KS57C3108
4-BIT CMOS Microcontroller

Product Specification

OVERVIEW

The KS57C3108 single-chip CMOS microcontroller has been designed for very high performance using Samsung's newest 4-bit CPU core, SAM4 (Samsung Arrangeable Microcontrollers). With an up-to-14-digit LCD direct drive capability, a 4-channel A/D converter, 8-bit timer/counter, and PLL frequency synthesizer, the KS57C3108 offers you an excellent design solution for a wide variety of DTS applications.

Up to 40 pins of the 80-pin QFP package can be dedicated to I/O. Eight vectored interrupts provide fast response to internal and external events. In addition, the KS57C3108's advanced CMOS technology provides for low power consumption and a wide operating voltage range.

FEATURES

Memory

- 512 × 4-bit RAM
- 8,192 × 8-bit ROM

56 I/O pins

- Input only: 4 pins
- Output only: 28 pins
- I/O: 24 pins

LCD Controller/Driver

- Maximum 14-digit LCD direct drive capability
- 28 segments × 4 common signals
- Display modes: Static, 1/2 duty (1/2 bias), 1/3 duty (1/2 or 1/3 bias), 1/4 duty (1/3 bias)

8-bit Basic Timer

- 4 interval timer functions

8-bit Timer/Counter

- Programmable 8-bit timer
- External event counter
- Arbitrary clock frequency output
- External clock signal divider
- Serial I/O interface clock generator

Watch Timer

- Time interval generation: 0.5 s, 3.9 ms at 32768 Hz
- 4 frequency outputs to BUZ pin
- Clock source generation for LCD

8-bit Serial I/O Interface

- 8-bit transmit/receive mode
- 8-bit receive mode
- LSB-first or MSB-first transmission selectable
- Internal or external clock source

A/D Converter

- 4 channels with 8-bit resolution
- 17.8 µs conversion speed at 4.5 MHz

Bit Sequential Carrier

- Support 16-bit serial data transfer in arbitrary format

PLL Frequency Synthesizer

- Level = 300 mVp-p (min)
- AMVCO range = 0.1 MHz to 30 MHz
- FMVCO range = 30 MHz to 150 MHz

16-Bit Intermediate Frequency (If) Counter

- Level = 300 mVp-p (min)
- AMIF range = 100 kHz to 1.0 MHz
- FMIF range = 5 MHz to 15 MHz

Interrupts

- Four internal vectored interrupts
- Four external vectored interrupts
- Two quasi-interrupts

Memory-Mapped I/O Structure

- Data memory bank 15

Four Power-Down Modes

- Idle: Only CPU clock stops
- Stop 1: Main system clock stops
- Stop 2: Main and subsystem clocks stop
- CE low: PLL stops

Oscillation Sources

- Crystal, ceramic, main system clock
- Crystal oscillator for subsystem clock
- Main system clock frequency: 4.5 MHz
- Subsystem clock frequency: 32.768 kHz
- CPU clock divider circuit (by 4, 8, or 64)

Instruction Execution Times

- 0.9, 1.8, 14.2 µs at 4.5 MHz
- 122 µs at 32.768 kHz

Operating Temperature

- – 40 °C to 85 °C

Operating Voltage Range

- 2.7 V to 6.0 V (4.0 V to 6.0 V in PLL mode)

