$_H$ cannot be used as the period value.*

Resolution	Frequency		
	T1CK[1:0] =00 (250nS)	T1CK[1:0] =01 (500nS)	T1CK[1:0] =10 (2uS)
10-bit	3.9KHz	1.95KHz	0.49KHZ
9-bit	7.8KHz	3.9KHz	0.98KHZ
8-bit	15.6KHz	7.8KHz	1.95KHz
7-bit	31.25KHz	15.6KHz	3.90KHz

Table 12-2 PWM Frequency vs. Resolution at 4MHz

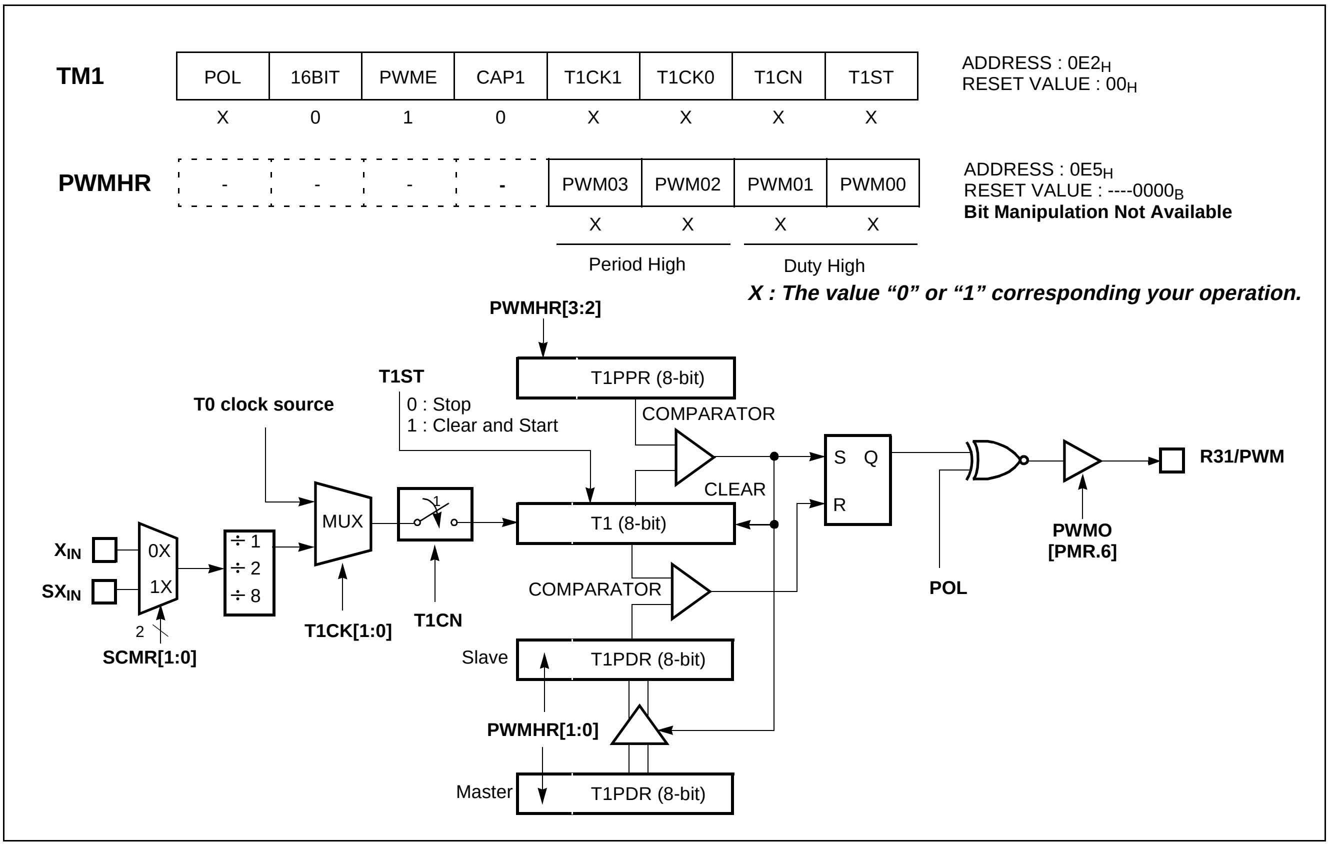


Figure 12-11 PWM Mode

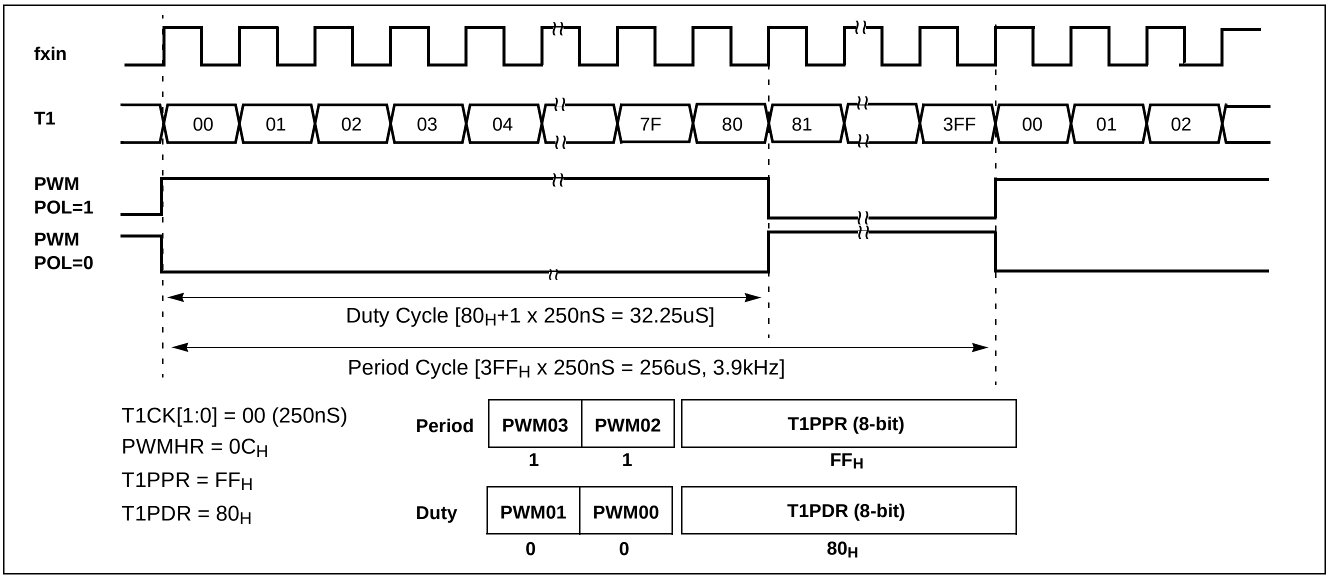


Figure 12-12 Example of PWM at 4MHz

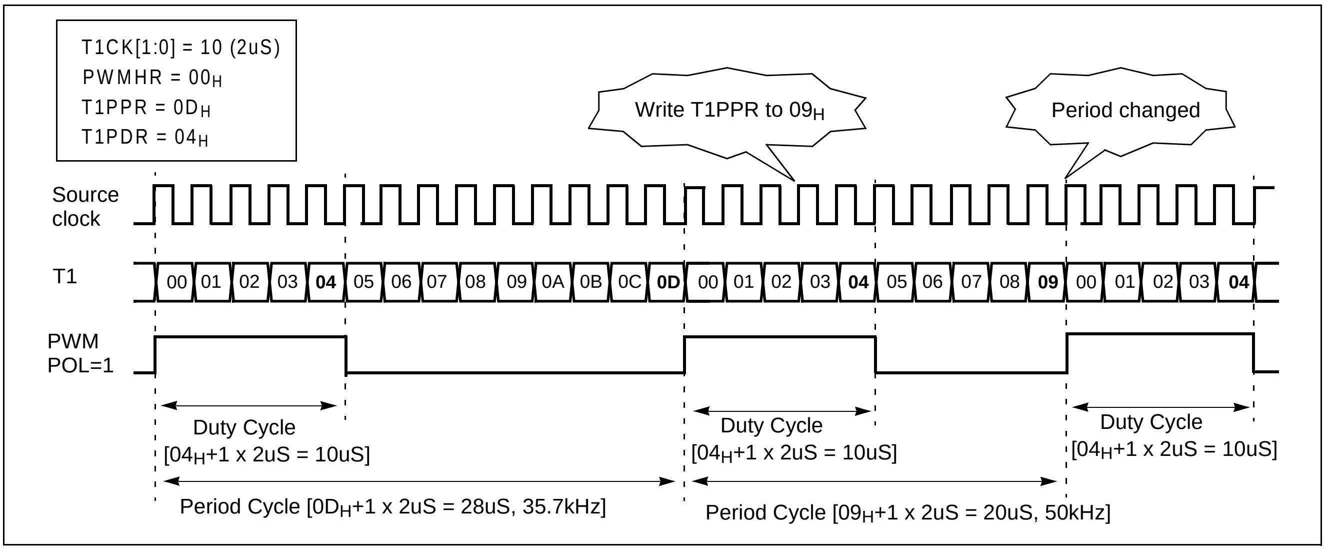


Figure 12-13 Example of Changing the Period in Absolute Duty Cycle (@4MHz)

Example:

```
Timer1 @4Mhz, 4kHz - 20% duty PWM mode

    LDM   R3DR,#0000_XX1XB   ;R31 output
    LDM   TM1,#0010_0000B    ;pwm enable
    LDM   T1PWHR,#0000_1100B ;20% duty
    LDM   T1PPR,#1110_0111B  ;period 250uS
    LDM   T1PDR,#1100_0111B  ;duty 50uS
    LDM   RSR,#X1XX_XXXXB    ;set pwm port.
    LDM   TM1,#0010_0011B    ;timer1 start
```

X means don't care

13. Watch Timer/Watch Dog Timer

This has two functions, one is the interrupt occurrence for watch time and the other is the signal generation of WDTOUTB for watch dog.

13.1 Watch Timer

The watch timer consists of the clock selector, 21-bit binary counter and watch timer mode register. It is a multi-purpose timer. It is generally used for watch design.

The bit 1,2 of WTMR select the clock source of watch timer among sub-clock, $f_{MAIN}÷2^7$ of main-clock and f_{MAIN} of main-clock. The f_{MAIN} of main-clock is used usually for watch timer test, so generally it is not used for the clock source of watch timer. The $f_{MAIN}÷2^7$ of main-clock is used when the single clock system is organized. In $f_{MAIN}÷2^7$ clock source, if the CPU enters into stop mode, the main-clock is stopped and then watch timer is also stopped. If the sub-clock is the source clock, the watch timer count cannot be stopped. Therefore, the sub-clock does not stop but continues to oscillate even when the CPU is in the STOP mode. The timer counter consists of 21-bit binary counter and it can count to max 64 seconds at sub-clock.

The bit 2, 3 of WTMR select the interrupt request interval of watch timer among 2Hz, 4Hz, 16Hz and 1/64Hz.

WTMR (Watch Timer Mode Register)

Bit :	7	6	5	4	3	2	1	0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W
	-	WTEN	WDTEN	WDTCL	WTIN1	WTIN0	WTCK1	WTCK0

ADDRESS: $0EF_H$
INITIAL VALUE: -0000000_B

WTEN (Watch Timer Enable Bit)
0: Watch Timer Disable
1: Watch Timer Enable

WDTEN (Watch Dog Timer Enable Bit)
0: Watch Dog Timer Disable
1: Watch Dog Timer Enable

WDTCL (Watch Dog Timer Clear Bit)
0: Timer running
1: WDT Clear (Auto reset after 1 cycle)

WTIN[1:0] (Watch Timer Interrupt Interval Selection)
00: 16Hz
01: 4Hz
10: 2Hz
11: 1/64Hz

WTCK[1:0] (Watch Timer Clock Source Selection)
00: Sub. Clock (f_{SUB})
01: Main Clock ($f_{MAIN}÷2^7$)
10: Main Clock (f_{MAIN})
11: -

** When f_{SUB} = 32.768 kHz and f_{MAIN} = 4.19 MHz*

Figure 13-1 Watch Timer Mode Register

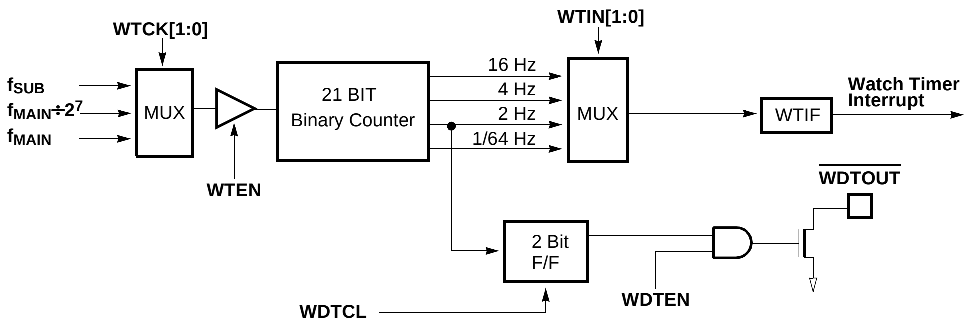


Figure 13-2 Watch Timer Block Diagram

13.2 Watch Dog Timer

The watch dog timer (WDT) function is used for checking program malfunction. If the watch dog timer is not reset in a fixed time, the WDTOUTB pin outputs a low signal. Therefore, by connecting the WDTOUTB pin and the reset pin externally, the MCU can be reset when the malfunction is occurred.

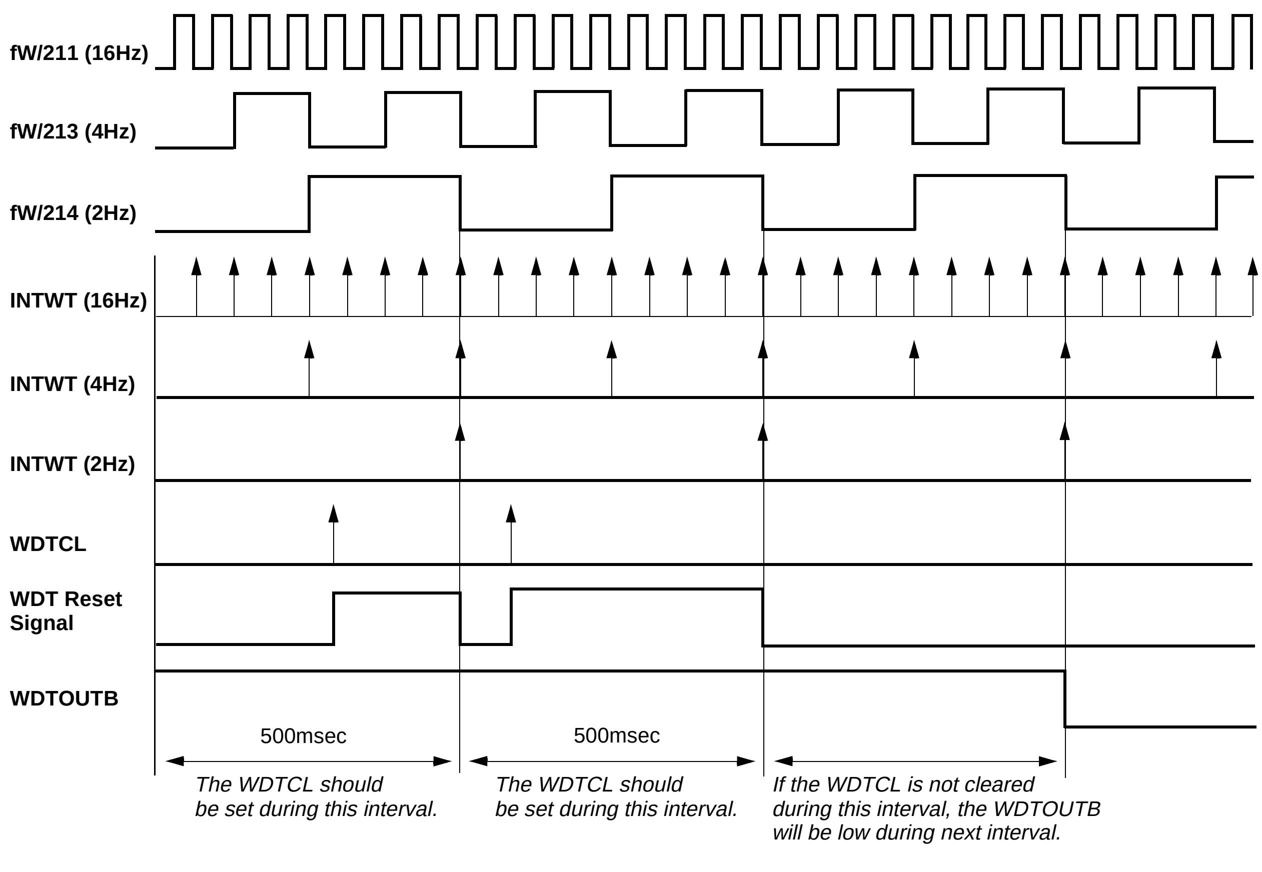
Usually the stop mode is used to reduce the power consumption. When the stop mode is released by watch timer interrupt, it is recommend to set the WDTCL to clear the 2-Bit counter and enter the stop mode. If the clock source is 1/64Hz, the WDTCL cannot be cleared in 500ms. In this case, the user should disable the WDT by clearing the WDTEN or disconnect the WDTOUTB pin and reset pin.

Usage of Watch Timer in STOP Mode

When the system is off and the watch should be kept working, follow the steps below.

1. Determines which mode is to be performed between main mode and sub mode when the MCU is released from Stop mode and set the clock source of watch timer to sub-clock.
2. Enters in STOP mode.
3. After released by watch timer interrupt, counts up timer and refreshes LCD Display. When the performing count up and refresh the LCD, the CPU operates either in main frequency mode or sub frequency mode.
4. Enters in STOP mode again.
5. Repeats 3 and 4.

When using STOP mode, if the watch timer interrupt interval is selected to 2Hz, the power consumption can be reduced considerably.

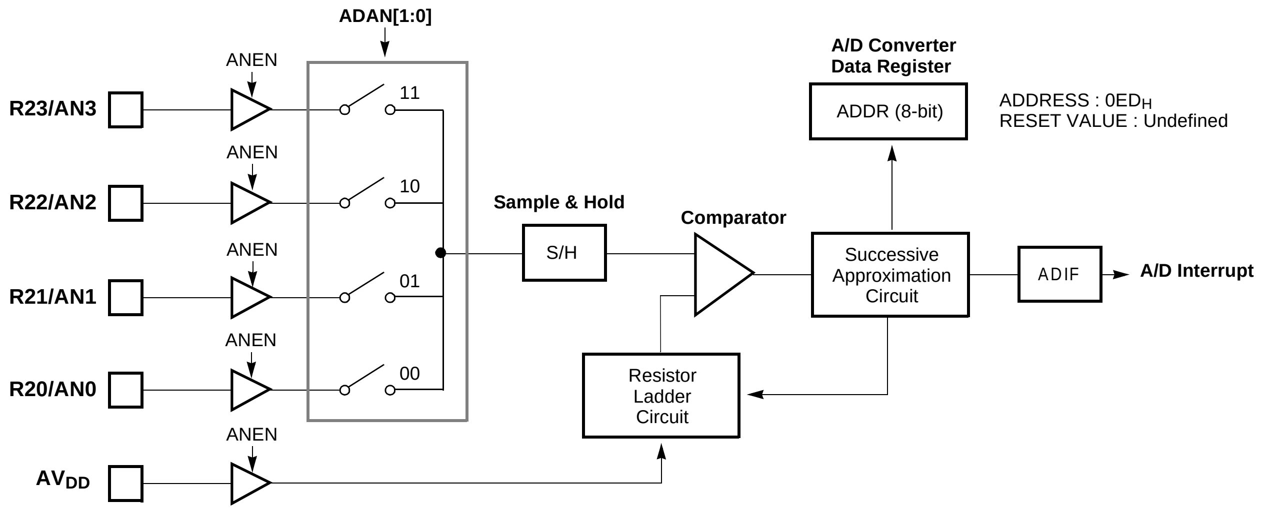


14. ANALOG TO DIGITAL CONVERTER

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a corresponding 8-bit digital value. The A/D module has four analog inputs, which are multiplexed into one sample and hold. The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog supply voltage is connected to AV_{DD} of ladder resistance of A/D module.