Package Type

- 80-pin QFP

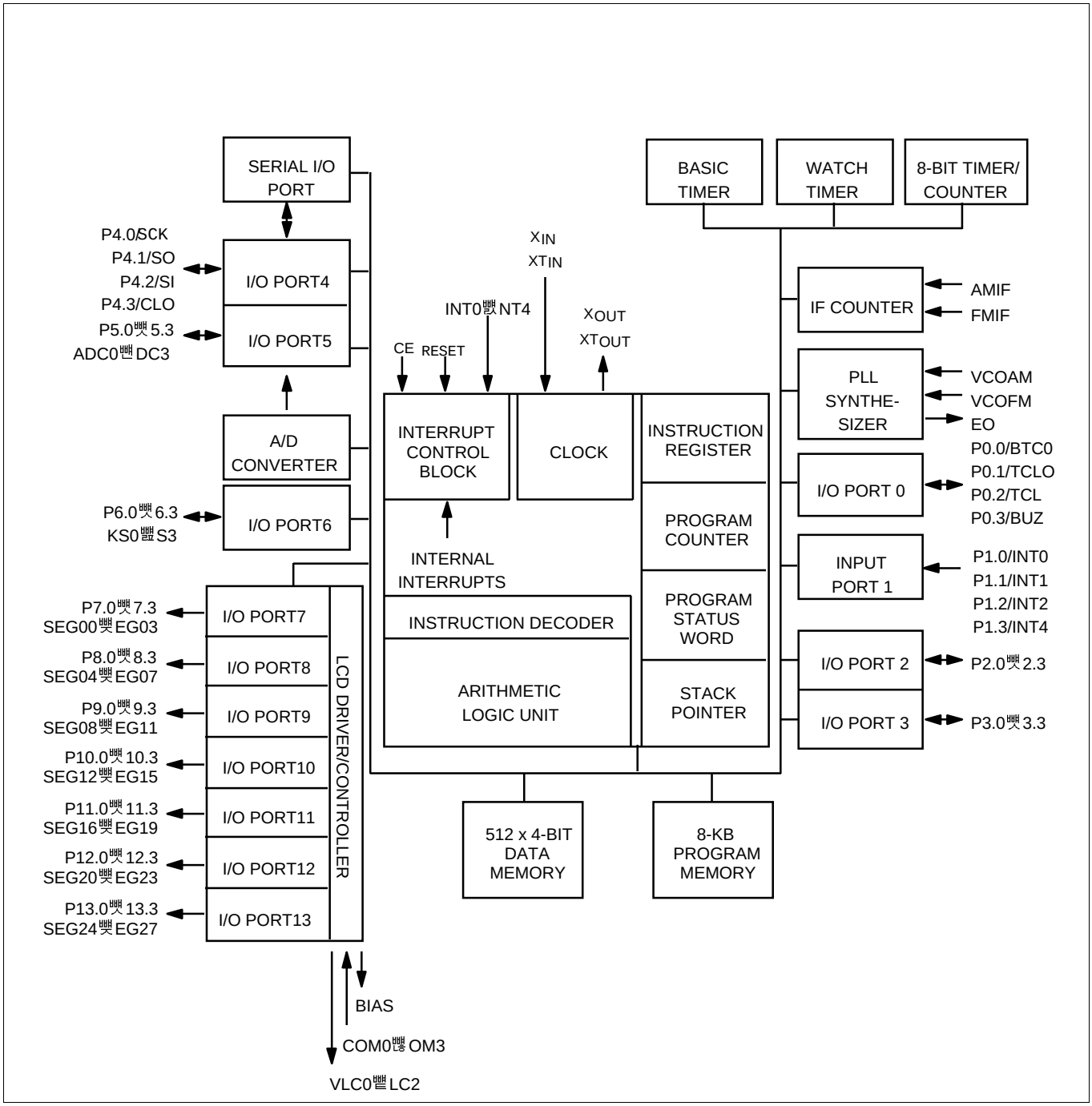


Figure 1. KS57C3108 Block Diagram

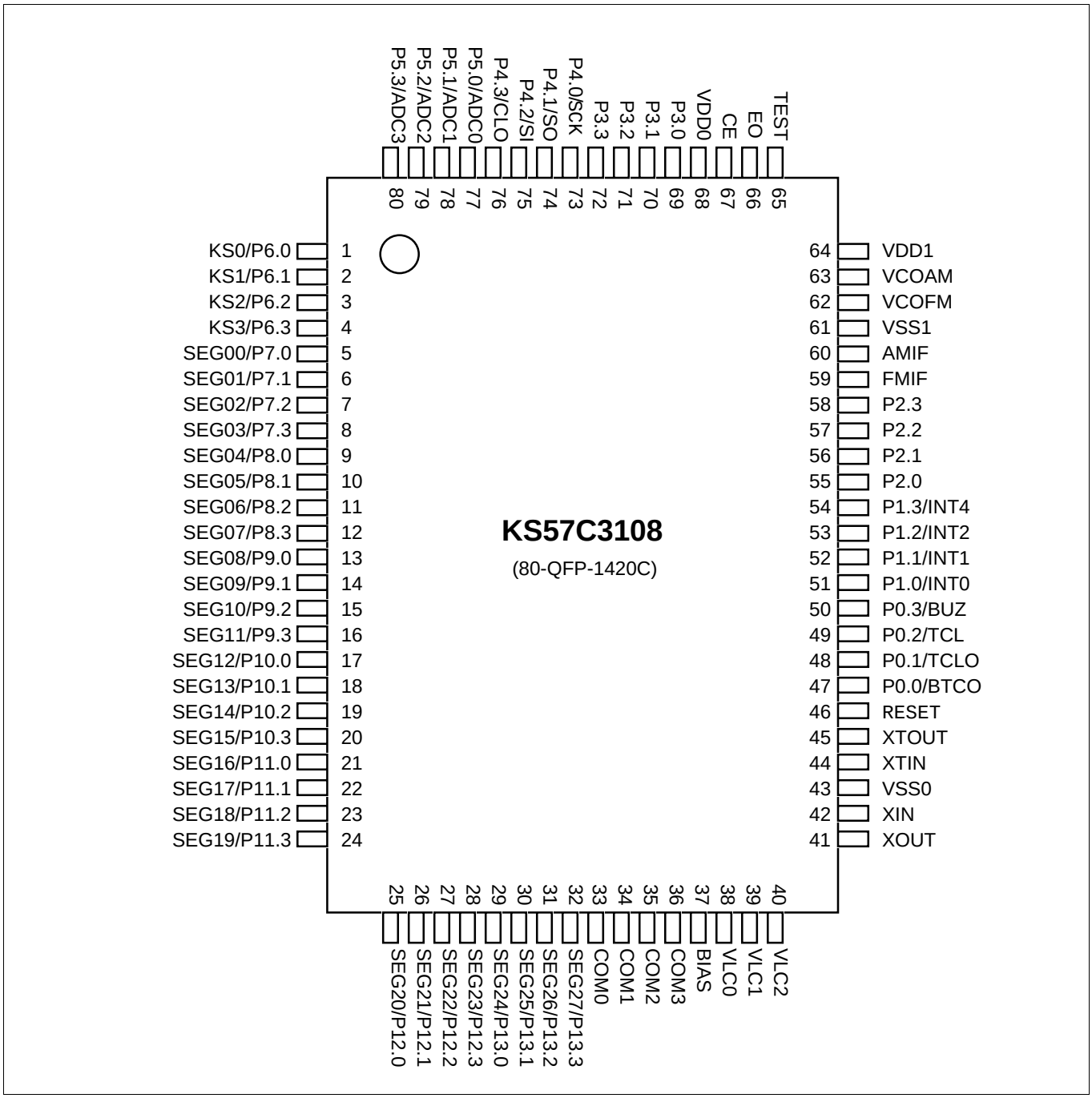


Figure 2. KS57C3108 (80-pin QFP) Pin Assignment Diagram

Table 1. KS57C3108 Pin Descriptions

Pin Name	Pin Type	Description	Number	Share Pin
P0.0 P0.1 P0.2 P0.3	I/O	4-bit I/O port. 1-bit or 4-bit read/write/test is possible. Pull-up resistors can be configured by software	47 48 49 50	BTCO TCLO TCL BUZ
P1.0 P1.1 P1.2 P1.3	I	4-bit input port. 1-bit or 4-bit read and test is possible. Pull-up resistors can be configured to P1.0, P1.1, and P1.2.	51 52 53 54	INT0 INT1 INT2 INT4
P2.0 – P2.3 P3.0 – P3.3	I/O	4-bit I/O port. 1-bit or 4-bit or 8-bit read/write/test are possible. Pull-up resistors can be configured by software. Ports 2 and 3 can be paired to support 8-bit data transfer.	55–58 69–72	—
P4.0 P4.1 P4.2 P4.3 P5.0 P5.1 P5.2 P5.3	I/O	4-bit I/O port. 1-bit or 4-bit or 8-bit read/write/test are possible. Pull-up resistors can be configured by software. Ports 4 and 5 can be paired to support 8-bit data transfer.	73 74 75 76 77 78 79 80	SCK SC SI CLO ADC0 ADC1 ADC2 ADC3
P6.0 P6.1 P6.2 P6.3	I/O	4-bit I/O port. 1-bit or 4-bit or 8-bit read/write/test is possible. Pull-up resistors can be configured by software.	1 2 3 4	KS0 KS1 KS2 KS3
P7.0 P7.1 P7.2 P7.3	O	4-bit output port. Alternatively used for LCD segment output.	5 6 7 8	SEG00 SEG01 SEG02 SEG03
P8.0 P8.1 P8.2 P8.3	O	4-bit output port. Alternatively used for LCD segment output.	9 10 11 12	SEG04 SEG05 SEG06 SEG07
P9.0 P9.1 P9.2 P9.3	O	4-bit output port. Alternatively used for LCD segment output.	13 14 15 16	SEG08 SEG09 SEG10 SEG11
P10.0 P10.1 P10.2 P10.3	O	4-bit output port. Alternatively used for LCD segment output.	17 18 19 20	SEG12 SEG13 SEG14 SEG15

Table 1. KS57C3108 Pin Descriptions (Continued)

Pin Name	Pin Type	Description	Number	Share Pin
P11.0 P11.1 P11.2 P11.3	O	4-bit output port. Alternatively used for LCD segment output.	21 22 23 24	SEG16 SEG17 SEG18 SEG19
P12.0 P12.1 P12.2 P12.3	O	4-bit output port. Alternatively used for LCD segment output.	25 26 27 28	SEG20 SEG21 SEG22 SEG23
P13.0 P13.1 P13.2 P13.3	O	4-bit output port. Alternatively used for LCD segment output.	29 30 31 32	SEG24 SEG25 SEG26 SEG27
COM0 COM1 COM2 COM3	O	Common signal output for LCD display	33 34 35 36	–
BIAS	I	LCD power control.	37	–
VLC0 VLC1 VLC2	I	LCD power supply. Voltage dividing resistors are assignable by software.	38 39 40	–
CE	I	Input pin for checking device power. Normal operation is High level and PLL operation stop is Low level.	67	–
V_{DD0}	–	Main power supply	68	–
V_{SS0}	–	Main ground pin	43	–
VCOFM VCOAM	I	External VCOFM/AM input pins	62 63	–
EO	O	Output pin for PLL error information	66	–
FMIF AMIF	I	FM/AM intermediate frequency input pins	59 60	–
V_{SS1}	–	PLL ground	61	–
V_{DD1}	–	PLL power supply	64	–
RESET	I	Chip reset signal pin	46	–
X_{OUT}	–	Main system clock output pin	41	–
X_{IN}	–	Main system clock input pin	42	–
XT_{IN}	–	Subsystem clock input pin	44	–
XT_{OUT}	–	Subsystem clock output pin	45	–
TEST	I	Chip test signal input (must be connected to V_{SS})	65	–

Table 2. Overview of KS57C3108 Pin Data

Pin Numbers	Pin Names	Share Pins	I/O Type	Reset Value	Circuit Type
47–49	P0.0–P0.1, P0.3	BTCO, TCLO, BUZ	I/O	Input	5
50	P0.2	TCL	I/O		6
51–53	P1.0–P1.2	INT0, INT1, INT2	I		3
54	P1.3	INT4	I		2
55–58	P2.0–P2.3	–	I/O		5
69–72	P3.0–P3.3	–			5
73–75	P4.0–P4.2	SCK, SO, SI			6
76	P4.3	CLO			5
77–80	P5.0–P5.3	AD0–AD3			7
1–4	P6.0–P6.3	KS0–KS3			14
5–8	P7.0–P7.3	SEG00–SEG03	O	Output	11
9–12	P8.0–P8.3	SEG04–SEG07			
13–16	P9.0–P9.3	SEG08–SEG11			
17–20	P10.0–P10.3	SEG12–SEG15			
21–24	P11.0–P11.3	SEG16–SEG19			
25–28	P12.0–P12.3	SEG20–SEG23			
29–32	P13.0–P13.3	SEG24–SEG27			
33–36	COM0–COM3	–			10
37	BIAS		I	–	–
38–40	VLC0–VLC2			–	–
67	CE			Input	12
68	V_{DD0}		–	–	–
43	V_{SS0}		–	–	–
62, 63	VCOFM, VCOAM		I	Input	8
66	EO		O	Output	9
59, 60	FMIF, AMIF		I	Input	8
61	V_{SS1}		–	–	–
64	V_{DD1}		–	–	–

Table 2. Overview of KS57C3108 Pin Data(Continued)

Pin Numbers	Pin Names	Share Pins	I/O Type	Reset Value	Circuit Type
46	RESET		–	Input	13
41	X_{OUT}		–	–	–
42	X_{IN}		–		
44	XT_{IN}		–		
45	XT_{OUT}		–		
65	TEST		I		

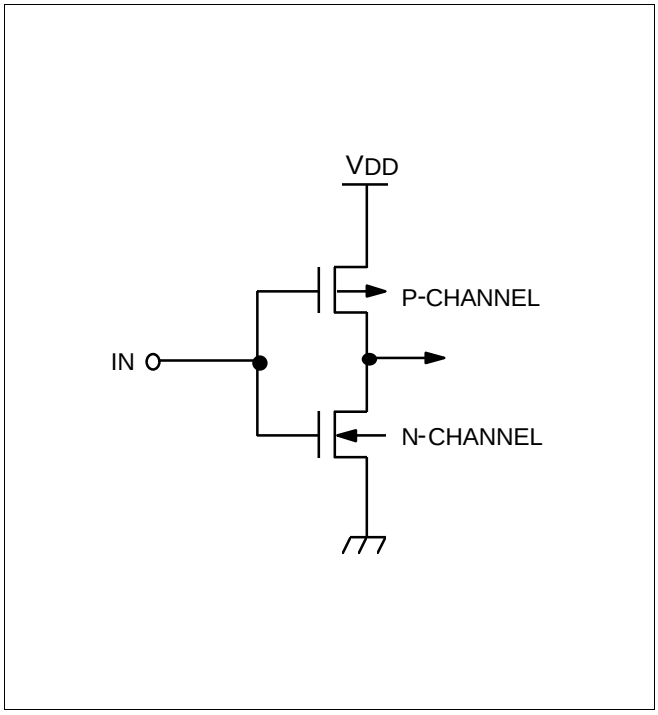
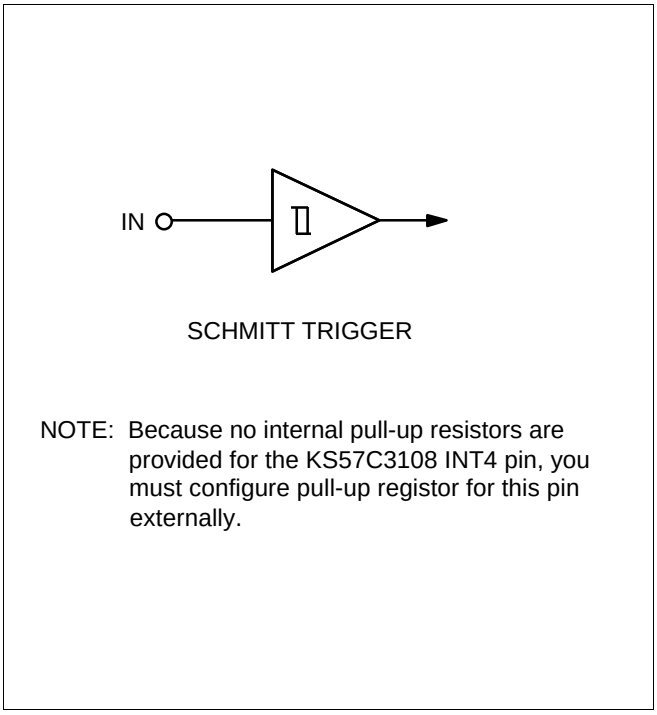


Figure 3. Pin Circuit Type 1



NOTE: Because no internal pull-up resistors are provided for the KS57C3108 INT4 pin, you must configure pull-up registor for this pin externally.