The A/D module has two registers which are the A/D mode register (ADMR) and A/D data register (ADDR). The ADMR register, shown in Figure 14-1, controls the operation of the A/D converter module. The port pins can be configured as analog inputs or digital I/O. To use analog inputs, each port should be assigned analog input port by setting input mode by R2DR direction register. And select the corresponding channel to be converted by setting ADAN[1:0].

The processing of conversion is start when the start bit ADST is set to “1”. After one cycle, it is cleared by hardware. The register ADDR contains the result of the A/D conversion. When the conversion is completed, the result is loaded into the ADDR, the A/D conversion status bit ADF is set to “1”, and the A/D interrupt flag ADIF is set. The block diagram of the A/D module is shown in Figure 14-1. The A/D status bit ADF is automatically set when A/D conversion is completed, cleared when A/D conversion is in process. The conversion time takes maximum 30 uS (at f_{MAIN} = 4MHz).



ADMR (A/D Mode Register)

Bit :	7	6	5	4	3	2	1	0
		R/W			R/W	R/W	R/W	R
	-	ADEN	-	-	ADAN1	ADAN0	ADST	ADF

ADDRESS : $0EC_H$
RESET VALUE : $-0--0001_B$

ADEN (A/D Converter Enable bit)
1 : Enable
0 : Disable

ADST (A/D Start bit)
1 : A/D Conversion is started
After 1 cycle, cleared to “0”
0 : Bit force to zero

ADF (A/D Status bit)
0 : A/D Conversion is in process
1 : A/D Conversion is completed

ADAN[1:0] (A/D Converter Input Selection)
00 : Channel 0 (R20/AN0)
01 : Channel 1 (R21/AN1)
10 : Channel 2 (R22/AN2)
11 : Channel 3 (R23/AN3)

ADDR (A/D Data Register)

Bit :	7	6	5	4	3	2	1	0
	R	R	R	R	R	R	R	R
	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

ADDRESS : $0ED_H$
RESET VALUE : Undefined

Figure 14-1 A/D Converter Block Diagram & Registers

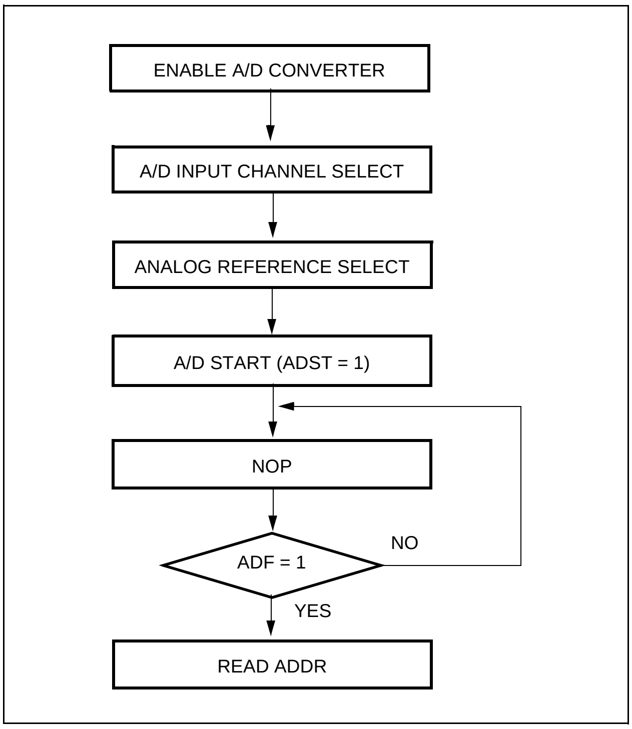


Figure 14-2 A/D Converter Operation Flow

A/D Converter Cautions

(1) Input range of AN0 to AN3

The input voltages of AN0 to AN3 should be within the specification range. In particular, if a voltage above AV_{DD} or below V_{SS} is input (even if within the absolute maximum rating range), the conversion value for that channel can not be indeterminated. The conversion values of the other channels may also be affected.

(2) Noise countermeasures

In order to maintain 8-bit resolution, attention must be paid to noise on pins AV_{DD} and AN0 to AN3. Since the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor is connected externally as shown below in order to reduce noise.

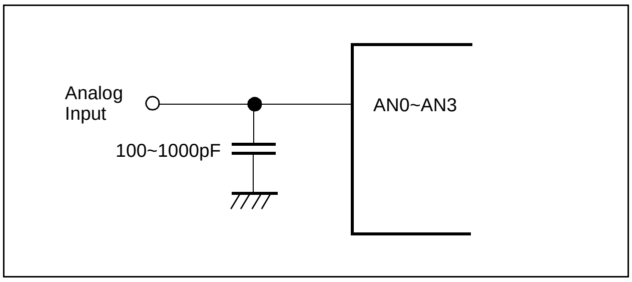


Figure 14-3 Analog Input Pin Connecting Capacitor

(3) Pins AN0/R20 to AN3/R23

The analog input pins AN0 to AN3 also function as input/output port (PORT R2) pins. When A/D conversion is performed with any of pins AN0 to AN3 selected, be sure not to execute a PORT input instruction while conversion is in progress, as this may reduce the conversion resolution.

Also, if digital pulses are applied to a pin adjacent to the pin in the process of A/D conversion, the expected A/D conversion value may not be obtainable due to coupling noise. Therefore, avoid applying pulses to pins adjacent to the pin undergoing A/D conversion.

(4) AV_{DD} pin input impedance

A series resistor string of approximately 10KΩ is connected between the AV_{DD} pin and the V_{SS} pin.

Therefore, if the output impedance of the reference voltage source is high, this will result in parallel connection to the series resistor string between the AV_{DD} pin and the V_{SS} pin, and there will be a large reference voltage error.

15. Buzzer Output Function

The buzzer driver consists of 6-bit binary counter, the buzzer data register BDR and the clock selector. It generates square-wave which is very wide range frequency (500 Hz~125 KHz at f_{MAIN} = 4MHz) by user programmable counter.

Pin R04 is assigned for output port of Buzzer driver by setting the bit BUZ of Port Mode Register (PMR) to "1".

The 6-bit buzzer counter is cleared and start the counting by writing signal to the register BDR. It is increased from 00_H until it matches with BDR[5:0].

Also, it is cleared by counter overflow and count up to output the square wave pulse of duty 50%.

The bit 0 to 5 of BDR determines output frequency for buzzer driving. BCD is undefined after reset, so it must be initialized to between 0_H and $3F_H$ by software. Note that BDR is a write-only register. Frequency calculation is following as shown below.

$$f_{BUZ}(Hz) = \frac{\text{Oscillator Frequency}}{2 \times \text{Prescaler Ratio} \times (BCD + 1)}$$

The bits BCK1, BCK0 of BDR select the source clock from prescaler output

f_{BUZ}: BUZ pin frequency
Prescaler ratio: Prescaler divide ratio by BDR[7:6]
BCD value: 6-bit compare data, BCD[5:0].

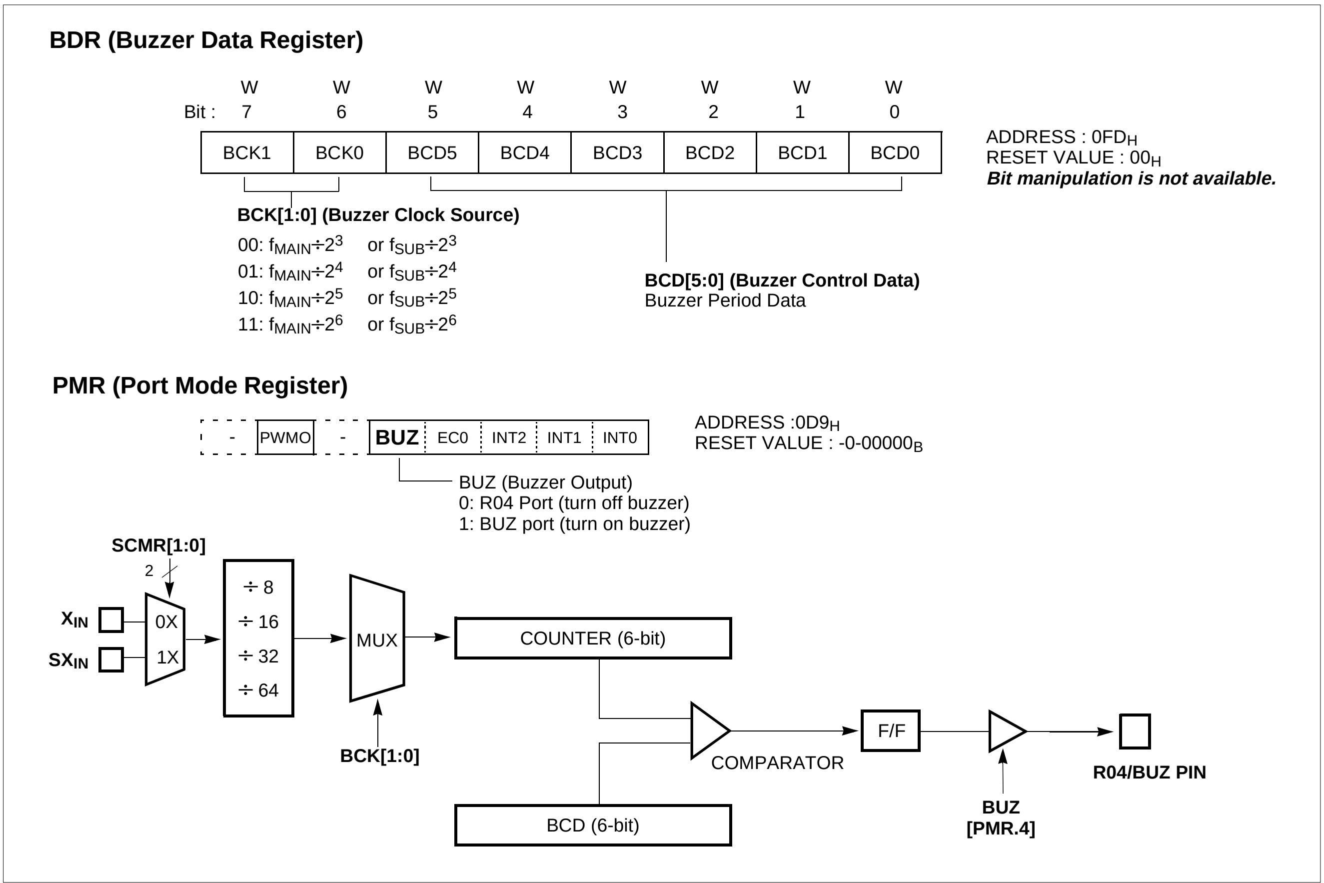


Figure 15-1 Buzzer Driver

Example: 2.5kHz output at 4MHz.

```
LDM    R0DR,#XXX1_XXXXB
LDM    BDR,#1001_1000B
LDM    PMR,#XXX1_XXXXB   ;Buzzer ON
```

X means don't care

Buzzer Output Frequency

When main-frequency is 4MHz, buzzer frequency is shown as below and if sub-frequency is selected as clock source, buzzer frequency is used after dividing by 128.

BDR [5:0]	Frequency Output (kHz)			
	00	01	10	11
00	250.000	125.000	62.500	31.250
01	125.000	62.500	31.250	15.625
02	83.333	41.667	20.833	10.417
03	62.500	31.250	15.625	7.813
04	50.000	25.000	12.500	6.250
05	41.667	20.833	10.417	5.208
06	35.714	17.857	8.929	4.464
07	31.250	15.625	7.813	3.906
08	27.778	13.889	6.944	3.472
09	25.000	12.500	6.250	3.125
0A	22.727	11.364	5.682	2.841
0B	20.833	10.417	5.208	2.604
0C	19.231	9.615	4.808	2.404
0D	17.857	8.929	4.464	2.232
0E	16.667	8.333	4.167	2.083
0F	15.625	7.813	3.906	1.953
10	14.706	7.353	3.676	1.838
11	13.889	6.944	3.472	1.736
12	13.158	6.579	3.289	1.645
13	12.500	6.250	3.125	1.563
14	11.905	5.952	2.976	1.488
15	11.364	5.682	2.841	1.420
16	10.870	5.435	2.717	1.359
17	10.417	5.208	2.604	1.302
18	10.000	5.000	2.500	1.250
19	9.615	4.808	2.404	1.202
1A	9.259	4.630	2.315	1.157
1B	8.929	4.464	2.232	1.116
1C	8.621	4.310	2.155	1.078
1D	8.333	4.167	2.083	1.042
1E	8.065	4.032	2.016	1.008
1F	7.813	3.906	1.953	0.977

BDR [5:0]	Frequency Output (kHz)			
	00	01	10	11
20	7.576	3.788	1.894	0.947
21	7.353	3.676	1.838	0.919
22	7.143	3.571	1.786	0.893
23	6.944	3.472	1.736	0.868
24	6.757	3.378	1.689	0.845
25	6.579	3.289	1.645	0.822
26	6.410	3.205	1.603	0.801
27	6.250	3.125	1.563	0.781
28	6.098	3.049	1.524	0.762
29	5.952	2.976	1.488	0.744
2A	5.814	2.907	1.453	0.727
2B	5.682	2.841	1.420	0.710
2C	5.556	2.778	1.389	0.694
2D	5.435	2.717	1.359	0.679
2E	5.319	2.660	1.330	0.665
2F	5.208	2.604	1.302	0.651
30	5.102	2.551	1.276	0.638
31	5.000	2.500	1.250	0.625
32	4.902	2.451	1.225	0.613
33	4.808	2.404	1.202	0.601
34	4.717	2.358	1.179	0.590
35	4.630	2.315	1.157	0.579
36	4.545	2.273	1.136	0.568
37	4.464	2.232	1.116	0.558
38	4.386	2.193	1.096	0.548
39	4.310	2.155	1.078	0.539
3A	4.237	2.119	1.059	0.530
3B	4.167	2.083	1.042	0.521
3C	4.098	2.049	1.025	0.512
3D	4.032	2.016	1.008	0.504
3E	3.968	1.984	0.992	0.496
3F	3.906	1.953	0.977	0.488

Table 15-1 Buzzer Output Frequency

16. Serial Communication Interface

The SCI module allows 8-bits of data to be synchronously transmitted and received. This is useful for communication with other peripheral of microcontroller devices.This consists of serial I/O data register, serial I/O mode register, clock selection circuit octal counter and control circuit as shown in Figure 16-1.

SIOM (Seriol I/O Mode Register)

Bit :	7	6	5	4	3	2	1	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	POL	MSBS	SIO1	SIO0	SICK1	SICK0	SIOST	SIOSF

ADDRESS : $0FE_H$
RESET VALUE : 00000001_B

POL (Polarity Selection)
0 : Data Transmission at falling edge
(Received data latch at rising edge)
1 : Data Transmission at rising edge
(Received data latch at falling edge)

SIO[1:0] (Serial I/O Operation Mode)
00 : Normal Port (R05, R06, R07)
01 : Transmit Mode (SCK, SO, R07)
10 : Receive Mode (SCK, R06, SI)
11 : Transmit & Receive Mode (SCK, SO, SI)

SIOST (Serial I/O Operation Start Control)
0 : SIO Operation Stop
1 : SIO Operation Start
(After one SCK clock become “0”)

MSBS (MSB First Transmit and Receive Selection)
0 : LSB First
1 : MSB First

SICK[1:0] (Serial I/O Clock Source Selection)
00: $f_{MAIN} \div 4$ or $f_{SUB} \div 4$
01: $f_{MAIN} \div 16$ or $f_{SUB} \div 16$
10: T0O (Timer 0 Output)
11: External Clock

SIOSF (Serial I/O Status Flag)
0 : During SIO Operation
1 : SIO Operation Finished

SIOD (Serial I/O Data Register)

Bit :	7	6	5	4	3	2	1	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	SIOD7	SIOD6	SIOD5	SIOD4	SIOD3	SIOD2	SIOD1	SIOD0

ADDRESS : $0FF_H$
RESET VALUE : Undefined

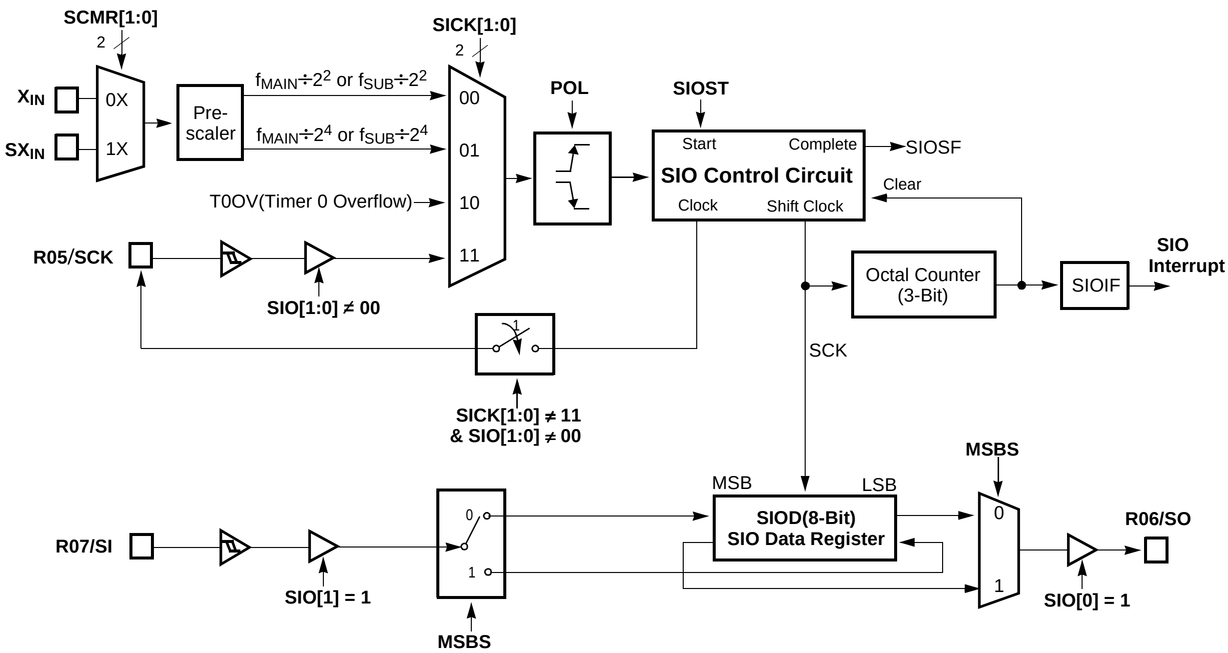


Figure 16-1 SCI Registers and Block Diagram

To accomplish communication, typically three pins are used:

- Serial Data In R07/SI
- Serial Data Out R06/SO
- Serial Clock R05/SCK

The serial data transfer operation mode is decided by setting the SIO1 and SIO0 and the transfer clock rate is decided by setting the SICK1 and SICK0 of SCI Mode Control Register as shown in Figure 16-1. And the polarity of transfer clock is selected by setting the POL. The MSBS bit is used to select which bit would be sending or receiving.

SIO1	SIO0	Function Selection	Port Selection		
			R05/SCK	R06/SO	R07/SI
0	0	-	R05	R06	R07
0	1	Transmit Mode	SCK	SO	R07
1	0	Receive Mode	SCK	R06	SI
1	1	Transmit and Receive	SCK	SO	SI

16.1 Data Transmit/Receive Timing

The SCI operation is executed by setting the SIOST bit to "1". The SIOST bit is cleared to "0" automatically after 1 machine cycle. The Serial output data is shift in or shift out at edge decided by POL. Interrupt is occurred when the eight in/out datas is counted by octal counter.