Figure 4. Pin Circuit Type 2

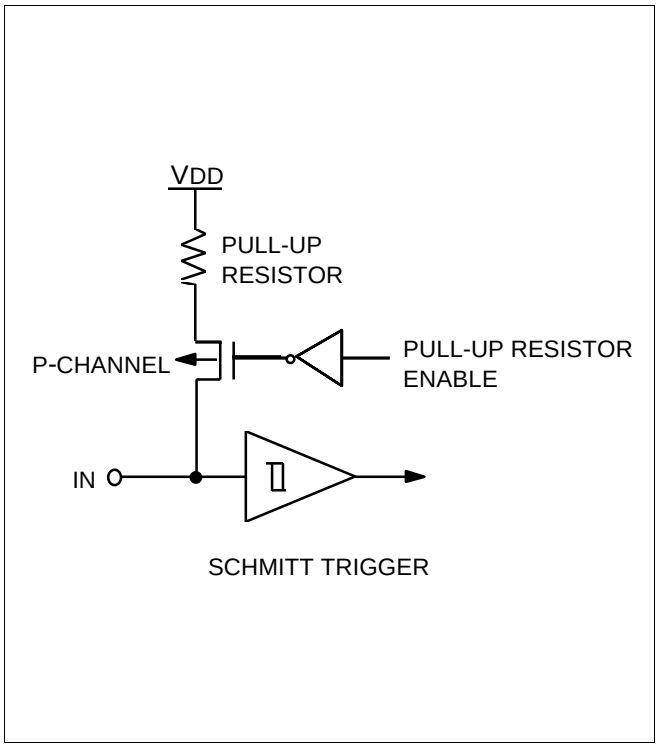


Figure 5. Pin Circuit Type 3

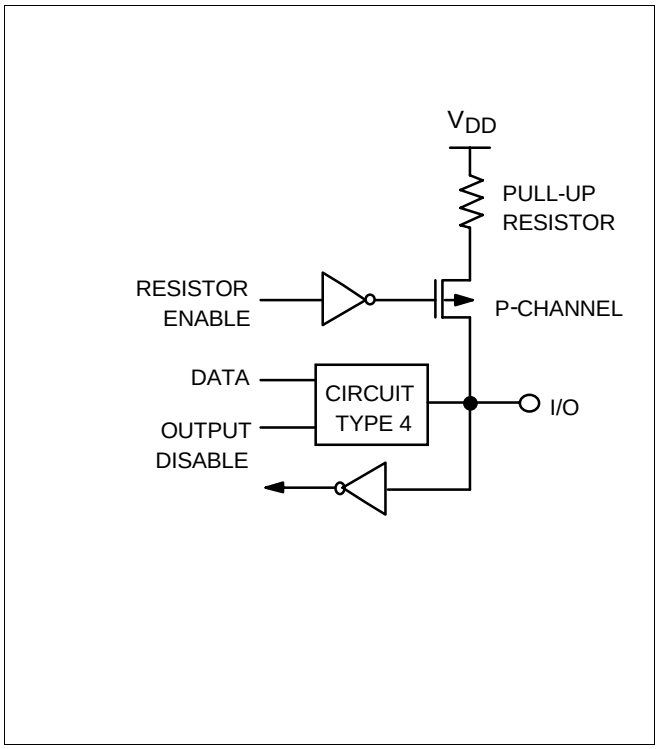


Figure 7. Pin Circuit Type 5

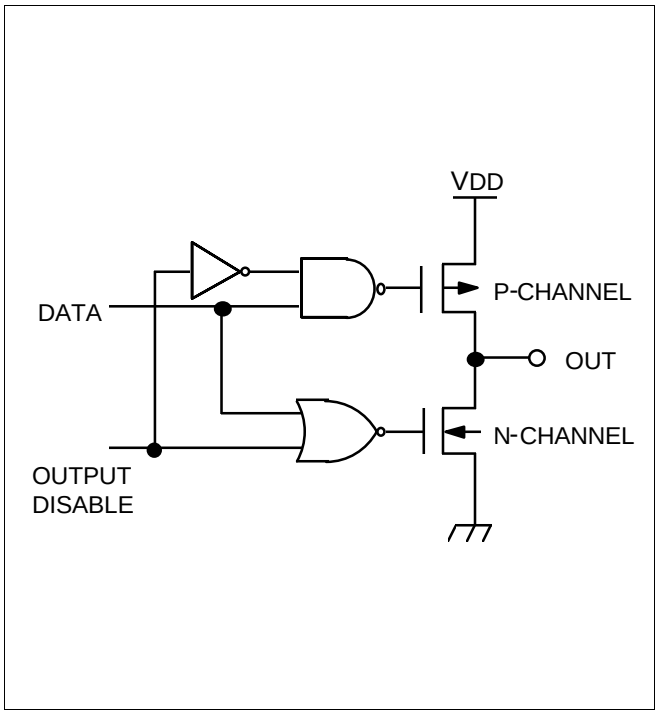


Figure 6. Pin Circuit Type 4

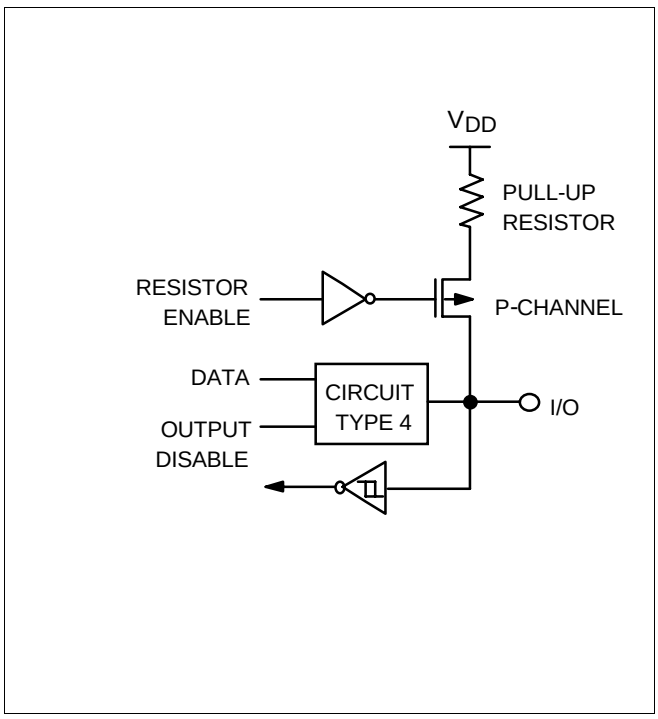


Figure 8. Pin Circuit Type 6

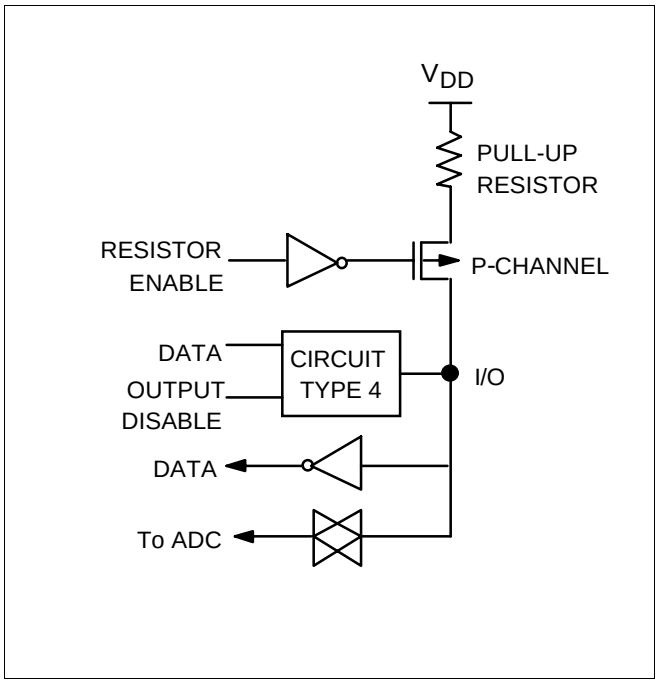


Figure 9. Pin Circuit Type 7

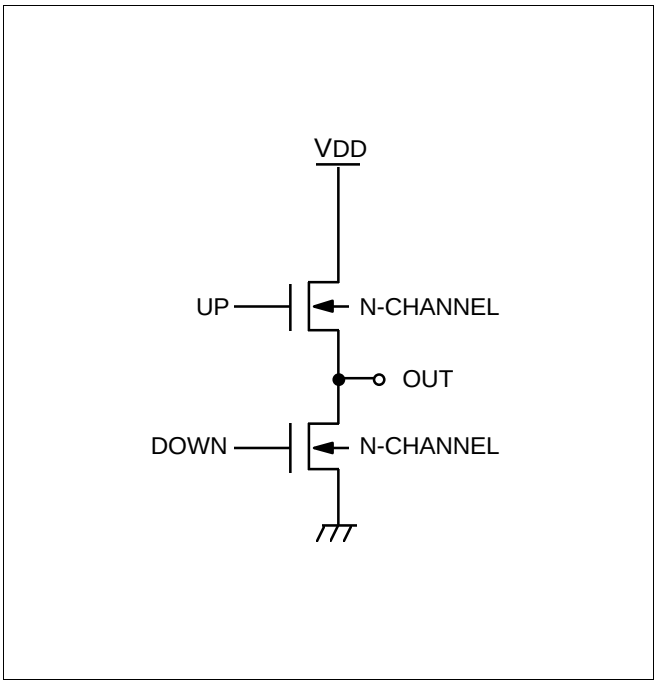


Figure 11. Pin Circuit Type 9

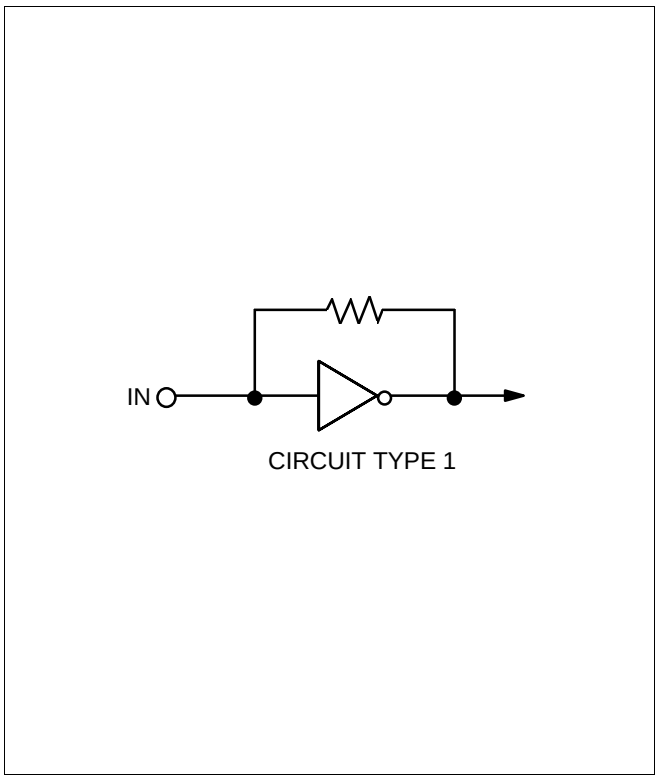


Figure 10. Pin Circuit Type 8

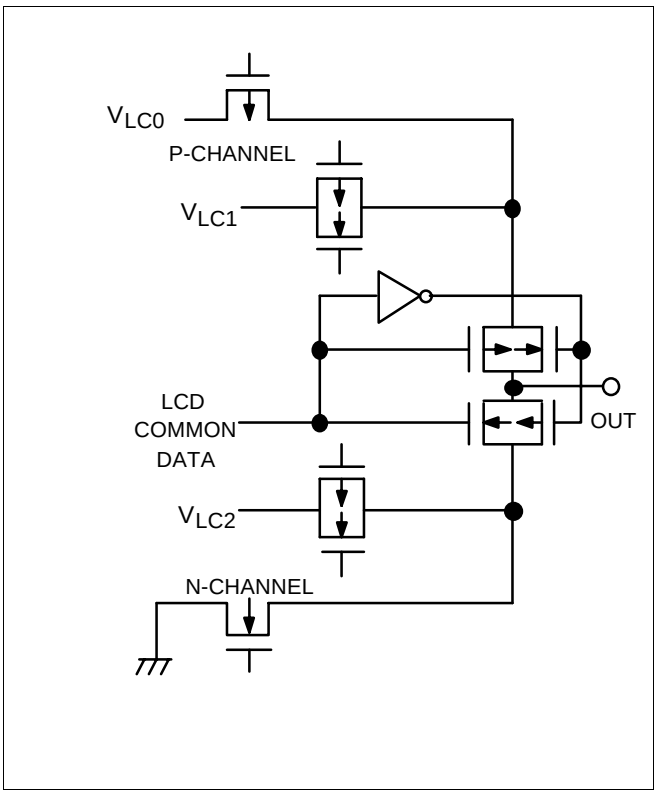


Figure 12. Pin Circuit Type 10

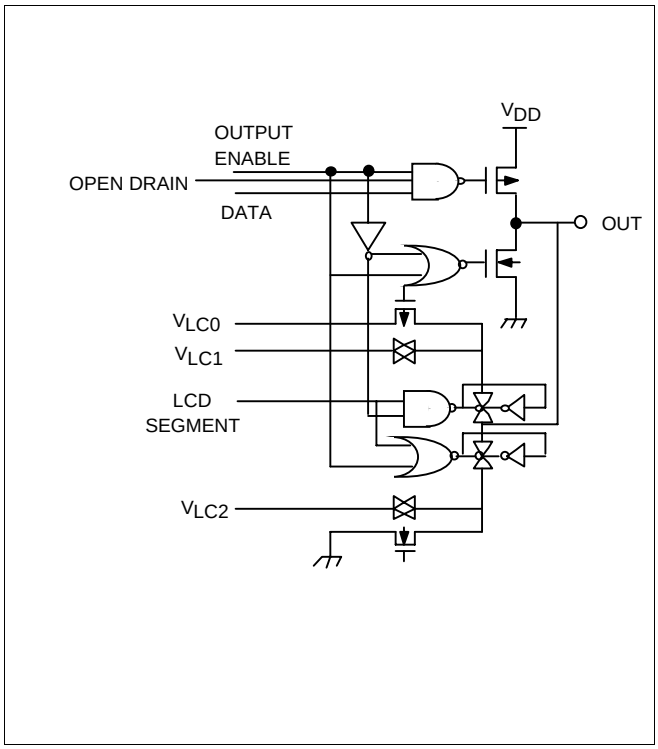


Figure 13. Pin Circuit Type 11

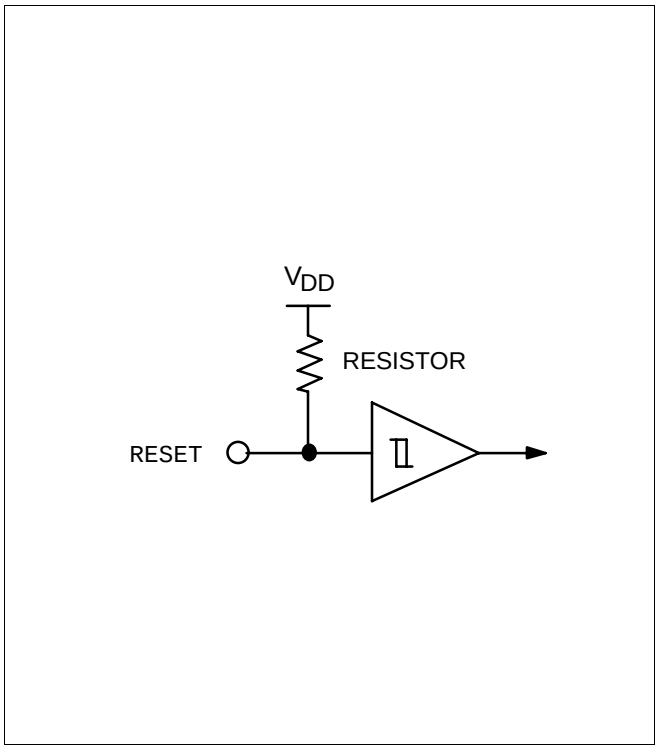


Figure 15. Pin Circuit Type 13

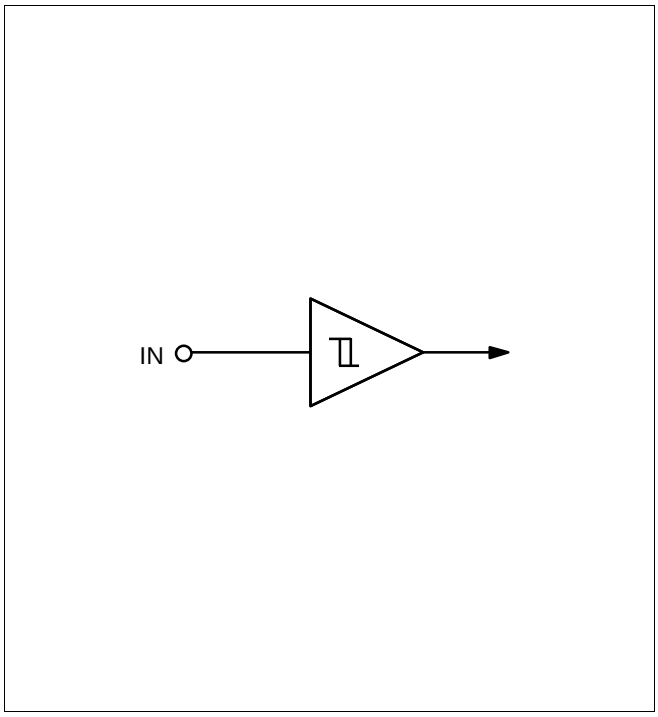


Figure 14. Pin Circuit Type 12