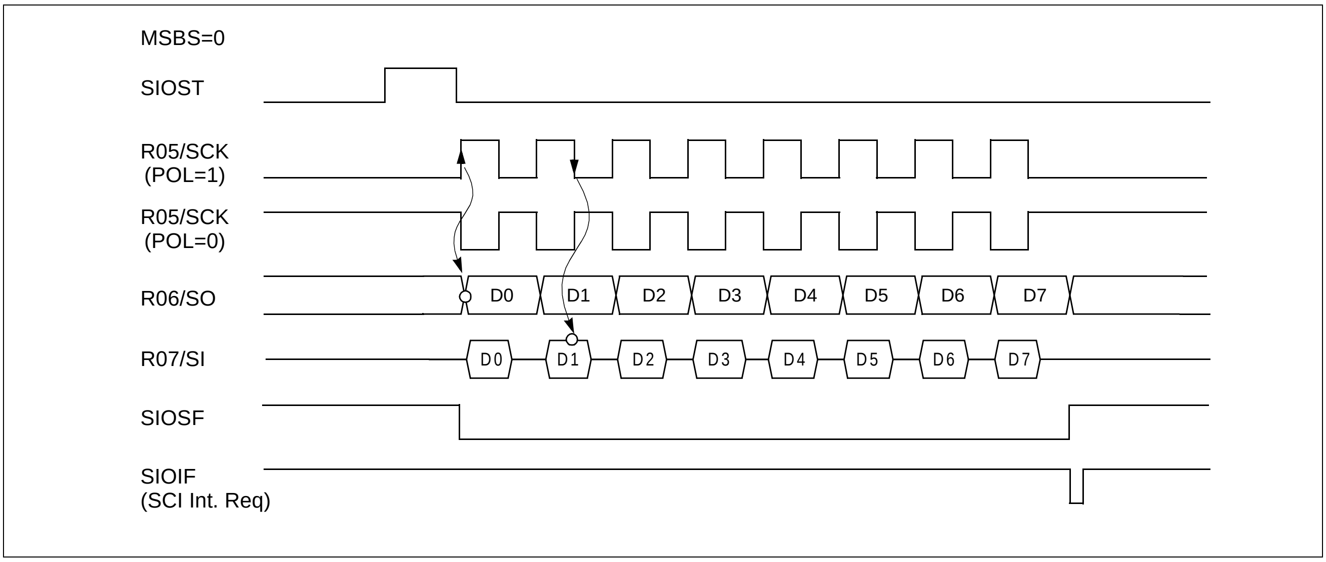


Figure 16-2 SCI Timing Diagram

16.2 The method of Serial I/O

1. Select transmission/receiving mode

When external clock is used, the frequency should be less than 1MHz and recommended duty is 50%.

2. In case of sending mode, write data to be send to SIOD.

3. Set SIOST to “1” to start serial transmission.

If both transmission mode is selected and transmission is performed simultaneously it would be made error.

4. The SIO interrupt is generated at the completion of SIO and SIOSF is set to “1”.

5. In case of receiving mode, the received data is acquired by reading the SIOD.

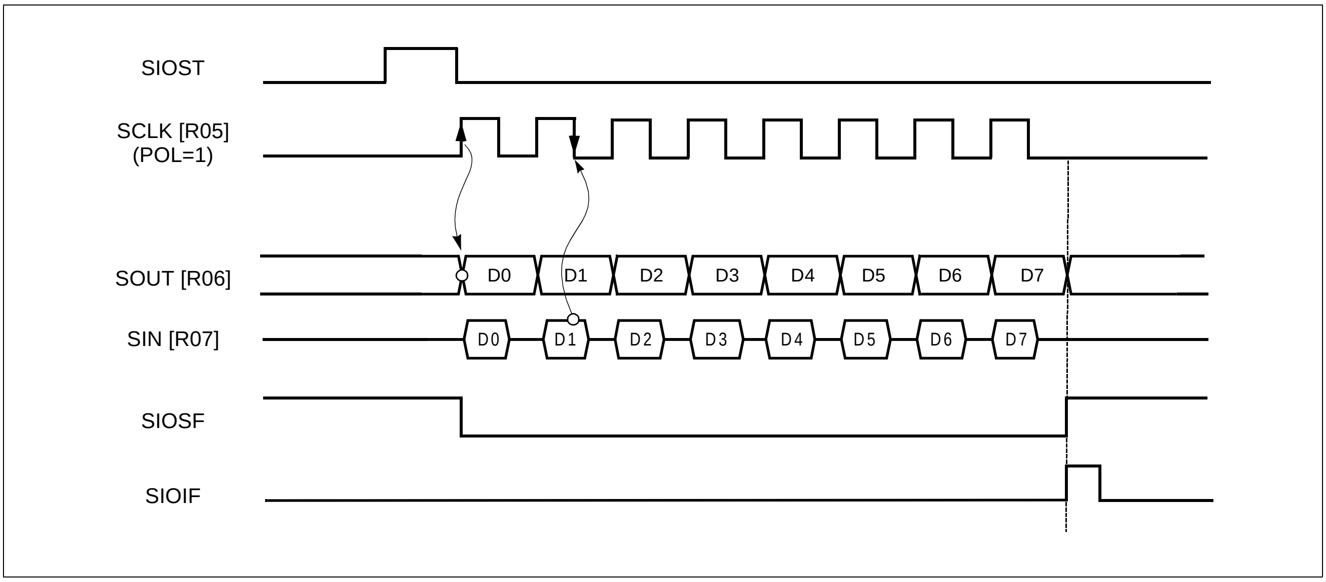


Figure 16-3 SCI Timing Diagram at POL=1

17. INTERRUPTS

The GMS81C5108 interrupt circuits consist of Interrupt enable register (IENH, IENL), Interrupt request flag (IRQH, IRQL), Interrupt Edge Selection Register (IESR), priority circuit and Master enable flag ("I" flag of PSW). The configuration of interrupt circuit is shown in Figure 17-1 and Interrupt priority is shown in Table 17-1 .

The flags that actually generate these interrupts are bit INT0F, INT1F and INT2F in Register IRQH. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated.

The Timer 0 and Timer 2 Interrupts are generated by T0IF and T1IF, which are set by a match in their respective timer/counter register. The AD converter Interrupt is generated by ADIF which is set by finishing the analog to digital conversion. The Basic Interval Timer Interrupt is generated by BITIF which is set by overflow of the Basic Interval Timer Register (BITR).

Reset/Interrupt	Symbol	Priority	Vector Addr.
Hardware Reset	RESET	-	$FFFE_H$
Key Scan Interrupt	KS	1	$FFFC_H$
BIT Interrupt	BIT	2	$FFFA_H$
External Interrupt 0	INT0	3	$FFF8_H$
External Interrupt 1	INT1	4	$FFF6_H$
Timer 0 Interrupt	T0	5	$FFF4_H$
Timer 1 Interrupt	T1	6	$FFF2_H$
External Interrupt 2	INT2	7	$FFF0_H$
Remocon Interrupt	REM	8	$FFEE_H$
AD Interrupt	AD	9	$FFEC_H$
SIO Interrupt	SIO	10	$FFEA_H$
Watch Timer Interrupt	WT	11	$FFE8_H$

Table 17-1 Interrupt Priority

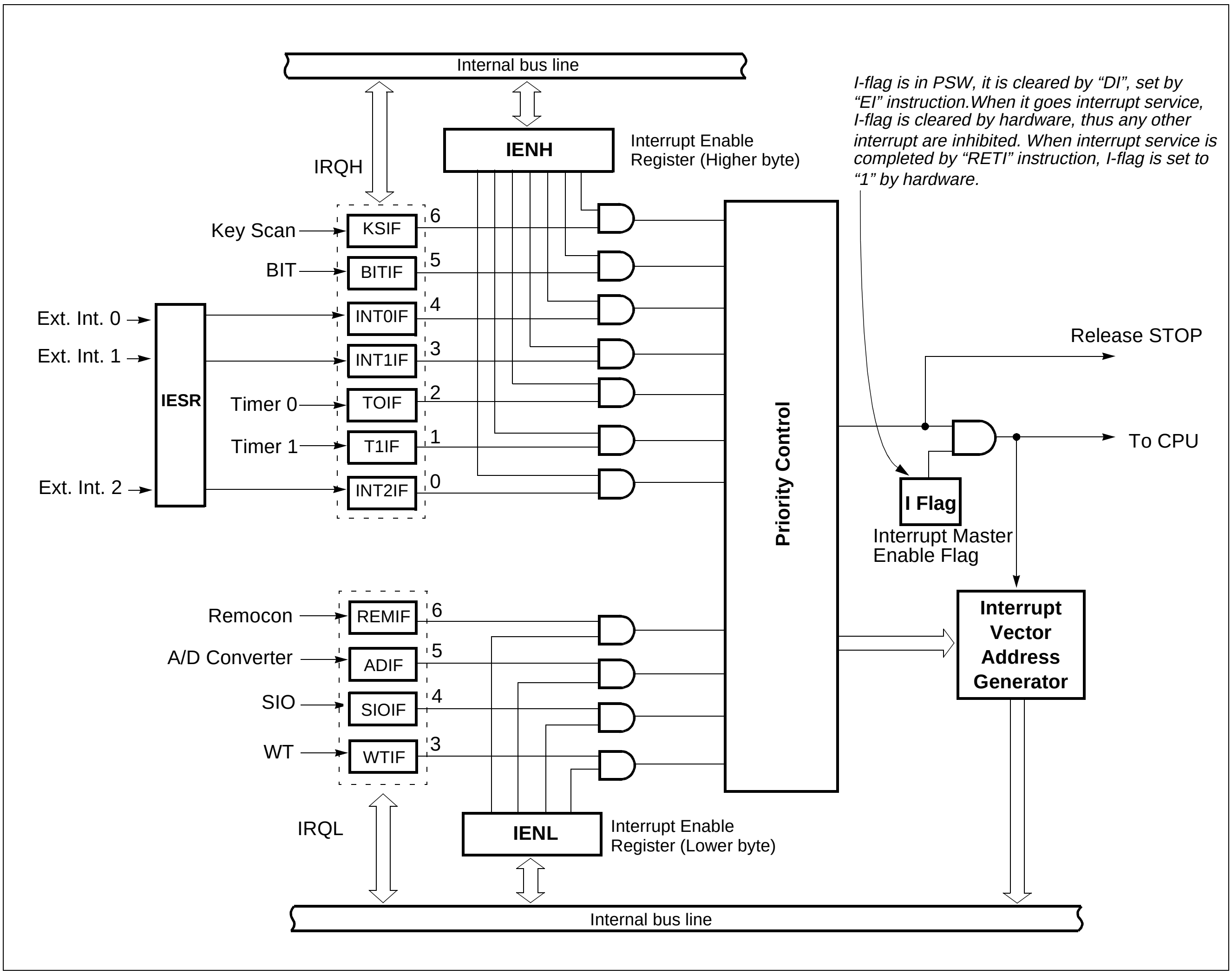


Figure 17-1 Block Diagram of Interrupt Function

The External Interrupts INT0, INT1 and INT2 can each be transition-activated (1-to-0, 0-to-1 and both transiton).The interrupts are controlled by the interrupt master enable flag I-flag (bit 2 of PSW), the interrupt enable register (IENH, IENL) and the interrupt request flag (IRQH, IRQL) except Power-on reset and software BRK interrupt.

Interrupt enable registers are shown in Figure 17-2. These registers are composed of interrupt enable flags of each interrupt source, these flags determine whether an interrupt will be accepted or not. When enable flag is “0”, a corresponding interrupt source is prohibited. Note that PSW contains also a master enable bit, I-flag, which disables all interrupts at once. When an interrupt is occurred, the I-flag is cleared and disable any further interrupt, the return address and PSW are pushed into the stack and the PC is vectored to. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt request flag bits.

The interrupt request flag bit(s) must be cleared by software before re-enabling interrupts to avoid recursive interrupts. The Interrupt Request flags are able to be read and written.

IENH (Interrupt Enable High Register)

Bit :	7	6	5	4	3	2	1	0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W
	-	KSE	BITE	INT0E	INT1E	T0E	T1E	INT2E

ADDRESS : $0DB_H$
RESET VALUE : -0000000_B

IENL (Interrupt Enable Low Register)

Bit :	7	6	5	4	3	2	1	0
		R/W	R/W	R/W	R/W			
	-	REME	ADE	SIOE	WTE	-	-	-

ADDRESS : $0DA_H$
RESET VALUE : $-0000---_B$

Enables or disables the interrupt individually
If flag is cleared, the interrupt is disabled.

0 : Disable
1 : Enable

IRQH (Interrupt Request High Register)

Bit :	7	6	5	4	3	2	1	0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W
	-	KSIF	BITIF	INT0IF	INT1IF	T0IF	T1IF	INT2IF

ADDRESS : $0DD_H$
RESET VALUE : -0000000_B

IRQL (Interrupt Request Low Register)

Bit :	7	6	5	4	3	2	1	0
		R/W	R/W	R/W	R/W			
	-	REMIF	ADIF	SIOIF	WTIF	-	-	-

ADDRESS : $0DC_H$
RESET VALUE : $-0000---_B$

Shows the interrupt occurrence
0 : Not occurred
1 : Interrupt request is occurred

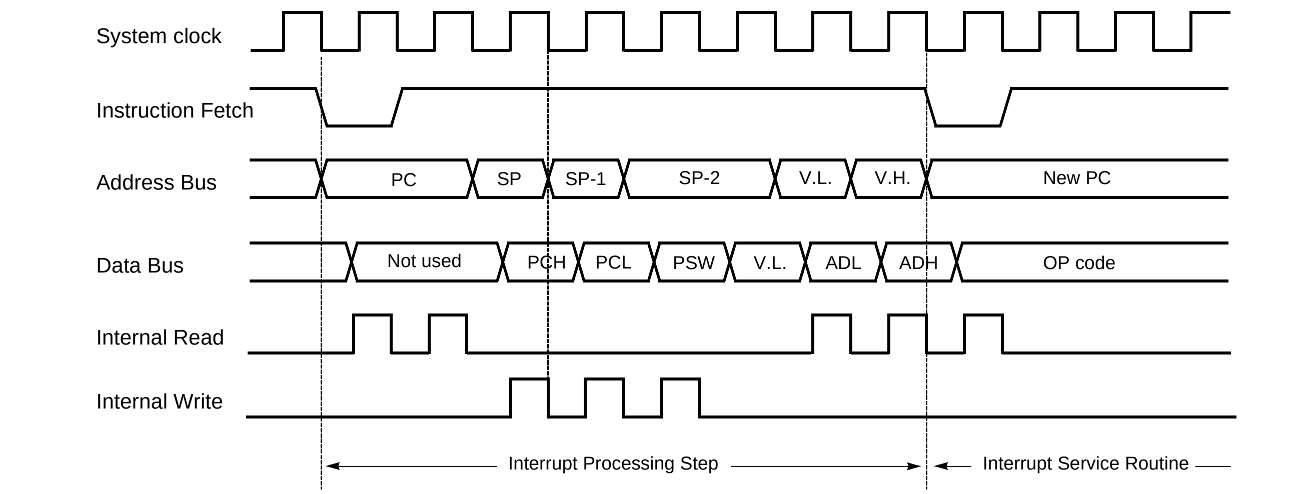
Figure 17-2 Interrupt Enable Registers and Interrupt Request Registers

17.1 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to “0” by a reset or an instruction. Interrupt acceptance sequence requires 8 f_{OSC} (2 µs at f_{MAIN}=4MHz) after the completion of the current instruction execution. The interrupt service task is terminated upon execution of an interrupt return instruction [RETI].

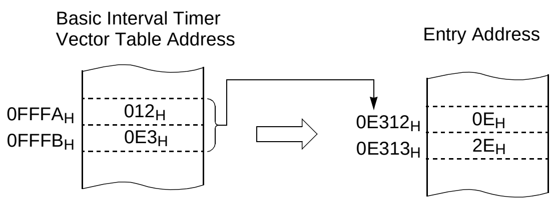
Interrupt acceptance

1. The interrupt master enable flag (I-flag) is cleared to “0” to temporarily disable the acceptance of any following maskable interrupts. When a non-maskable interrupt is accepted, the acceptance of any following interrupts is temporarily disabled.
2. Interrupt request flag for the interrupt source accepted is cleared to “0”.
3. The contents of the program counter (return address) and the program status word are saved (pushed) onto the stack area. The stack pointer decreases 3 times.
4. The entry address of the interrupt service program is read from the vector table address and the entry address is loaded to the program counter.
5. The instruction stored at the entry address of the interrupt service program is executed.



V.L. and V.H. are vector addresses.
ADL and ADH are start addresses of interrupt service routine as vector contents.

Figure 17-3 Timing chart of Interrupt Acceptance and Interrupt Return Instruction



Correspondence between vector table address for BIT interrupt and the entry address of the interrupt service program.

An interrupt request is not accepted until the I-flag is set to “1” even if a requested interrupt has higher priority than that of the current interrupt being serviced.

When nested interrupt service is required, the I-flag should be set to “1” by “EI” instruction in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

Saving/Restoring General-purpose Register

During interrupt acceptance processing, the program counter and the program status word are automatically saved on the stack, but accumulator and other registers are not saved itself. If necessary, these registers should be saved by the software. Also, when multiple interrupt services are nested, it is necessary to avoid using the same data memory area for saving registers.

The following method is used to save/restore the general-purpose registers.

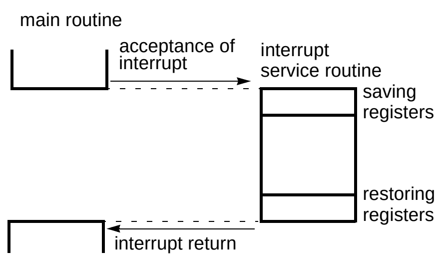
Example: Register saving

```
INTxx:  PUSH    A       ;SAVE ACC.
        PUSH    X       ;SAVE X REG.
        PUSH    Y       ;SAVE Y REG.

        interrupt processing

        POP     Y       ;RESTORE Y REG.
        POP     X       ;RESTORE X REG.
        POP     A       ;RESTORE ACC.
        RETI            ;RETURN
```

General-purpose registers are saved or restored by using push and pop instructions.



17.2 BRK Interrupt

Software interrupt can be invoked by BRK instruction, which has the lowest priority order.

Interrupt vector address of BRK is shared with the vector of TCALL 0 (Refer to Program Memory Section). When BRK interrupt is generated, B-flag of PSW is set to distinguish BRK from TCALL 0.

Each processing step is determined by B-flag as shown in Figure 17-4.

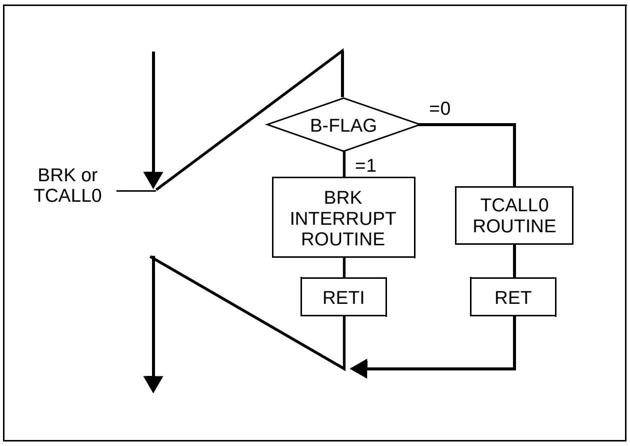


Figure 17-4 Execution of BRK/TCALL0

17.3 Multi Interrupt

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the interrupt are received at the same time simultaneously, an internal polling sequence determines by hardware which request is serviced.