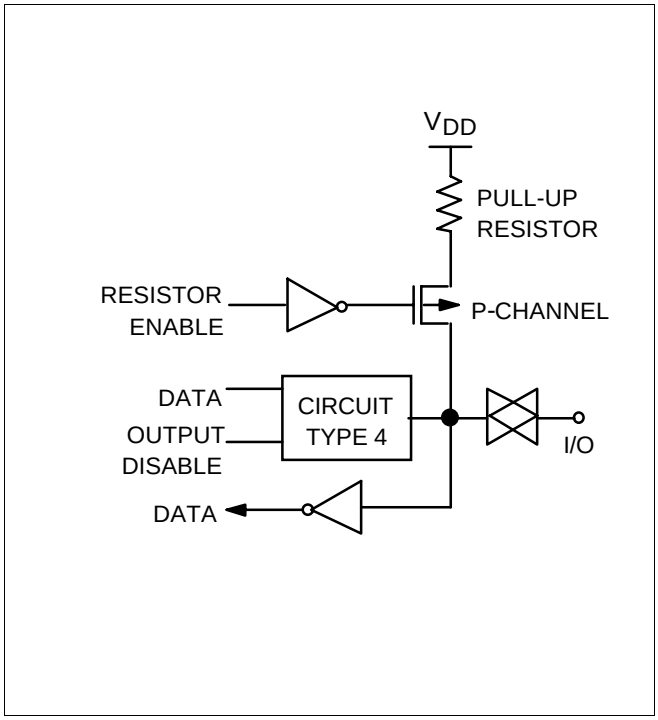


Figure 16. Pin Circuit Type 14

PROGRAM MEMORY (ROM)

ROM maps for KS57-series devices are mask programmed at the factory. The KS57C3108's standard 8192 × 8-bit program memory has four areas that are directly addressable by the program counter (PC):

- 16-byte area for vector addresses
- 16-byte general-purpose area
- 96-byte instruction reference area
- 8,064-byte general-purpose area