However, multiple processing through software for special features is possible. Generally when an interrupt is accepted, the I-flag is cleared to disable any further interrupt. But as user sets I-flag in interrupt routine, some further interrupt can be serviced even if certain interrupt is in progress.

Example: Even though Timer1 interrupt is in progress, INT0 interrupt serviced without any suspend.

```
TIMER1: PUSH   A
        PUSH   X
        PUSH   Y
        LDM    IENH,#80H    ;Enable INT0 only
        LDM    IENL,#0      ;Disable other
        EI                  ;Enable Interrupt
        :
        :
        :


        :
        :
        :
        LDM    IENH,#0FFH   ;Enable all interrupts
        LDM    IENL,#0F0H
        POP    Y
        POP    X
        POP    A
        RETI
```

.

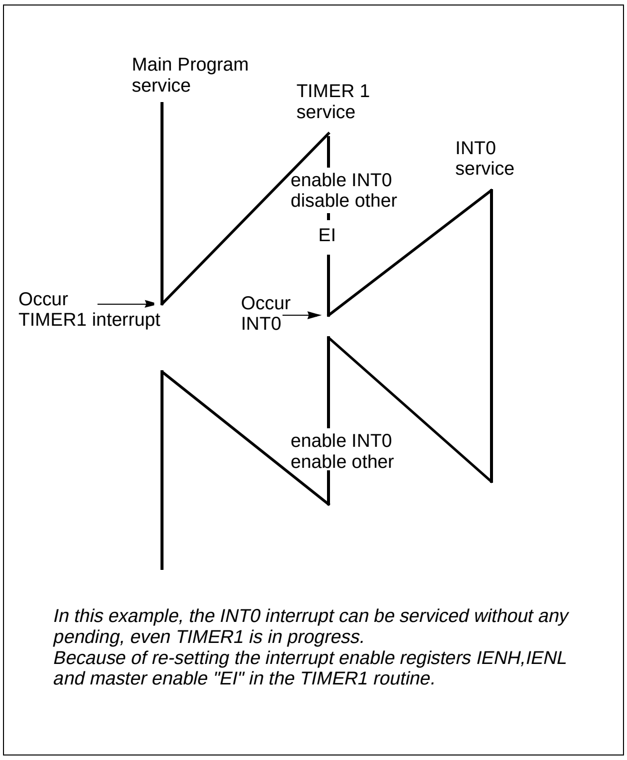
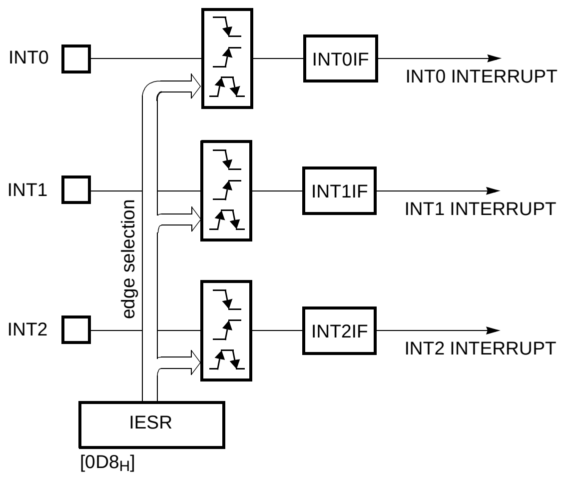


Figure 17-5 Execution of Multi Interrupt

17.4 External Interrupt

The external interrupt on INT0, INT1 and INT2 pins are edge triggered depending on the edge selection register IESR (address $0D8_H$) as shown in Figure 17-6.

The edge detection of external interrupt has three transition activated mode: rising edge, falling edge, and both edge.



IESR (Ext. Interrupt Edge Selection Register)

Bit	7	6	5	4	3	2	1	0
			R/W	R/W	R/W	R/W	R/W	R/W
	-	-	INT21	INT20	INT11	INT10	INT01	INT00

ADDRESS : $0D8_H$
RESET VALUE : $--000000_B$

INT2[1:0] (INT2 Edge Selections)
00 : Int. Disable
01 : Falling (1-to-0 transition)
10 : Rising (0-to-1 transition)
11 : Both (Rising & Falling)

INT1[1:0] (INT1 Edge Selection)
00 : Int. Disable
01 : Falling (1-to-0 transition)
10 : Rising (0-to-1 transition)
11 : Both (Rising & Falling)

INT0[1:0] (INT0 Edge Selections)
00 : Int. Disable
01 : Falling (1-to-0 transition)
10 : Rising (0-to-1 transition)
11 : Both (Rising & Falling)

Figure 17-6 External Interrupt Block Diagram

Example: To use as an INT0 and INT2

```
        :
        :
;**** Set port as an input port R0
        LDM    R0DR,#1111_1010B
        ;
;**** Set port as an interrupt port
        LDM    PMR,#0000_0101B
        ;
;**** Set Falling-edge Detection
        LDM    IESR,#0001_0001B
        :
        :
        :
```

Response Time

The INT0, INT1 and INT2 edge are latched into INT0F, INT1F and INT2F at every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The DIV itself takes twelve cycles. Thus, a maximum of twelve complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine.

Interrupt response timings are shown in Figure 17-7.

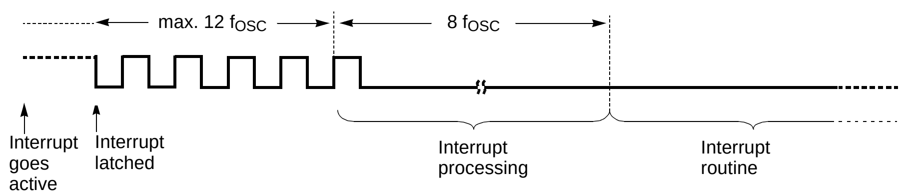
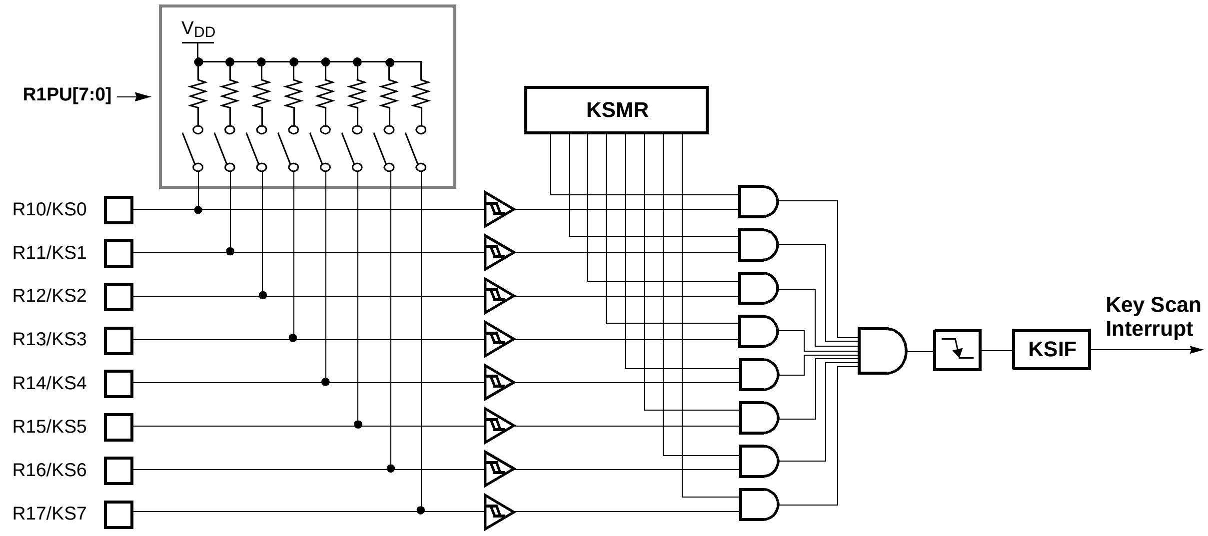


Figure 17-7 Interrupt Response Timing Diagram

18. KEY SCAN

The key-scan block consists of key scan mode register (KSMR) and R1 pull-up register (R1PU). When the key scan interrupt is used, key scan mode register KSMR (address $0F0_H$) should be set properly as shown in Figure 18-1. The pins which is to be used as key scan input should be set by KSMR and the strobe output pins should be set as open drain. The strobe output pins could be selected from among R0[7:0], R1[7:0], R2[3:0] and R3[3:0].

If the “L” signal is input to any one or more of key scan input pins, the KSIF request flag is set to “1”. This generates an interrupt request. It also can be used in the way of release from STOP mode.



KSMR (Key Scan Mode Register)

Bit :	7	6	5	4	3	2	1	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	KS7	KS6	KS5	KS4	KS3	KS2	KS1	KS0

ADDRESS : $0F0_H$
RESET VALUE : 00_H

0 : Port Function (I/O) Selection
1 : Key Scan Input Selection

Figure 18-1 Key Scan Interrupt Block Diagram

Usage of Key Scan

When key board scanning, it is recommended that set the output strobe to “L” first and then read R1 port after 60us delay time. Because the rising time of the output strobe port from “L” to “H” is so long. The Figure 18-2 explain this reason.

```
;Program Example,

LDM     R3,#0000_1110b ;R3<0> Port set to low
CALL    Delay_60us     ;60us time delay routine
LDA     R1             ;read R1 port
```

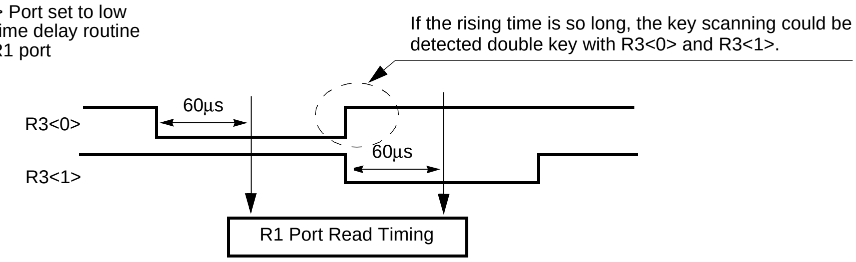


Figure 18-2 Key Scan Timing

19. LCD DRIVER

The GMS81C5108 has the circuit that directly drives the liquid crystal display (LCD) and its control circuit. The Segment/Common Driver directly drives the LCD panel, and the LCD Controller generates the segment/common signals according to the RAM which stores display data. In addition, VCL2 ~ VCL0 pin are provided as the drive power pins.

The GMS81C5108 has the following pins connected with LCD.

1. Segment output port 37 pins (SEG0-SEG36)

2.Common output port 4 pins (COM0-COM3)

19.1 Configuration of LCD driver

Figure 19-1 shows the configuration of the LCD driver.

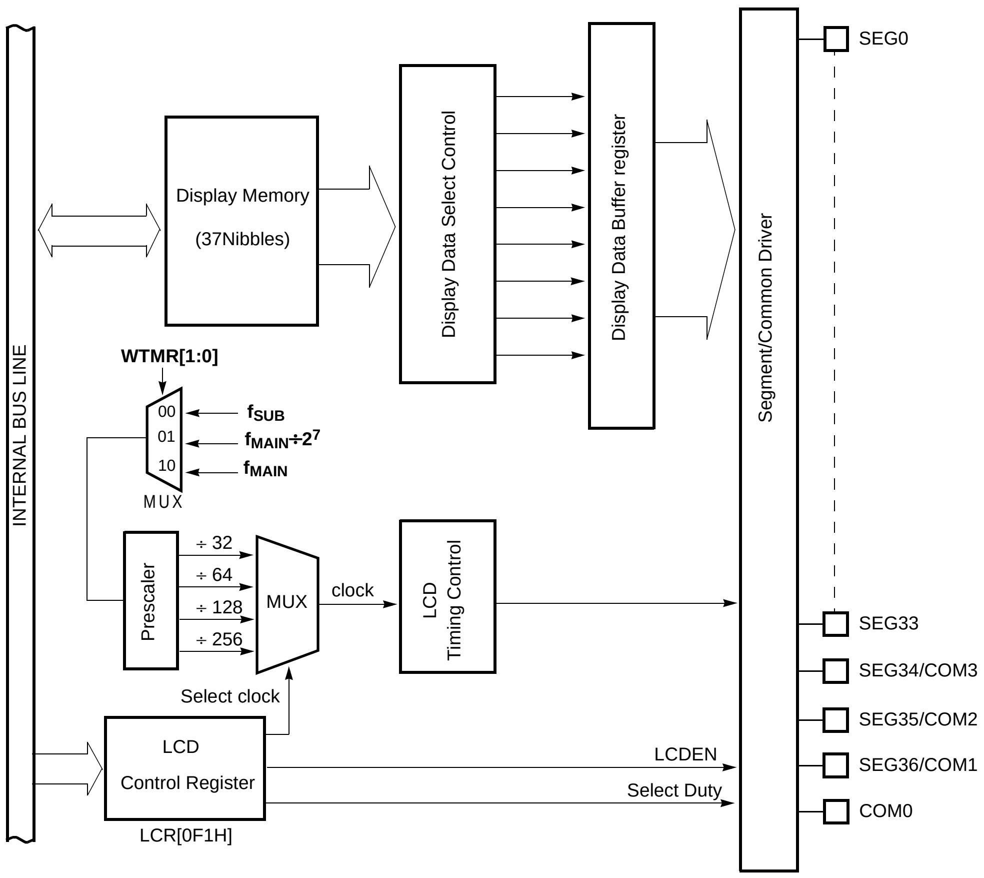


Figure 19-1 LCD Driver Block Diagram

19.2 Control of LCD Driver Circuit

The LCD driver is controlled by the LCD Control Register (LCR). The LCR[1:0] determines the frequency of COM signal scanning of each segment output. RESET clears the LCD control register LCR values to logic zero. The LCD display can continue to operate during SLEEP and STOP modes if a sub-frequency clock is used as system clock source. The constant voltage booster circuit for using LCD driver is built in, so the definite voltage could supplied regardless of power source voltage fluctuations.

Note: *The Sub clock is used as voltage booster source clock, so the stabilization time is need to use voltage booster. Normally, the stabilization time is need more than 500ms. The external bias registers cannot be used for LCD display supply voltage.*

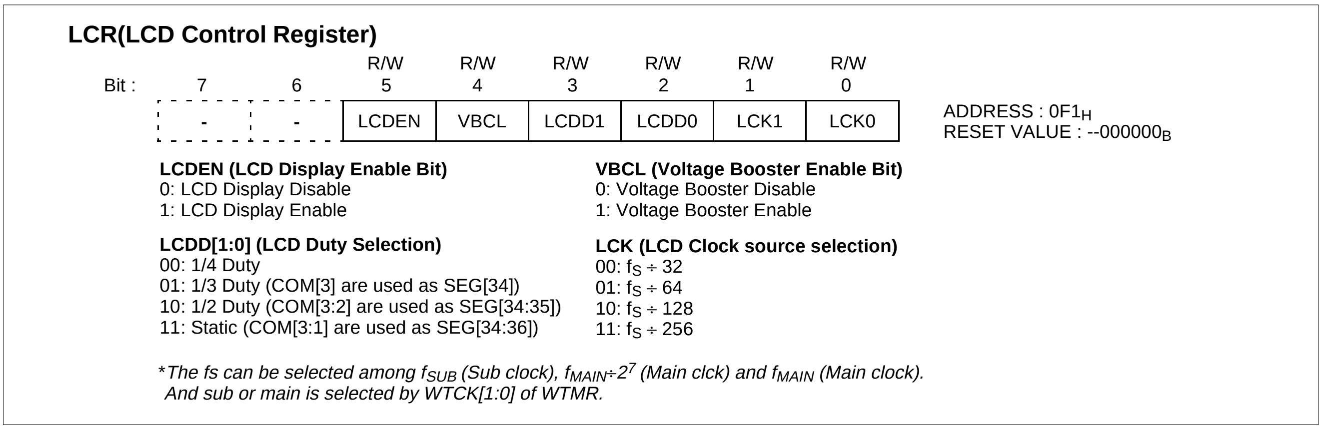


Figure 19-2 LCD Control Register

Selecting Frame Frequency

Frame frequency is set to the base frequency as shown in the following Table 19-1. The f_S is selected to f_{SUB} (sub clock) which is 32.768kHz.

LCR[1:0]	LCD clock	Frame Frequency (Hz)			
		Duty = Static	Duty = 1/2	Duty = 1/3	Duty = 1/4
00	$f_S \div 32$	1024	512	341.3	256
01	$f_S \div 64$	512	256	170.7	128
10	$f_S \div 128$	256	128	85.3	64
11	$f_S \div 256$	128	64	42.7	32

Table 19-1 Setting of LCD Frame Frequency

The matters to be attended to use LCD driver

In reset state, LCD source clock is sub clock. So, when the power is supplied, the LCD display would be flickered before the oscillation of sub clock is stabilized. It is recommended to use LCD display on after the stabilization time of sub clock is considered enough. If the LCD is reset during display, the display would be blotted by the capacity of LCD power circuit. The external circuit of constant voltage booster for using LCD driver is shown at right.

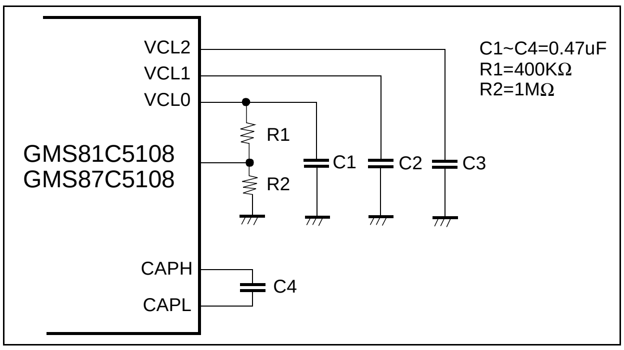


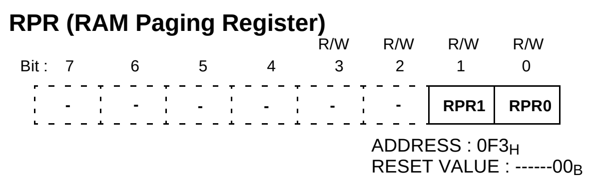
Figure 19-3 LCD Power Booster Circuit

19.3 LCD Display Memory

Display data are stored to the display data area (page 1) in the data memory.
The display data stored to the display data area (address 0100_H-0124_H) are read automatically and sent to the LCD driver by the hardware. The LCD driver generates the segment signals and common signals in accordance with the display data and drive method. Therefore, display patterns can be changed by only overwriting the contents of the display data area with a program. The table look up instruction is mainly used for this overwriting.
Figure 19.3 shows the correspondence between the display data area and the SEG/COM pins. The LCD lights when the display data is “1” and turn off when “0”.

LCD display memory in this location that are not used for LCD display can be allocated for general purpose use.

The SEG data for display is controlled by RPR (RAM Paging Register).



RAM Page	Instruction	RPR1	RPR0
0 Page	CLRG	x	x
0 Page	SETG	0	0
1 Page	SETG	0	1

Figure 19-4 Setting of RAM Paging Register

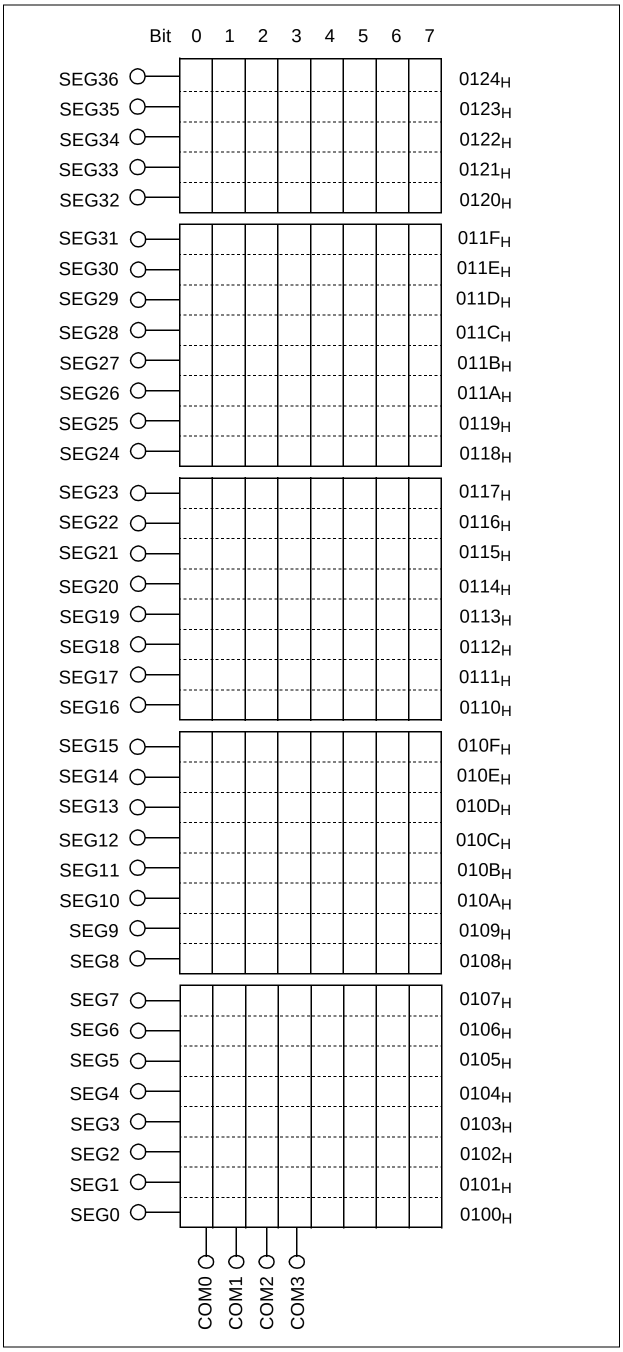


Figure 19-5 LCD Display Memory

19.4 Control Method of LCD Driver

Initial Setting

Flow chart of initial setting is shown in Figure 19-6.

Example: Driving of LCD

```
Select Frame Frequency          LDM     LCR,#12H      ;fF=64Hz, 1/4 duty(fSUB= 32.768kHz)
                                :
                                LDM     RPR,#1        ;Select LCD Memory(1 page)
                                SETG
Clear                           LDX     #0
LCD Display         C_LCD1:     LDA     #0            ;RAM Clear
Memory                                                ;(0100H->0124H)
                                STA     {X}+
                                CMPX    #025H
                                BNE     C_LCD1
                                CLRG
                                :
Turn on LCD                     SET1    LCR.5         ;Enable display
                                :
```

.

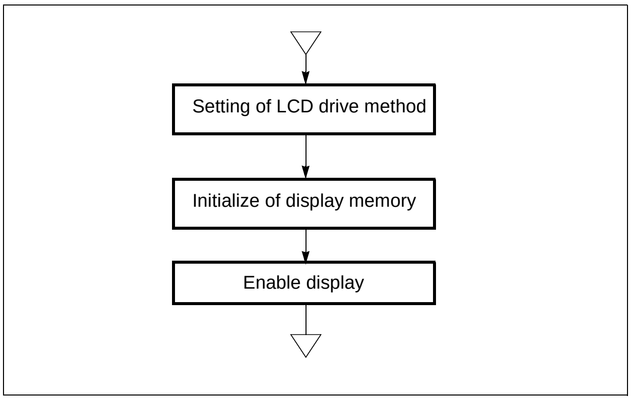


Figure 19-6 Initial Setting of LCD Driver

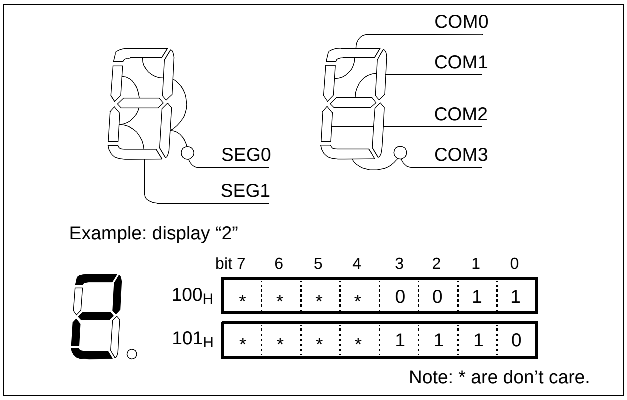


Figure 19-7 Example of Connection COM & SEG

Display Data

Normally, display data are kept permanently in the program memory and then stored at the display data area by the table look-up instruction. This can be explained using character display with 1/4 duty LCD as an example as well as any LCD panel. The COM and SEG connections to the LCD and display data are the same as those shown is Figure 19-7. Following is showing the Programming example for displaying character.

Note: *When power on RESET, sub oscillation start up time is required. Enable LCD display after sub oscillation is stabilized, or LCD may occur flicker at power on time shortly.*

Write into the LCD Memory

```
                :
                CLRG
                LDX     #<DISPRAM           ;Address included the data
                                            ;to be displayed.
GOLCD:          LDA     {X}
                TAY
                LDA     !FONT+Y             ;LOAD FONT DATA
                LDM     RPR,#1              ;Set RPR = 1 to access LCD
                SETG                        ;Set Page 1
                LDX     #0
                STA     {X}+                ;LOWER 4 BITS OF ACC. seg0
                XCN
                STA     {X}                 ;UPPER 4 BITS OF ACC. seg1
                CLRG                        ;Set Page = 0
                :
```

Font data

```
FONT            DB      1101_0111B          ;   "0"
                DB      0000_0110B          ;   "1"
                DB      1110_0011B          ;   "2"
                DB      1010_0111B          ;   "3"
                DB      0011_0110B          ;   "4"
                DB      1011_0101B          ;   "5"
                DB      1111_0101B          ;   "6"
                DB      0000_0111B          ;   "7"
                DB      1111_0111B          ;   "8"
                DB      0011_0111B          ;   "9"
```

LCD Waveform

The LCD duty can be selected by LCR register. The kinds of LCD waveforms are four totally. Among them, static and 1/4 duty waveforms are shown Figure 19-8.

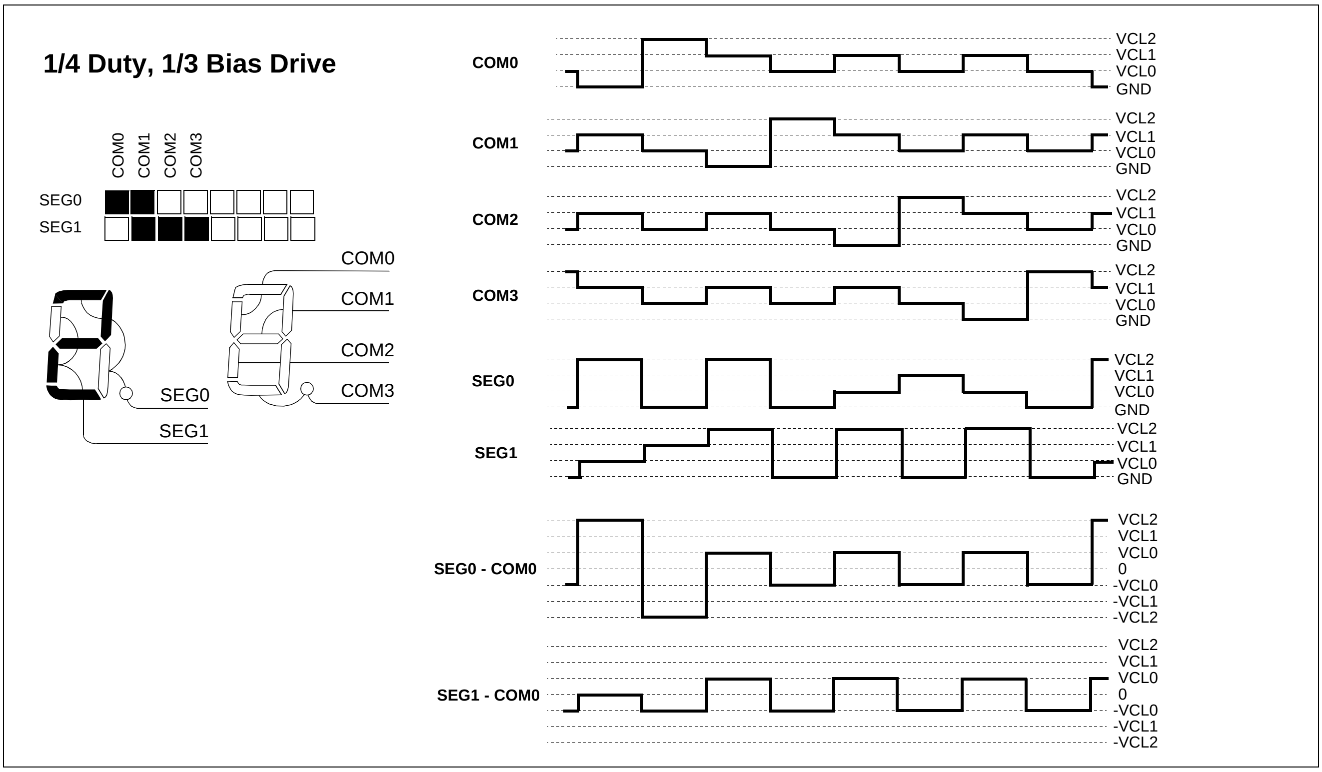


Figure 19-8 Example of LCD drive output

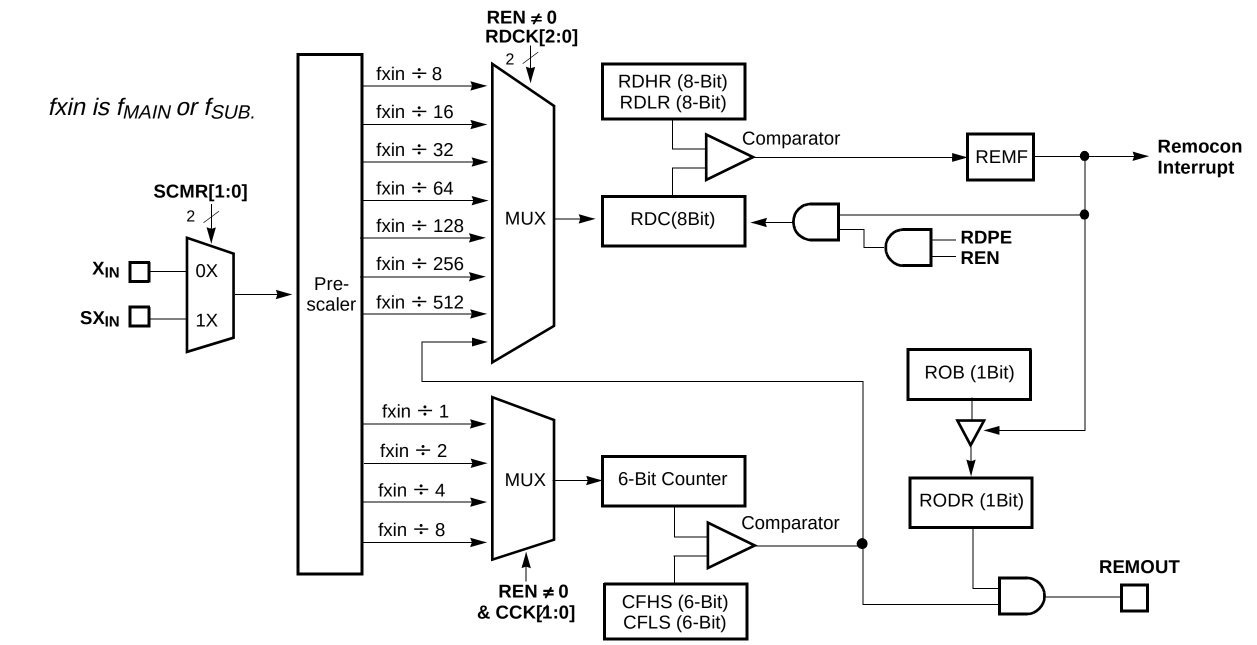
20. REMOCON CARRIER GENERATOR

The GMS81C5108 has a circuit to generate carriers for the remote controller. This circuit consists of Remocon Mode Register (RMR), Carrier Frequency High Selection (CFHS), Carrier Frequency Low Selection (CFLS), Remocon Data High Register (RDHR), Remocon Data Low Register (RDLR), Remocon Data Counter (RDC), Remocon Output Data Register (RODR) and Remocon Output Buffer (ROB) as shown in Figure 20-1. A carrier duty and frequency are determined by the contents of these registers. A source clock input to the 6-bit counter is selected by diving the frequency of the system clock by two (main or sub clock).

20.1 Remocon Signal Output Control

The output of the REMOUT pin which outputs carriers is controlled by RODR and ROB register. While the Bit-0 of RODR is “1”, the REMOUT pin outputs a carrier signal generated by the remote controller carrier generator. While this Bit is “0”, the output of the REMOUT pin is low.

The content of the ROB is automatically transferred to the RODR by an interrupt signal generated by the 8-Bit timer. The content of the RODR.0 is output to the REMOUT pin. Namely, the REMOUT pin outputs a high-level signal when RODR.0 is “1” and a low-level signal when RODR.0 is “0”.



RMR (Remocon Mode Register)

Bit :	7	6	5	4	3	2	1	0
		R/W	R/W	R/W	R/W	R/W	R/W	R/W
	-	REN	CCK1	CCK0	RDPE	RDCK2	RDCK1	RDCK0

ADDRESS : $0F6_H$
RESET VALUE : -0000000_B

REN (Remocon Operation Enable)
0 : Disable
1 : Enable

RDPE (Remocon Data Pulse Enable)
0 : Disable
1 : Enable

RDCK[2:0] (Remocon Data Clock Selection)
000: $f_{MAIN}\div2^3$ or $f_{SUB}\div2^3$
001: $f_{MAIN}\div2^4$ or $f_{SUB}\div2^4$
010: $f_{MAIN}\div2^5$ or $f_{SUB}\div2^5$
011: $f_{MAIN}\div2^6$ or $f_{SUB}\div2^6$
100: $f_{MAIN}\div2^7$ or $f_{SUB}\div2^7$
101: $f_{MAIN}\div2^8$ or $f_{SUB}\div2^8$
110: $f_{MAIN}\div2^9$ or $f_{SUB}\div2^9$
111: Carrier Signal

CCK[1:0] (Carrier Clock Source Selection)
00: f_{MAIN} or f_{SUB}
01: $f_{MAIN}\div2$ or $f_{SUB}\div2$
10: $f_{MAIN}\div2^2$ or $f_{SUB}\div2^2$
11: $f_{MAIN}\div2^3$ or $f_{SUB}\div2^3$

f_{MAIN}: main-clock frequency
f_{SUB}: sub-clock frequency

Figure 20-1 Remocon Carrier Generator Block Diagram

CFHS (Carrier Frequency High Selection)

Bit :	7	6	5	4	3	2	1	0
			W	W	W	W	W	W
	-	-	CFH5	CFH4	CFH3	CFH2	CFH1	CFH0

ADDRESS : $0F7_H$
RESET VALUE : $\text{--}111111_B$

Carrier High Interval = The Value of CFHS x Clock Source Period

CFLS (Carrier Frequency Low Selection)

Bit :	7	6	5	4	3	2	1	0
			W	W	W	W	W	W
	-	-	CFL5	CFL4	CFL3	CFL2	CFL1	CFL0

ADDRESS : $0F8_H$
RESET VALUE : $\text{--}111111_B$

Carrier Low Interval = The Value of CFLS x Clock Source Period

RDHR (Remocon Data High Register)

Bit :	7	6	5	4	3	2	1	0
	W	W	W	W	W	W	W	W
	RDH7	RDH6	RDH5	RDH4	RDH3	RDH2	RDH1	RDH0

ADDRESS : $0F9_H$
RESET VALUE : 11111111_B

Remocon Data High Interval = The Value of RDHR x Clock Source Period

RDLR (Remocon Data Low Register)

Bit :	7	6	5	4	3	2	1	0
	W	W	W	W	W	W	W	W
	RDL7	RDL6	RDL5	RDL4	RDL3	RDL2	RDL1	RDL0

ADDRESS : $00FA_H$
RESET VALUE : 11111111_B

Remocon Data Low Interval = The Value of RDLR x Clock Source Period

RDC (Remocon Data Counter)

Bit :	7	6	5	4	3	2	1	0
	R	R	R	R	R	R	R	R
	RDC7	RDC6	RDC5	RDC4	RDC3	RDC2	RDC1	RDC0

ADDRESS : $00FA_H$
RESET VALUE : 00000000_B

Remocon Data Counter Value

RODR (Remocon Output Data Register)

Bit :	7	6	5	4	3	2	1	0
								R/W
	-	-	-	-	-	-	-	RDD0

ADDRESS : $0FB_H$
RESET VALUE : $\text{-------}0_B$

Remocon Data Output Value

ROB (Remocon Output Buffer)

Bit :	7	6	5	4	3	2	1	0
								R/W
	-	-	-	-	-	-	-	RDB0

ADDRESS : $0FC_H$
RESET VALUE : $\text{-------}0_B$

Remocon Data Output Buffer

Figure 20-2 Remocon Registers

20.2 Carrier Frequency

The carrier frequency and the pulse of data are calculated by below formula. The the lengths of carrier frequency and pulse of data are shown in Figure 20-3.

t_H = source clock(RMR[5:4]) × CFHS
t_L = source clock(RMR[5:4]) × CFHS
f_C (Carrier Frequency) = $1/(t_H+t_L)$
t_{DH} = source clock(RMR[2:0]) × RDHR
t_{DL} = source clock(RMR[2:0]) × RDLR

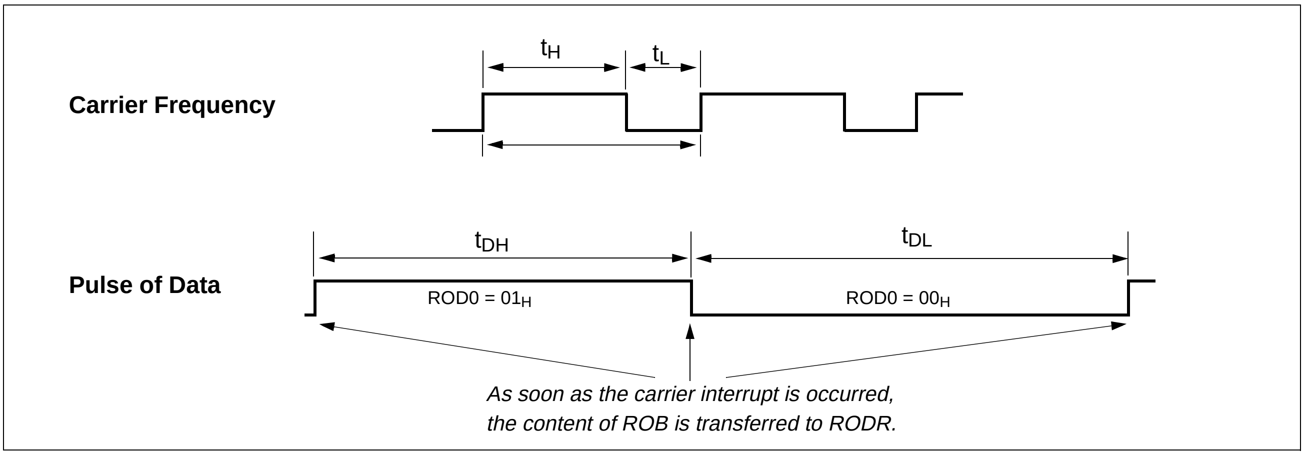


Figure 20-3 Carrier Frequency & Pulse of Data

The Table 20-1 shows high and low length of carrier frequency according to CFLS and CFHS. This only shows when the source clock is selected f_{MAIN} and $f_{MAIN} \div 2^2$ at 4MHz.