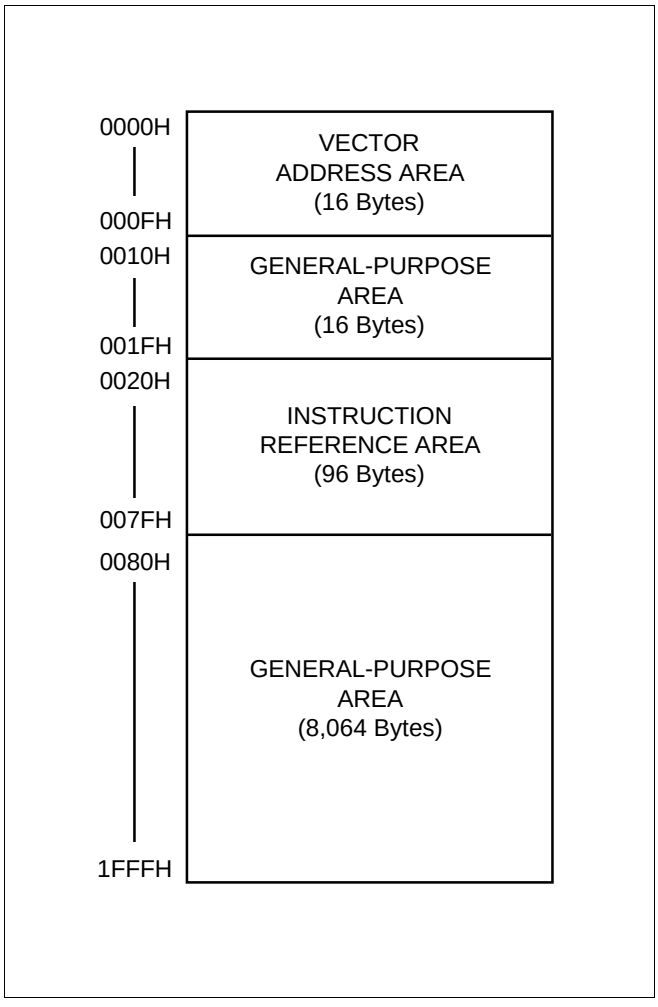


Figure 17. ROM Address Structure

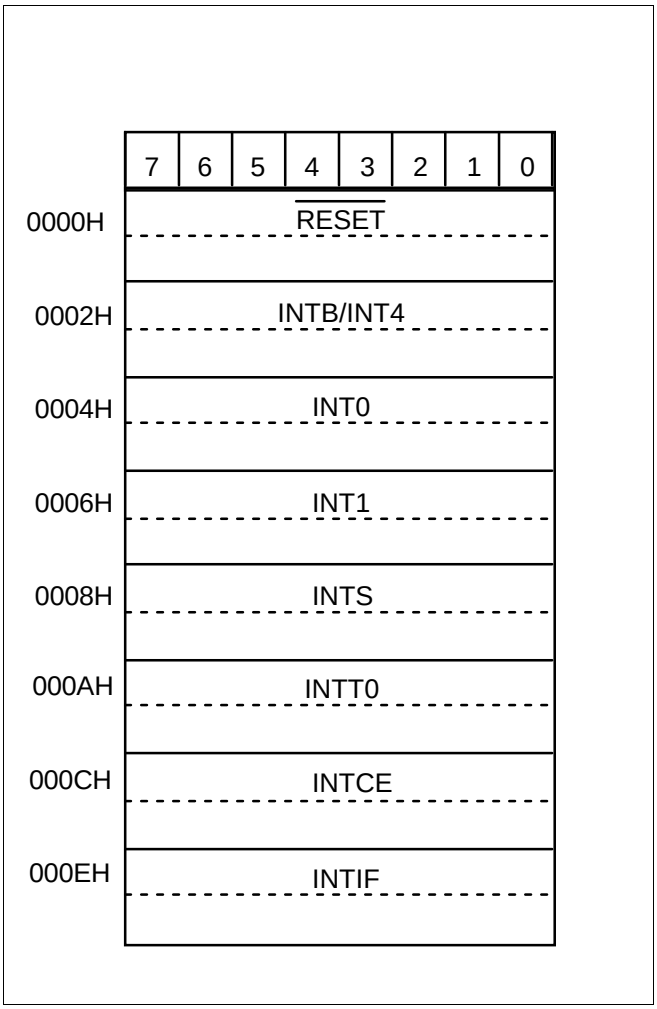


Figure 18. Vector Address Map

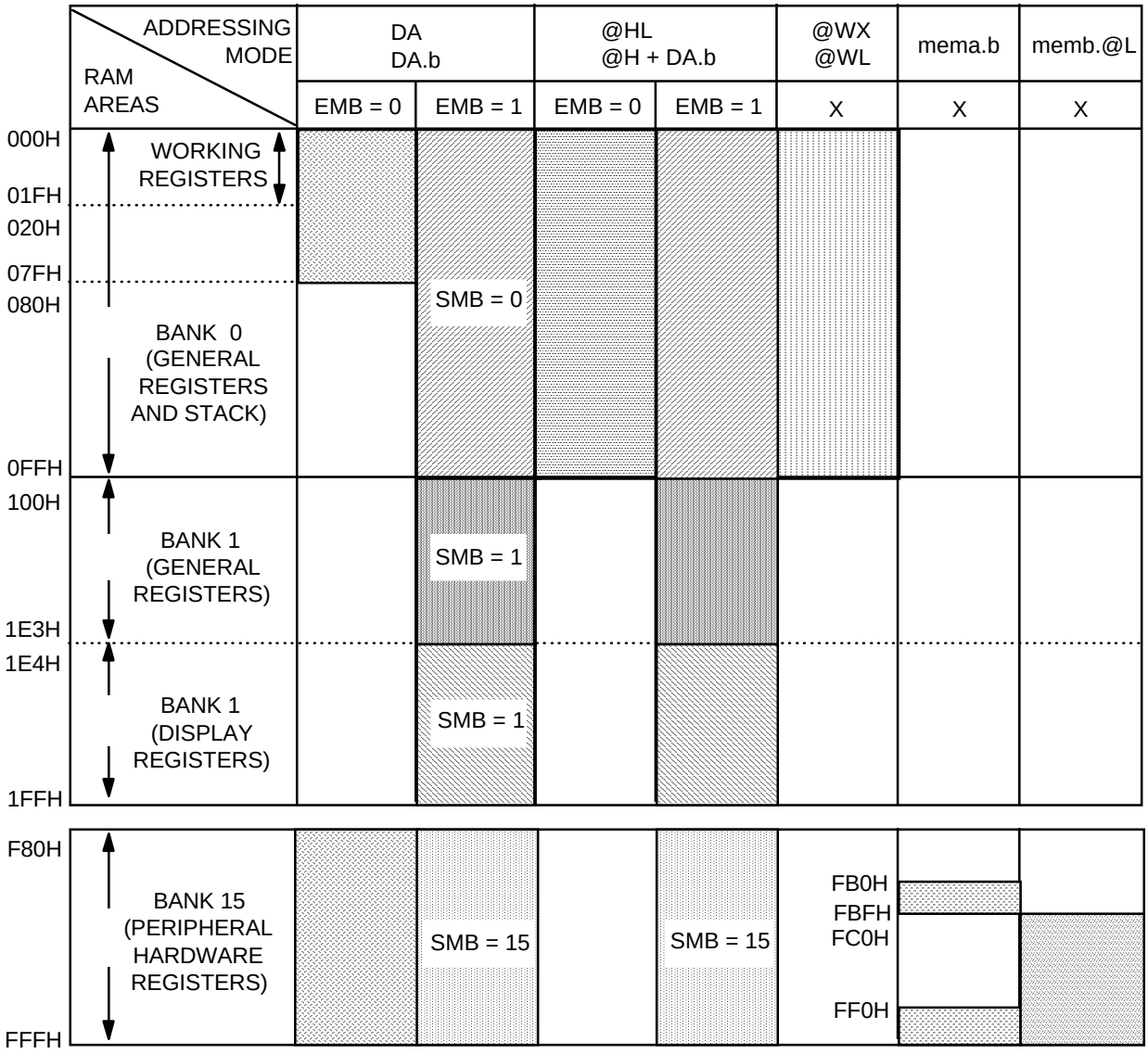
DATA MEMORY (RAM)

In its standard configuration, the 512荳 4-bit data memory has two areas:

- 32荳刊4-bit working register area
- 224荳 4-bit general-purpose area in bank 0 which is also used as the stack area
- 224荳 4-bit general-purpose area in bank 1
- 32荳 4-bit area for LCD data in bank 1

I/O MAP FOR HARDWARE REGISTERS

Table 3 contains detailed information about I/O mapping for peripheral hardware in bank 15 (register locations F80H-FFFH).



NOTES: 1. 'X' means don't care.
2. Blank columns indicate RAM areas that are not addressable, given the addressing method and enable memory bank (EMB) flag setting shown in the column headers.

Figure 19. RAM Address Structure

Table 3. I/O Map for Memory Bank 15

Memory Bank 15							Addressing Mode		
Address	**Register**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**R/W**	**1-Bit**	**4-Bit**	**8-Bit**
F80H	SP	.3	.2	.1	.0	R/W	No	No	Yes
F81H		.7	.6	.5	.4				
•									
•									
•									