Set Value		Selection of PS0		Selection of PS2	
CFHS	**CFLS**	**t_H(us)**	**t_L(us)**	**t_H(us)**	**t_L(us)**
00$_H$	00$_H$	-	-	-	-
01$_H$	01$_H$	0.25	0.25	1.00	1.00
02$_H$	02$_H$	0.50	0.50	2.00	2.00
03$_H$	03H	0.75	0.75	3.00	3.00
04$_H$	04$_H$	1.00	1.00	4.00	4.00
05$_H$	05$_H$	1.25	1.25	5.00	5.00
06$_H$	06$_H$	1.50	1.50	6.00	6.00
07$_H$	07$_H$	1.75	1.75	7.00	7.00
08$_H$	08$_H$	2.00	2.00	8.00	8.00
09$_H$	09$_H$	2.25	2.25	9.00	9.00
0A$_H$	0A$_H$	2.50	2.50	10.00	10.00
0B$_H$	0B$_H$	2.75	2.75	11.00	11.00
0C$_H$	0C$_H$	3.00	3.00	12.00	12.00
0D$_H$	0D$_H$	3.25	3.25	13.00	13.00
0E$_H$	0E$_H$	3.50	3.50	14.00	14.00
0F$_H$	0F$_H$	3.75	3.75	15.00	15.00
10$_H$	10$_H$	4.00	4.00	16.00	16.00
11$_H$	11$_H$	4.25	4.25	17.00	17.00
12$_H$	12$_H$	4.50	4.50	18.00	18.00
13$_H$	13$_H$	4.75	4.75	19.00	19.00
14$_H$	14$_H$	5.00	5.00	20.00	20.00
15$_H$	15$_H$	5.25	5.25	21.00	21.00
16$_H$	16$_H$	5.50	5.50	22.00	22.00
17$_H$	17$_H$	5.75	5.75	23.00	23.00
18$_H$	18$_H$	6.00	6.00	24.00	24.00
19$_H$	19$_H$	6.25	6.25	25.00	25.00
1A$_H$	1A$_H$	6.50	6.50	26.00	26.00
1B$_H$	1B$_H$	6.75	6.75	27.00	27.00
1C$_H$	1C$_H$	7.00	7.00	28.00	28.00
1D$_H$	1D$_H$	7.25	7.25	29.00	29.00
1E$_H$	1E$_H$	7.50	7.50	30.00	30.00
1F$_H$	1F$_H$	7.75	7.75	31.00	31.00

Set Value		Selection of PS0		Selection of PS2	
CFHS	**CFLS**	**t_H(us)**	**t_L(us)**	**t_H(us)**	**t_L(us)**
20$_H$	20$_H$	8.00	8.00	32.00	32.00
21$_H$	21$_H$	8.25	8.25	33.00	33.00
22$_H$	22$_H$	8.50	8.50	34.00	34.00
23$_H$	23$_H$	8.75	8.75	35.00	35.00
24$_H$	24$_H$	9.00	9.00	36.00	36.00
25$_H$	25$_H$	9.25	9.25	37.00	37.00
26$_H$	26$_H$	9.50	9.50	38.00	38.00
27$_H$	27$_H$	9.75	9.75	39.00	39.00
28$_H$	28$_H$	10.00	10.00	40.00	40.00
29$_H$	29$_H$	10.25	10.25	41.00	41.00
2A$_H$	2A$_H$	10.50	10.50	42.00	42.00
2B$_H$	2B$_H$	10.75	10.75	43.00	43.00
2C$_H$	2C$_H$	11.00	11.00	44.00	44.00
2D$_H$	2D$_H$	11.25	11.25	45.00	45.00
2E$_H$	2E$_H$	11.50	11.50	46.00	46.00
2F$_H$	2F$_H$	11.75	11.75	47.00	47.00
30$_H$	30$_H$	12.00	12.00	48.00	48.00
31$_H$	31$_H$	12.25	12.25	49.00	49.00
32$_H$	32$_H$	12.50	12.50	50.00	50.00
33$_H$	33$_H$	12.75	12.75	51.00	51.00
34$_H$	34$_H$	13.00	13.00	52.00	52.00
35$_H$	35$_H$	13.25	13.25	53.00	53.00
36$_H$	36$_H$	13.50	13.50	54.00	54.00
37$_H$	37$_H$	13.75	13.75	55.00	55.00
38$_H$	38$_H$	14.00	14.00	56.00	56.00
39$_H$	39$_H$	14.25	14.25	57.00	57.00
3A$_H$	3A$_H$	14.50	14.50	58.00	58.00
3B$_H$	3B$_H$	14.75	14.75	59.00	59.00
3C$_H$	3C$_H$	15.00	15.00	60.00	60.00
3D$_H$	3D$_H$	15.25	15.25	61.00	61.00
3E$_H$	3E$_H$	15.50	15.50	62.00	62.00
3F$_H$	3F$_H$	15.75	15.75	63.00	63.00

Table 20-1 Length of Carrier Frequency (at 4MHz)

Example:

```
        Carrier Frequency = 37.8kHz, high = 8.52ms, low = 4.24ms, @4MHz

Rem_sig: LDM    RMR,#0001_0010B     ;carrier clock(PS1), remocon data clock(PS5)
         LDM    CFHS,#18           ;carrier low(IR LED)=18*PS1(0.5us)=9us
         LDM    CFLS,#35           ;carrier high(IR LED)=35*PS1(0.5us)=17.5us
         CLR1   ROD0
         LDM    R_bit,#1111_1000B
         LDM    RDHR,#213          ;213*5*PS5(8us)=8.52ms
         LDM    RDLR,#177          ;177*3*PS5(8us)=4.248ms
         LDX    #9

         CALL   DATA
         SET1   RMR.6              ;Remocon operation enable
         SET1   RMR.3              ;Remocon data pulse enable
         SET1   IENL.6             ;Remocon int.

Loop1:   NOP
         CMPX   #0
         BNE    Loop1

Finish:  CLR1   ROD0
         CLR1   ROB0
         RET

;********
Data:    ROL    R_bit
         BCS    Set_rob0
         CLR1   ROB0
         RET

Set_rob0:SET1ROB0
         RET

;************************************************;
;          Remocon int service routine            ;
;************************************************;
;
Remocon_INT:

         CALL   Data
         DEC    X
         RETI
```

21. OSCILLATOR CIRCUIT

The GMS81C5108 has two oscillation circuits internally. X_{IN} and X_{OUT} are input and output for main frequency and SX_{IN} and SX_{OUT} are input and output for sub frequency, respectively, inverting amplifier which can be configured for being used as an on-chip oscillator, as shown in Figure 21-1.

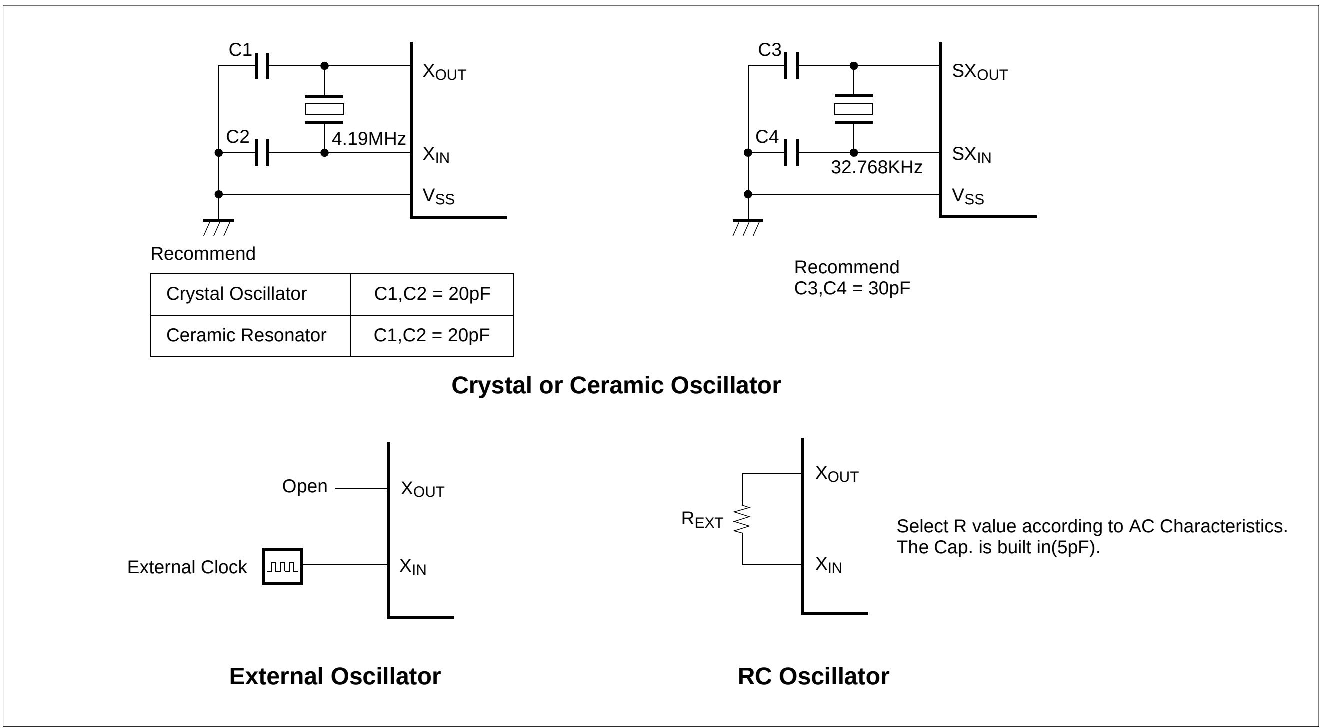


Figure 21-1 Oscillation Circuit

Oscillation circuit is designed to be used either with a ceramic resonator or crystal oscillator. Since each crystal and ceramic resonator have their own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

In addition, see Figure 21-2 for the layout of the crystal.

Note: *Minimize the wiring length. Do not allow the wiring to intersect with other signal conductors. Do not allow the wiring to come near changing high current. Set the potential of the grounding position of the oscillator capacitor to that of V_{SS}. Do not ground it to any ground pattern where high current is present. Do not fetch signals from the oscillator.*

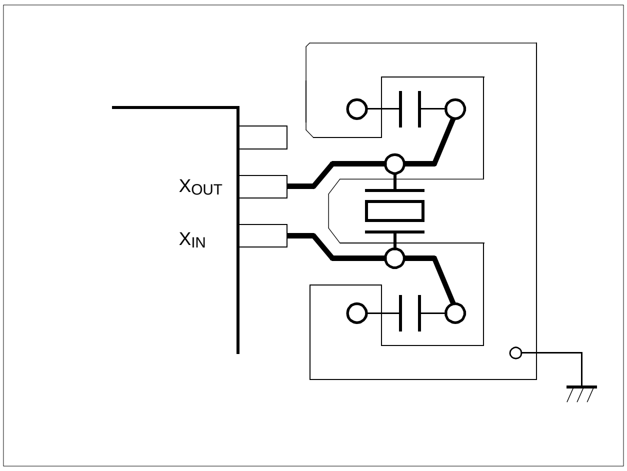


Figure 21-2 Layout of Oscillator PCB circuit

22. RESET

The GMS81C5108 have two types of reset generation procedures; one is an external reset input, the other is a watchdog timer reset. Table 22-1 shows on-chip hardware initialization by reset action.

On-chip Hardware	Initial Value	On-chip Hardware	Initial Value
Program counter (PC)	($FFFF_H$) - ($FFFE_H$)	Peripheral clock	On
RAM page register (RPR)	0	SVD	Enable
G-flag (G)	0	Control registers	Refer to Table 8-1 on page 25
Operation mode	Main-frequency clock	Voltage Booster	Disable

Table 22-1 Initializing Internal Status by Reset Action

22.1 External Reset Input

The reset input is the RESET pin, which is the input to a Schmitt Trigger. A reset in accomplished by holding the RESET pin to low for at least 8 oscillator periods, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. After reset, 65.5ms (at 4MHz) add with 7 oscillator periods are required to start execution as shown in Figure 22-2.

Internal RAM is not affected by reset. When V_{DD} is turned on, the RAM content is indeterminate. Therefore, this RAM should be initialized before read or tested it.

When the RESET pin input goes to high, the reset operation is released and the program execution starts at the vector address stored at $FFFE_H$ - $FFFF_H$.

A connection for simple power-on-reset is shown in Figure 22-1.

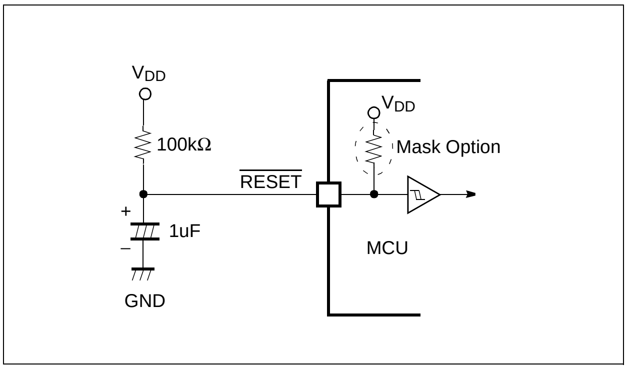


Figure 22-1 Simple Power-on-Reset Circuit

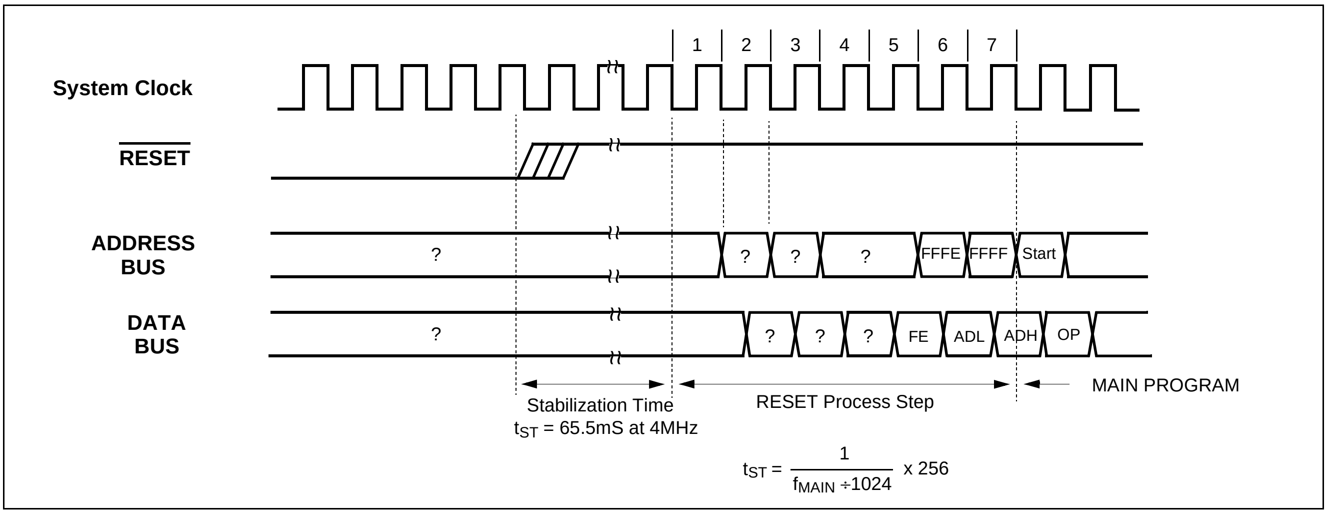


Figure 22-2 Timing Diagram after RESET

22.2 Watchdog Timer Reset

Refer to “13.2 Watch Dog Timer” on page 57.

23. SUPPLY VOLTAGE DETECTION

The GMS81C5108 has an on-chip low voltage detection circuitry to detect the V_{DD} voltage. A configuration register, SCMR, can enable or disable the low voltage detect circuitry. This GMS81C5108 has two level detector(SVD0, SVD1). The SVD0 flag is set when the V_{DD} falls below 2.2V and if the V_{DD} is rise above 2.2V the SVD0 is cleared automatically. The SVD1 flag is set when the V_{DD} falls below 1.7V and if this flag is set once, it is not cleared automatically although the V_{DD} rises above 1.7V. It can be cleared by writing.

If the SVD1 is set, the MCU can be RESET or frozen by the flag SVRT. In the in-circuit emulator, supply voltage detection is not implemented and user can not experiment with it. Therefore, after final development of user program, this function may be experimented or evaluated.

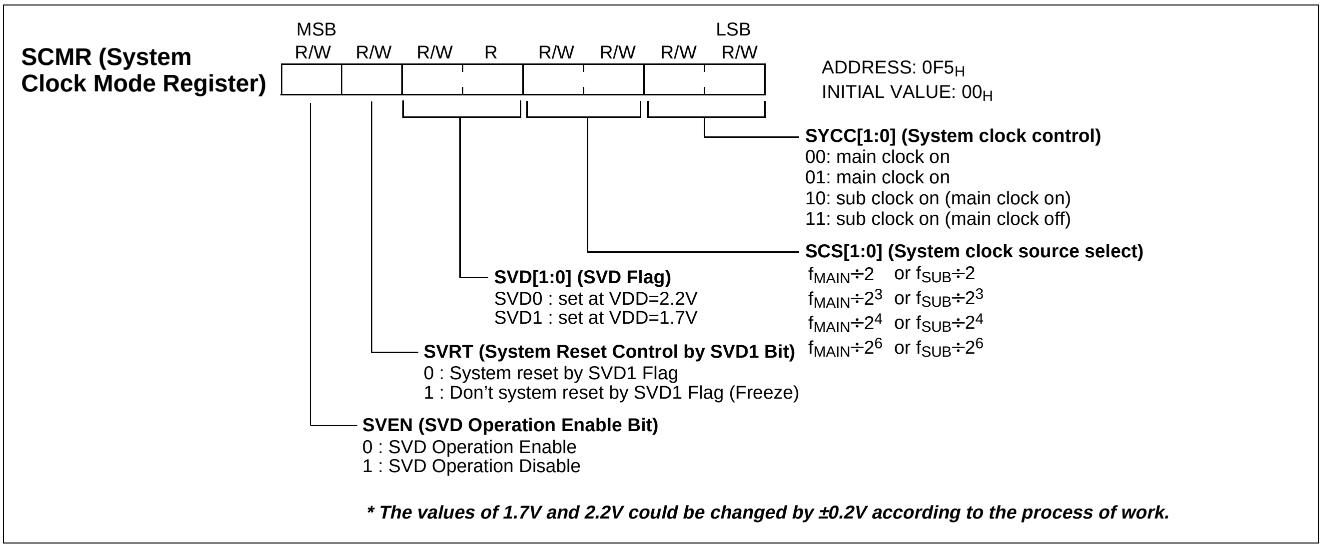


Figure 23-1 Low Voltage Detector Register

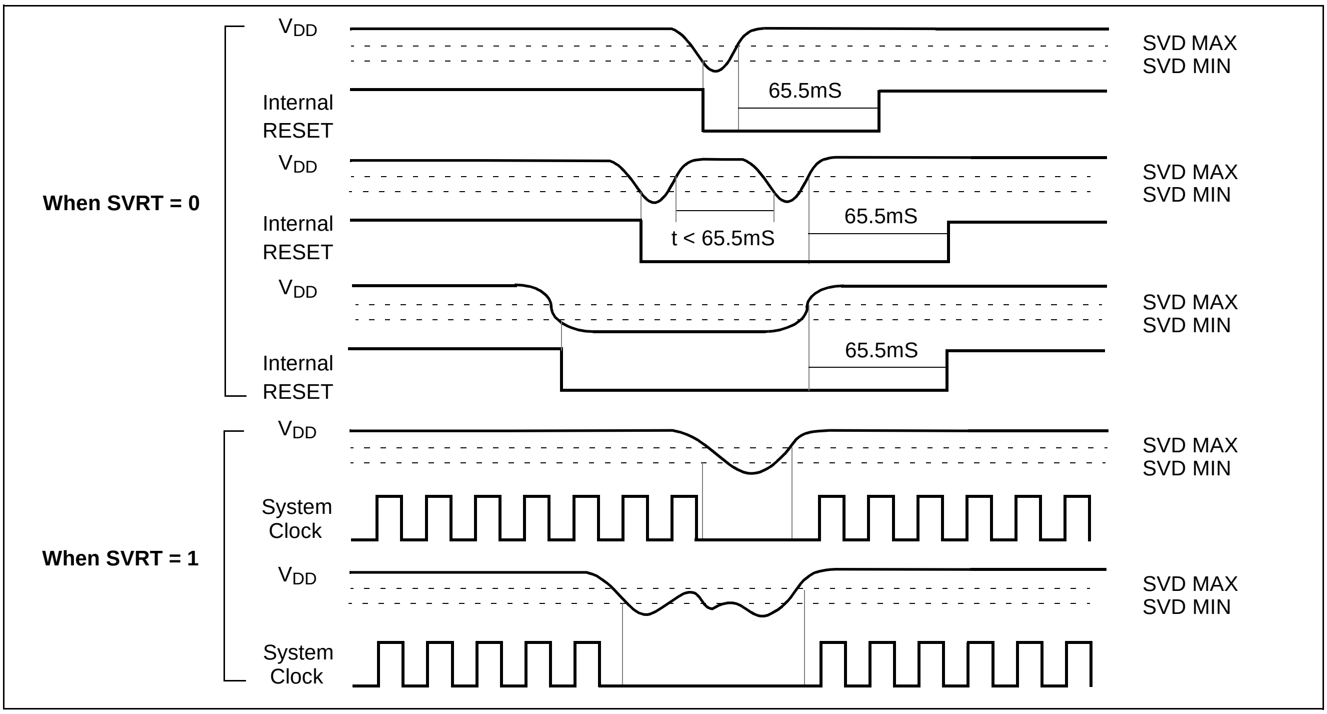


Figure 23-2 Power Fail Processor Situations

24. DEVEMOPMENT TOOLS

24.1 OTP Programming

The GMS87C5108 is an OTP (One Time Programmable) microcontrollers. Its internal user memory is constructed with EPROM (Electrically Programmable Read Only Memory).

The OTP microcontroller is generally used for chip evaluation, first production, small amount production, fast mass production, etc.

Blank OTP's internal EPROM is filled by 00_H, not FF_H.

Note: *In any case, you have to use the *.OTP file for programming, not the *.HEX file. After assemble, both OTP and HEX file are generated by automatically. The HEX file is used during program emulation on the emulator.*

How to Program

To program the OTP devices, user can use Hynix own programmer.

Hynix own programmer list

Manufacturer: Hynix Semiconductor Programmer:

Choice-Sigma
Choice-Gang4

The Choice-Sigma is a Hynix Universal Single Programmer for all of Hynix OTP devices, also the Choice-Gang4 can program four OTPs at once for Hynix OTP.

Ask to Hynix sales part for purchasing or more detail

Programming Procedure

1. Select device GMS87C5108 you want.
2. Load the *.OTP file from the PC. The file is composed of Motorola-S1 format.
3. Set the programming address range as below table.

Address	Set Value
Buffer start address	$E000_H$
Buffer end address	$FFFF_H$
Device start address	$E000_H$

4. Mount the socket adapter on the programmer.
5. Start program/verify.

Pin Function

V_{PP} (Program Voltage)
V_{PP} is the input for the program voltage for programming the EPROM.

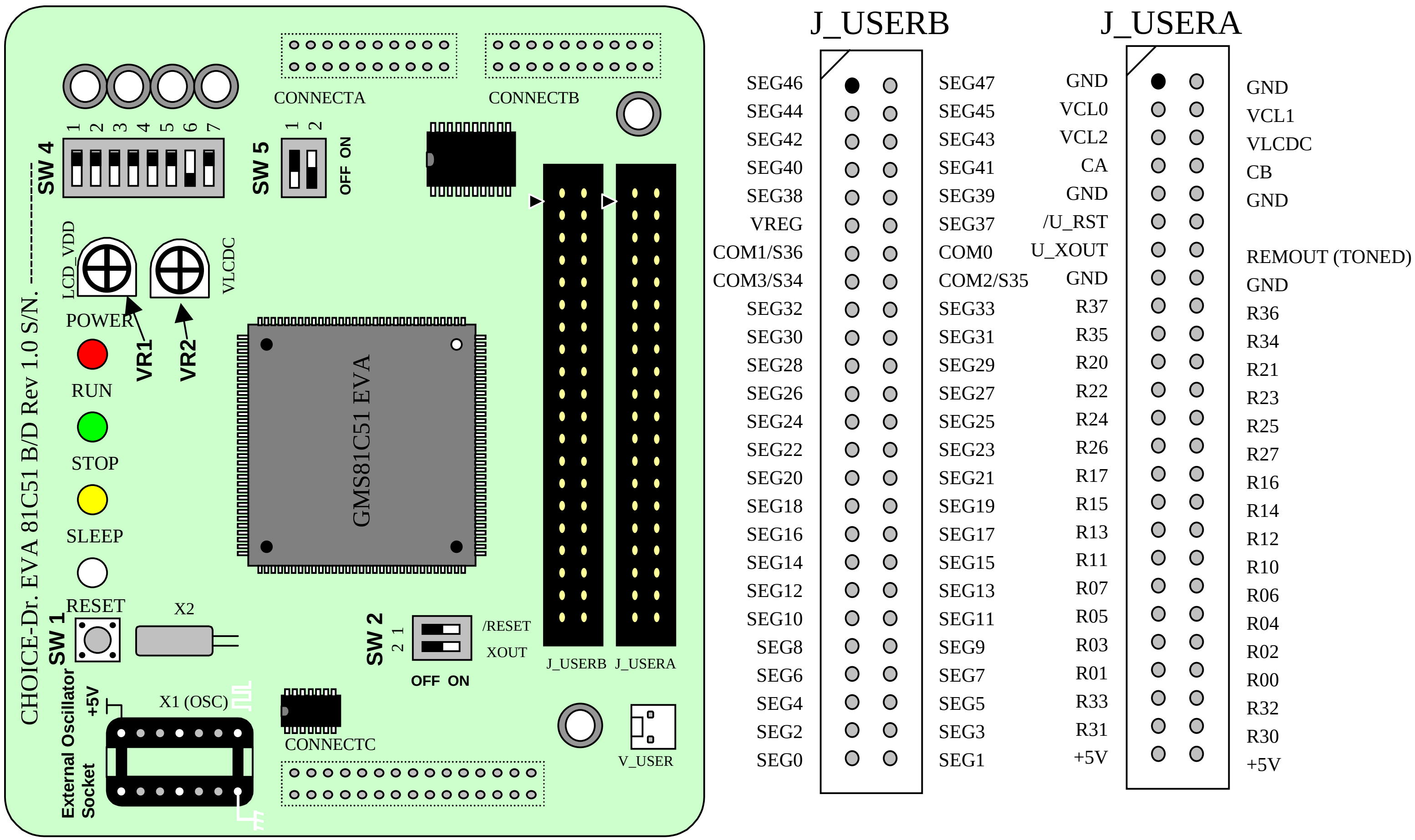
$\overline{CE}$ (Chip Enable)
CE is the input for programming and verifying internal EPROM.

$\overline{OE}$ (Output Enable)
OE is the input of data output control signal for verify.

A0~A15 (Address Bus)
A0~A15 are address input pins for internal EPROM.

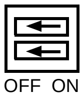
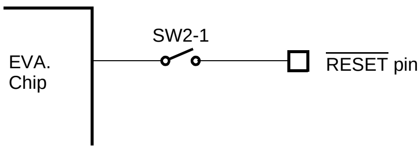
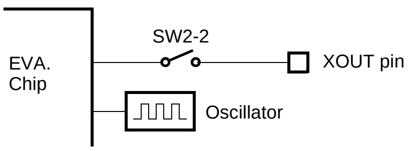
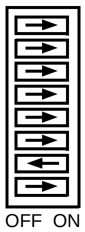
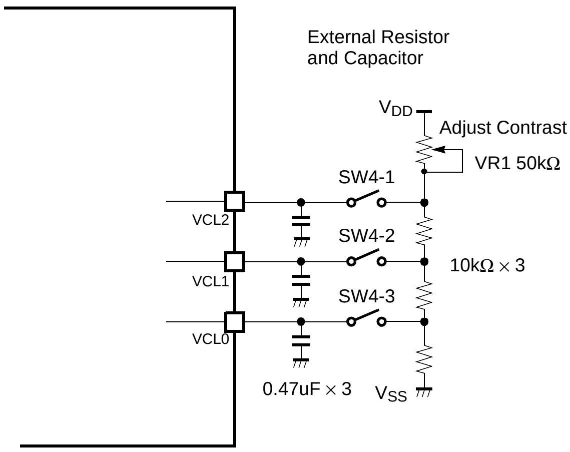
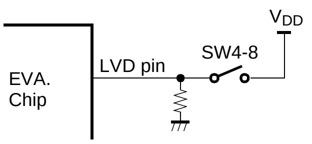
O0~O7 (EPROM Data Bus)
These are data bus for internal EPROM.

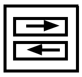
24.2 Emulator S/W Setting



DIP Switch and VR Setting

Before execute the user program, keep in your mind the below configuration

DIP S/W, VR		Description	ON/OFF Setting
SW1	-	Emulator Reset Switch. Reset the Emulator.	Reset the Emulator.
 SW2	1		Normally **OFF**. EVA. chip can be reset by external user target system board. **ON** : Reset is available by user target system board. **OFF** : MCU is reset by REST switch on EVA. board.
	2		Normally **OFF**. MCU XOUT pin are disconnected by Emulator internally. Some circumstance user may connect this circuit. **ON** : Output XOUT signal **OFF** : Disconnect circuit
 SW4	1 2 3		Normally **ON**. It serves the external bias resistors. If user want to use external circuit instead of internal R, turn on these switches.
	4 5 6	LCD Voltage booster circuit.	Must be **ON** position. It is used for the GMS81C5108.
	7	Select the Stack Page.	Must be **OFF** position. This switch decide the Stack page 0 (off) or page 1 (on). **ON** : For the GMS81C7XXX **OFF** : For the GMS81C5108
	8	 GMS81C5108 detect the V_{DD} voltage but Emulator can not do because Emulator can not operate if V_{DD} is below normal opr. voltage (5V), This switch serves LVD environment through the applying 0V to LVD pin of EVA. chip during 5V normal operation.	Position **ON** during normal operation. **ON** : Normal operation **OFF** : Force to detect the LVD, refer to "23. SUPPLY VOLTAGE DETECTION" on page 82.

DIP S/W, VR		Description	ON/OFF Setting
SW5	1	Internal power supply to sub-oscillation circuit.	Must be **ON** position.
	2	Reserved for other purpose.	Must be **OFF** position.
VR1	-	Adjust the LCD contrast. It control the VCL2 voltage. Refer to above SW4-1,2,3 figure.	Adjust the proper position as well as LCD display good.
VR2	-	Reserved for other purpose.	Don’t care.

Book History

This Book Ver 1.0 (JUNE 2001)

First edition.

APPENDIX

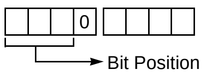
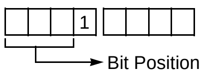
A. CONTROL REGISTER LIST

Address	Register Name	Symbol	R/W	Initial Value								Page
				7	6	5	4	3	2	1	0	
00C0	R0 port data register	R0	R/W	0	0	0	0	0	0	0	0	32
00C1	R1 port data register	R1	R/W	0	0	0	0	0	0	0	0	32
00C2	R2 port data register	R2	R/W	-	-	-	-	0	0	0	0	33
00C3	R3 port data register	R3	R/W	-	-	-	-	0	0	0	0	33
00C8	R0 port I/O direction register	R0DR	W	0	0	0	0	0	0	0	0	32
00C9	R1 port I/O direction register	R1DR	W	0	0	0	0	0	0	0	0	32
00CA	R2 port I/O direction register	R2DR	W	-	-	-	-	0	0	0	0	33
00CB	R3 port I/O direction register	R3DR	W	-	-	-	-	0	0	0	0	33
00D0	R0 port pull-up register	R0PU	W	0	0	0	0	0	0	0	0	32
00D1	R1 port pull-up register	R1PU	W	0	0	0	0	0	0	0	0	32
00D2	R2 port pull-up register	R2PU	W	-	-	-	-	0	0	0	0	33
00D3	R3 port pull-up register	R3PU	W	-	-	-	-	0	0	0	0	33
00D4	R0 port open drain control register	R0CR	W	0	0	0	0	0	0	0	0	32
00D5	R1 port open drain control register	R1CR	W	0	0	0	0	0	0	0	0	32
00D6	R2 port open drain control register	R2CR	W	-	-	-	-	0	0	0	0	33
00D7	R3 port open drain control register	R3CR	W	-	-	-	-	0	0	0	0	33
00D8	Ext. interrupt edge selection register	IESR	R/W	-	-	0	0	0	0	0	0	69
00D9	Port selection register	PMR	R/W	-	0	-	0	0	0	0	0	32
00DA	Interrupt enable low register	IENL	R/W	-	0	0	0	0	-	-	-	65
00DB	Interrupt enable high register	IENH	R/W	-	0	0	0	0	0	0	0	65
00DC	Interrupt request flag low register	IRQL	R/W	-	0	0	0	0	-	-	-	65
00DD	Interrupt request flag high register	IRQH	R/W	-	0	0	0	0	0	0	0	65
00DE	Sleep mode register	SMR	R/W	-	-	-	-	-	-	-	0	39
00E0	Timer 0 mode register	TM0	R/W	-	-	0	0	0	0	0	0	45
00E1	Timer 0 counter register	T0	R	0	0	0	0	0	0	0	0	45
	Timer 0 data register	TDR0	W	1	1	1	1	1	1	1	1	45
	Timer 0 input capture register	CDR0	R	0	0	0	0	0	0	0	0	45
00E2	Timer 1 mode register	TM1	R/W	0	0	0	0	0	0	0	0	45
00E3	Timer 1 data register	TDR1	W	1	1	1	1	1	1	1	1	45
	PWM0 pulse period register	T1PPR	W	1	1	1	1	1	1	1	1	45
00E4	Timer 1 counter register	T1	R	0	0	0	0	0	0	0	0	45
	Timer 1 input capture register	CDR1	R	0	0	0	0	0	0	0	0	45
	PWM0 pulse duty register	T1PDR	R/W	0	0	0	0	0	0	0	0	45
00E5	PWM0 high register	PWMHR	W	-	-	-	-	0	0	0	0	45
00EC	A/D converter mode register	ADMR	R/W	-	0	-	-	0	0	0	1	58
00ED	A/D converter data register	ADDR	R	x	x	x	x	x	x	x	x	58

Address	Register Name	Symbol	R/W	Initial Value								Page
				7	6	5	4	3	2	1	0	
00EF	Watch timer mode register	WTMR	R/W	-	0	0	0	0	0	0	0	56
00F0	Key scan mode register	KSMR	R/W	0	0	0	0	0	0	0	0	70
00F1	LCD control register	LCR	R/W	0	0	0	0	0	0	0	0	72
00F3	RAM paging register	RPR	R/W	-	-	-	-	-	-	0	0	73
00F4	Basic interval timer register	BITR	R	0	0	0	0	0	0	0	0	43
	Clock control register	CKCTLR	W	-	-	-	-	0	1	1	1	43
00F5	System clock mode register	SCMR	R/W	0	0	0	0	0	0	0	0	34
00F6	Remocon mode register	RMR	R/W	-	0	0	0	0	0	0	0	76
00F7	Carrier frequency high selection	CFHS	W	-	-	1	1	1	1	1	1	76
00F8	Carrier frequency low selection	CFLS	W	-	-	1	1	1	1	1	1	76
00F9	Remocon data high register	RDHR	W	1	1	1	1	1	1	1	1	76
00FA	Remocon data low register	RDLR	W	1	1	1	1	1	1	1	1	76
	Remocon data counter	RDC	R	0	0	0	0	0	0	0	0	76
00FB	Remocon output data register	RODR	R/W	-	-	-	-	-	-	-	0	76
00FC	Remocon output buffer	ROB	R/W	-	-	-	-	-	-	-	0	76
00FD	Buzzer data register	BDR	W	0	0	0	0	0	0	0	0	60
00FE	Serial I/O mode register	SIOM	R/W	0	0	0	0	0	0	0	1	62
00FF	Serial I/O data register	SIOD	R/W	x	x	x	x	x	x	x	x	62

B. INSTRUCTION

B.1 Terminology List

Terminology	Description
A	Accumulator
X	X - register
Y	Y - register
PSW	Program Status Word
#imm	8-bit Immediate data
dp	Direct Page Offset Address
!abs	Absolute Address
[ ]	Indirect expression
{ }	Register Indirect expression
{ }+	Register Indirect expression, after that, Register auto-increment
.bit	Bit Position
A.bit	Bit Position of Accumulator
dp.bit	Bit Position of Direct Page Memory
M.bit	Bit Position of Memory Data (000_H~$0FFF_H$)
rel	Relative Addressing Data
upage	U-page ($0FF00_H$~$0FFFF_H$) Offset Address
n	Table CALL Number (0~15)
+	Addition
x	 Upper Nibble Expression in Opcode
y	 Upper Nibble Expression in Opcode
–	Subtraction
×	Multiplication
/	Division
( )	Contents Expression
∧	AND
∨	OR
⊕	Exclusive OR
~	NOT
←	Assignment / Transfer / Shift Left
→	Shift Right
↔	Exchange
=	Equal
≠	Not Equal

B.2 Instruction Map

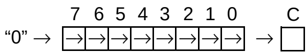
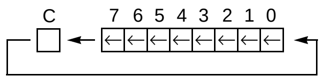
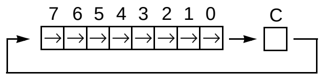
LOW / HIGH	00000 00	00001 01	00010 02	00011 03	00100 04	00101 05	00110 06	00111 07	01000 08	01001 09	01010 0A	01011 0B	01100 0C	01101 0D	01110 0E	01111 0F
000	-	SET1 dp.bit	BBS A.bit,rel	BBS dp.bit,rel	ADC #imm	ADC dp	ADC dp+X	ADC !abs	ASL A	ASL dp	TCALL 0	SETA1 .bit	BIT dp	POP A	PUSH A	BRK
001	CLRC				SBC #imm	SBC dp	SBC dp+X	SBC !abs	ROL A	ROL dp	TCALL 2	CLRA1 .bit	COM dp	POP X	PUSH X	BRA rel
010	CLRG				CMP #imm	CMP dp	CMP dp+X	CMP !abs	LSR A	LSR dp	TCALL 4	NOT1 M.bit	TST dp	POP Y	PUSH Y	PCALL Upage
011	DI				OR #imm	OR dp	OR dp+X	OR !abs	ROR A	ROR dp	TCALL 6	OR1 OR1B	CMPX dp	POP PSW	PUSH PSW	RET
100	CLRV				AND #imm	AND dp	AND dp+X	AND !abs	INC A	INC dp	TCALL 8	AND1 AND1B	CMPY dp	CBNE dp+X	TXSP	INC X
101	SETC				EOR #imm	EOR dp	EOR dp+X	EOR !abs	DEC A	DEC dp	TCALL 10	EOR1 EOR1B	DBNE dp	XMA dp+X	TSPX	DEC X
110	SETG				LDA #imm	LDA dp	LDA dp+X	LDA !abs	TXA	LDY dp	TCALL 12	LDC LDCB	LDX dp	LDX dp+Y	XCN	DAS
111	EI				LDM dp,#imm	STA dp	STA dp+X	STA !abs	TAX	STY dp	TCALL 14	STC M.bit	STX dp	STX dp+Y	XAX	STOP