F85H	BMOD	.3	.2	.1	.0	W	.3	Yes	No
F86H	BCNT	.3	.2	.1	.0	R	No	No	Yes
F87H		.7	.6	.5	.4				
F88H	WMOD	.3	.2	.1	.0	W	.3 (1)	No	Yes
F89H		.7	–	.5	.4				
F8AH	LPOT	.3	.2	.1	.0	R/W	.3	Yes	No
F8BH									
F8CH	LMOD	.3	.2	.1	.0	W	.3	No	Yes
F8DH		.7	–	.5	.4				
F8EH	LCON	–	–	.1	.0	W	Yes	Yes	No
F8FH	SUBSTP	.3	–	–	–	W	.3	Yes	No
F90H	TMOD0	.3	.2	–	–	W	.3	No	Yes
F91H		–	.6	.5	.4				
F92H	TOE	–	TOE0	BOE	–	R/W	Yes	Yes	No
F93H									
F94H	TCNT0	.3	.2	.1	.0	R	No	No	Yes
F95H		.7	.6	.5	.4				
F96H	TREF0	.3	.2	.1	.0	W	No	No	Yes
F97H		.7	.6	.5	.4				
F98H	IFCNT0	.3	.2	.1	.0	R	No	No	Yes
F99H		.7	.6	.5	.4				
F9AH	IFCNT1	.3	.2	.1	.0	R	No	No	Yes
F9BH		.7	.6	.5	.4				
F9CH	IFMOD	.3	.2	.1	.0	R/W	Yes	Yes	No
F9DH	PLLREG	ULFG	CEFG	IFCFG	–	R	Yes	Yes	No
•									
•									
•									

Table 3. I/O Map for Memory Bank 15 (Continued)

Memory Bank 15							Addressing Mode		
Address	**Register**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**R/W**	**1-Bit**	**4-Bit**	**8-Bit**
FB0H	PSW	IS1	IS0	EMB	ERB	R/W	Yes	Yes	Yes
FB1H		C (2)	SC2	SC1	SC0	R	No	No	
FB2H	IPR	IME	.2	.1	.0	W	IME	Yes	No
FB3H	PCON	.3	.2	.1	.0	W	No	Yes	No
FB4H	IMOD0	.3	–	.1	–	W	No	Yes	No
FB5H	IMOD1	–	–	–	.0	W	No	Yes	No
FB6H