LOW / HIGH	10000 10	10001 11	10010 12	10011 13	10100 14	10101 15	10110 16	10111 17	11000 18	11001 19	11010 1A	11011 1B	11100 1C	11101 1D	11110 1E	11111 1F
000	BPL rel	CLR1 dp.bit	BBC A.bit,rel	BBC dp.bit,rel	ADC {X}	ADC !abs+Y	ADC [dp+X]	ADC [dp]+Y	ASL !abs	ASL dp+X	TCALL 1	JMP !abs	BIT !abs	ADDW dp	LDX #imm	JMP [!abs]
001	BVC rel				SBC {X}	SBC !abs+Y	SBC [dp+X]	SBC [dp]+Y	ROL !abs	ROL dp+X	TCALL 3	CALL !abs	TEST !abs	SUBW dp	LDY #imm	JMP [dp]
010	BCC rel				CMP {X}	CMP !abs+Y	CMP [dp+X]	CMP [dp]+Y	LSR !abs	LSR dp+X	TCALL 5	MUL	TCLR1 !abs	CMPW dp	CMPX #imm	CALL [dp]
011	BNE rel				OR {X}	OR !abs+Y	OR [dp+X]	OR [dp]+Y	ROR !abs	ROR dp+X	TCALL 7	DBNE Y	CMPX !abs	LDYA dp	CMPY #imm	RETI
100	BMI rel				AND {X}	AND !abs+Y	AND [dp+X]	AND [dp]+Y	INC !abs	INC dp+X	TCALL 9	DIV	CMPY !abs	INCW dp	INC Y	TAY
101	BVS rel				EOR {X}	EOR !abs+Y	EOR [dp+X]	EOR [dp]+Y	DEC !abs	DEC dp+X	TCALL 11	XMA {X}	XMA dp	DECW dp	DEC Y	TYA
110	BCS rel				LDA {X}	LDA !abs+Y	LDA [dp+X]	LDA [dp]+Y	LDY !abs	LDY dp+X	TCALL 13	LDA {X}+	LDX !abs	STYA dp	XAY	DAA
111	BEQ rel				STA {X}	STA !abs+Y	STA [dp+X]	STA [dp]+Y	STY !abs	STY dp+X	TCALL 15	STA {X}+	STX !abs	CBNE dp	XYX	NOP

B.3 Instruction Set

Arithmetic / Logic Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	ADC #imm	04	2	2	Add with carry.	
2	ADC dp	05	2	3	A ← (A) + (M) + C	
3	ADC dp + X	06	2	4		
4	ADC !abs	07	3	4		NV--H-ZC
5	ADC !abs + Y	15	3	5		
6	ADC [dp + X]	16	2	6		
7	ADC [dp] + Y	17	2	6		
8	ADC { X }	14	1	3		
9	AND #imm	84	2	2	Logical AND	
10	AND dp	85	2	3	A ← (A) ∧ (M)	
11	AND dp + X	86	2	4		
12	AND !abs	87	3	4		N-----Z-
13	AND !abs + Y	95	3	5		
14	AND [dp + X]	96	2	6		
15	AND [dp] + Y	97	2	6		
16	AND { X }	94	1	3		
17	ASL A	08	1	2	Arithmetic shift left	N-----ZC
18	ASL dp	09	2	4	C 7 6 5 4 3 2 1 0	
19	ASL dp + X	19	2	5	□ ← ←←←←←←←← ← "0"	
20	ASL !abs	18	3	5		
21	CMP #imm	44	2	2		
22	CMP dp	45	2	3		
23	CMP dp + X	46	2	4		
24	CMP !abs	47	3	4	Compare accumulator contents with memory contents (A) - (M)	N-----ZC
25	CMP !abs + Y	55	3	5		
26	CMP [dp + X]	56	2	6		
27	CMP [dp] + Y	57	2	6		
28	CMP { X }	54	1	3		
29	CMPX #imm	5E	2	2	Compare X contents with memory contents	
30	CMPX dp	6C	2	3	(X) - (M)	N-----ZC
31	CMPX !abs	7C	3	4		
32	CMPY #imm	7E	2	2	Compare Y contents with memory contents	
33	CMPY dp	8C	2	3	(Y) - (M)	N-----ZC
34	CMPY !abs	9C	3	4		
35	COM dp	2C	2	4	1'S Complement : (dp) ← ~(dp)	N-----Z-
36	DAA	DF	1	3	Decimal adjust for addition	N-----ZC
37	DAS	CF	1	3	Decimal adjust for subtraction	N-----ZC
38	DEC A	A8	1	2	Decrement	N-----Z-
39	DEC dp	A9	2	4	M ← (M) - 1	N-----Z-
40	DEC dp + X	B9	2	5		N-----Z-
41	DEC !abs	B8	3	5		N-----Z-
42	DEC X	AF	1	2		N-----Z-
43	DEC Y	BE	1	2		N-----Z-

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
44	DIV	9B	1	12	Divide : YA / X Q: A, R: Y	NV--H-Z-
45	EOR #imm	A4	2	2	Exclusive OR	
46	EOR dp	A5	2	3	A ← (A) ⊕ (M)	
47	EOR dp + X	A6	2	4		
48	EOR !abs	A7	3	4		N-----Z-
49	EOR !abs + Y	B5	3	5		
50	EOR [dp + X]	B6	2	6		
51	EOR [dp] + Y	B7	2	6		
52	EOR { X }	B4	1	3		
53	INC A	88	1	2	Increment	N-----ZC
54	INC dp	89	2	4	M ← (M) + 1	N-----Z-
55	INC dp + X	99	2	5		N-----Z-
56	INC !abs	98	3	5		N-----Z-
57	INC X	8F	1	2		N-----Z-
58	INC Y	9E	1	2		N-----Z-
59	LSR A	48	1	2	Logical shift right	
60	LSR dp	49	2	4		N-----ZC
61	LSR dp + X	59	2	5		
62	LSR !abs	58	3	5		
63	MUL	5B	1	9	Multiply : YA ← Y × A	N-----Z-
64	OR #imm	64	2	2	Logical OR	
65	OR dp	65	2	3	A ← (A) ∨ (M)	
66	OR dp + X	66	2	4		
67	OR !abs	67	3	4		N-----Z-
68	OR !abs + Y	75	3	5		
69	OR [dp + X]	76	2	6		
70	OR [dp] + Y	77	2	6		
71	OR { X }	74	1	3		
72	ROL A	28	1	2	Rotate left through Carry	
73	ROL dp	29	2	4		N-----ZC
74	ROL dp + X	39	2	5		
75	ROL !abs	38	3	5		
76	ROR A	68	1	2	Rotate right through Carry	
77	ROR dp	69	2	4		N-----ZC
78	ROR dp + X	79	2	5		
79	ROR !abs	78	3	5		
80	SBC #imm	24	2	2	Subtract with Carry	
81	SBC dp	25	2	3	A ← (A) - (M) - ~(C)	
82	SBC dp + X	26	2	4		
83	SBC !abs	27	3	4		NV--HZC
84	SBC !abs + Y	35	3	5		
85	SBC [dp + X]	36	2	6		
86	SBC [dp] + Y	37	2	6		
87	SBC { X }	34	1	3		
88	TST dp	4C	2	3	Test memory contents for negative or zero, (dp) - 00_H	N-----Z-
89	XCN	CE	1	5	Exchange nibbles within the accumulator A_7~A_4 ↔ A_3~A_0	N-----Z-

Register / Memory Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	LDA #imm	C4	2	2	Load accumulator	N-----Z-
2	LDA dp	C5	2	3	A ← (M)	
3	LDA dp + X	C6	2	4		
4	LDA !abs	C7	3	4		
5	LDA !abs + Y	D5	3	5		
6	LDA [dp + X]	D6	2	6		
7	LDA [dp] + Y	D7	2	6		
8	LDA { X }	D4	1	3		
9	LDA { X }+	DB	1	4	X- register auto-increment : A ← (M) , X ← X + 1	
10	LDM dp,#imm	E4	3	5	Load memory with immediate data : (M) ← imm	--------
11	LDX #imm	1E	2	2	Load X-register	N-----Z-
12	LDX dp	CC	2	3	X ← (M)	
13	LDX dp + Y	CD	2	4		
14	LDX !abs	DC	3	4		
15	LDY #imm	3E	2	2	Load Y-register	N-----Z-
16	LDY dp	C9	2	3	Y ← (M)	
17	LDY dp + X	D9	2	4		
18	LDY !abs	D8	3	4		
19	STA dp	E5	2	4	Store accumulator contents in memory	--------
20	STA dp + X	E6	2	5	(M) ← A	
21	STA !abs	E7	3	5		
22	STA !abs + Y	F5	3	6		
23	STA [dp + X]	F6	2	7		
24	STA [dp] + Y	F7	2	7		
25	STA { X }	F4	1	4		
26	STA { X }+	FB	1	4	X- register auto-increment : (M) ← A, X ← X + 1	
27	STX dp	EC	2	4	Store X-register contents in memory	--------
28	STX dp + Y	ED	2	5	(M) ← X	
29	STX !abs	FC	3	5		
30	STY dp	E9	2	4	Store Y-register contents in memory	--------
31	STY dp + X	F9	2	5	(M) ← Y	
32	STY !abs	F8	3	5		
33	TAX	E8	1	2	Transfer accumulator contents to X-register : X ← A	N-----Z-
34	TAY	9F	1	2	Transfer accumulator contents to Y-register : Y ← A	N-----Z-
35	TSPX	AE	1	2	Transfer stack-pointer contents to X-register : X ← sp	N-----Z-
36	TXA	C8	1	2	Transfer X-register contents to accumulator: A ← X	N-----Z-
37	TXSP	8E	1	2	Transfer X-register contents to stack-pointer: sp ← X	N-----Z-
38	TYA	BF	1	2	Transfer Y-register contents to accumulator: A ← Y	N-----Z-
39	XAX	EE	1	4	Exchange X-register contents with accumulator :X ↔ A	--------
40	XAY	DE	1	4	Exchange Y-register contents with accumulator :Y ↔ A	--------
41	XMA dp	BC	2	5	Exchange memory contents with accumulator	N-----Z-
42	XMA dp+X	AD	2	6	(M) ↔ A	
43	XMA {X}	BB	1	5		
44	XYX	FE	1	4	Exchange X-register contents with Y-register : X ↔ Y	--------

16-BIT operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	ADDW dp	1D	2	5	16-Bits add without Carry YA ← (YA) + (dp +1) (dp)	NV--H-ZC
2	CMPW dp	5D	2	4	Compare YA contents with memory pair contents : (YA) – (dp+1)(dp)	N-----ZC
3	DECW dp	BD	2	6	Decrement memory pair (dp+1)(dp) ← (dp+1) (dp) - 1	N-----Z-
4	INCW dp	9D	2	6	Increment memory pair (dp+1) (dp) ← (dp+1) (dp) + 1	N-----Z-
5	LDYA dp	7D	2	5	Load YA YA ← (dp +1) (dp)	N-----Z-
6	STYA dp	DD	2	5	Store YA (dp +1) (dp) ← YA	--------
7	SUBW dp	3D	2	5	16-Bits subtract without carry YA ← (YA) - (dp +1) (dp)	NV--H-ZC

Bit Manipulation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	AND1 M.bit	8B	3	4	Bit AND C-flag : C ← (C) ∧ (M .bit)	-------C
2	AND1B M.bit	8B	3	4	Bit AND C-flag and NOT : C ← (C) ∧ ~(M .bit)	-------C
3	BIT dp	0C	2	4	Bit test A with memory :	MM----Z-
4	BIT !abs	1C	3	5	Z ← (A) ∧ (M) , N ← (M_7) , V ← (M_6)	
5	CLR1 dp.bit	y1	2	4	Clear bit : (M.bit) ← "0"	--------
6	CLRA1 A.bit	2B	2	2	Clear A bit : (A.bit) ← "0"	--------
7	CLRC	20	1	2	Clear C-flag : C ← "0"	-------0
8	CLRG	40	1	2	Clear G-flag : G ← "0"	--0-----
9	CLRV	80	1	2	Clear V-flag : V ← "0"	-0--0---
10	EOR1 M.bit	AB	3	5	Bit exclusive-OR C-flag : C ← (C) ⊕ (M .bit)	-------C
11	EOR1B M.bit	AB	3	5	Bit exclusive-OR C-flag and NOT : C ← (C) ⊕ ~(M .bit)	-------C
12	LDC M.bit	CB	3	4	Load C-flag : C ← (M .bit)	-------C
13	LDCB M.bit	CB	3	4	Load C-flag with NOT : C ← ~(M .bit)	-------C
14	NOT1 M.bit	4B	3	5	Bit complement : (M .bit) ← ~(M .bit)	--------
15	OR1 M.bit	6B	3	5	Bit OR C-flag : C ← (C) ∨ (M .bit)	-------C
16	OR1B M.bit	6B	3	5	Bit OR C-flag and NOT : C ← (C) ∨ ~(M .bit)	-------C
17	SET1 dp.bit	x1	2	4	Set bit : (M.bit) ← "1"	--------
18	SETA1 A.bit	0B	2	2	Set A bit : (A.bit) ← "1"	--------
19	SETC	A0	1	2	Set C-flag : C ← "1"	-------1
20	SETG	C0	1	2	Set G-flag : G ← "1"	--1-----
21	STC M.bit	EB	3	6	Store C-flag : (M .bit) ← C	--------
22	TCLR1 !abs	5C	3	6	Test and clear bits with A : A - (M) , (M) ← (M) ∧ ~(A)	N-----Z-
23	TSET1 !abs	3C	3	6	Test and set bits with A : A - (M) , (M) ← (M) ∨ (A)	N-----Z-

Branch / Jump Operation

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	BBC A.bit,rel	y2	2	4/6	Branch if bit clear :	--------
2	BBC dp.bit,rel	y3	3	5/7	if (bit) = 0 , then pc ← (pc) + rel	
3	BBS A.bit,rel	x2	2	4/6	Branch if bit set :	--------
4	BBS dp.bit,rel	x3	3	5/7	if (bit) = 1 , then pc ← (pc) + rel	
5	BCC rel	50	2	2/4	Branch if carry bit clear if (C) = 0 , then pc ← (pc) + rel	--------
6	BCS rel	D0	2	2/4	Branch if carry bit set if (C) = 1 , then pc ← (pc) + rel	--------
7	BEQ rel	F0	2	2/4	Branch if equal if (Z) = 1 , then pc ← (pc) + rel	--------
8	BMI rel	90	2	2/4	Branch if minus if (N) = 1 , then pc ← (pc) + rel	--------
9	BNE rel	70	2	2/4	Branch if not equal if (Z) = 0 , then pc ← (pc) + rel	--------
10	BPL rel	10	2	2/4	Branch if minus if (N) = 0 , then pc ← (pc) + rel	--------
11	BRA rel	2F	2	4	Branch always pc ← (pc) + rel	--------
12	BVC rel	30	2	2/4	Branch if overflow bit clear if (V) = 0 , then pc ← (pc) + rel	--------
13	BVS rel	B0	2	2/4	Branch if overflow bit set if (V) = 1 , then pc ← (pc) + rel	--------
14	CALL !abs	3B	3	8	Subroutine call	
15	CALL [dp]	5F	2	8	M(sp)←(pc_H), sp←sp - 1, M(sp)← (pc_L), sp ←sp - 1, if !abs, pc← abs ; if [dp], pc_L← (dp), pc_H← (dp+1) .	--------
16	CBNE dp,rel	FD	3	5/7	Compare and branch if not equal :	--------
17	CBNE dp+X,rel	8D	3	6/8	if (A) ≠ (M) , then pc ← (pc) + rel.	
18	DBNE dp,rel	AC	3	5/7	Decrement and branch if not equal :	--------
19	DBNE Y,rel	7B	2	4/6	if (M) ≠ 0 , then pc ← (pc) + rel.	
20	JMP !abs	1B	3	3	Unconditional jump	
21	JMP [!abs]	1F	3	5	pc ← jump address	--------
22	JMP [dp]	3F	2	4		
23	PCALL upage	4F	2	6	U-page call M(sp) ←(pc_H), sp ←sp - 1, M(sp) ← (pc_L), sp ← sp - 1, pc_L ← (upage), pc_H ← "$0FF_H$" .	--------
24	TCALL n	nA	1	8	Table call : (sp) ←(pc_H), sp ← sp - 1, M(sp) ← (pc_L),sp ← sp - 1, pc_L ← (Table vector L), pc_H ← (Table vector H)	--------

Control Operation & Etc.

No.	Mnemonic	Op Code	Byte No	Cycle No	Operation	Flag NVGBHIZC
1	BRK	0F	1	8	Software interrupt : B ← "1", M(sp) ← (pc_H), sp ←sp-1, M(s) ← (pc_L), sp ← sp - 1, M(sp) ← (PSW), sp ← sp -1, pc_L ← ($0FFDE_H$) , pc_H ← ($0FFDF_H$) .	---1-0--
2	DI	60	1	3	Disable all interrupts : I ← "0"	-----0--
3	EI	E0	1	3	Enable all interrupt : I ← "1"	-----1--
4	NOP	FF	1	2	No operation	--------
5	POP A	0D	1	4	sp ← sp + 1, A ← M(sp)	--------
6	POP X	2D	1	4	sp ← sp + 1, X ← M(sp)	
7	POP Y	4D	1	4	sp ← sp + 1, Y ← M(sp)	
8	POP PSW	6D	1	4	sp ← sp + 1, PSW ← M(sp)	restored
9	PUSH A	0E	1	4	M(sp) ← A , sp ← sp - 1	--------
10	PUSH X	2E	1	4	M(sp) ← X , sp ← sp - 1	
11	PUSH Y	4E	1	4	M(sp) ← Y , sp ← sp - 1	
12	PUSH PSW	6E	1	4	M(sp) ← PSW , sp ← sp - 1	
13	RET	6F	1	5	Return from subroutine sp ← sp +1, pc_L ← M(sp), sp ← sp +1, pc_H ← M(sp)	--------
14	RETI	7F	1	6	Return from interrupt sp ← sp +1, PSW ← M(sp), sp ← sp + 1, pc_L ← M(sp), sp ← sp + 1, pc_H ← M(sp)	restored
15	STOP	EF	1	3	Stop mode (halt CPU, stop oscillator)	--------

C. MASK ORDER SHEET

MASK ORDER & VERIFICATION SHEET

GMS81C5108 - UD☐☐☐

Customer should write inside thick line box.

1. Customer Information

Company Name	
Application	
Order Date	YYYY . MM . DD

Tel: Fax:

E-mail address:

Name & Signature:

2. Device Information

Package	80QFP	
ROM Size	8K	
OSC Option	☐ Crystal	☐ R
Reset Pull Up	☐ YES	☐ NO

Mask Data: ☐ Hitel ☐ Chollian ☐ Internet

File Name : (.OTP)

Check Sum : ( )

Address	
0000_H – $DFFF_H$	Set "00" in this area
$E000_H$ – $FFFF_H$	.OTP file data

3. Marking Specification *(Please check mark into ☐)*

☐
hynix
GMS81C5108-UD☐☐☐
YYWW KOREA

☐
Customer's Area
GMS81C5108-UD☐☐☐
YYWW KOREA

If the customer logo & part number must be used in this area, please submit a clean original logo & part number.

4. Delivery Schedule

	Date	Quantity	HYNIX Confirmation
Customer sample	YYYY . MM . DD	pcs	
Risk order	YYYY . MM . DD	pcs	

5. ROM Code Verification

Please confirm out verification data.

Verification date: YYYY . MM . DD
Check sum:

Tel: Fax:

E-mail address:

Name & Signature:

Approval date: YYYY . MM . DD

I agree with your verification data and confirm you to make mask set.

Tel: Fax:

E-mail address:

Name & Signature: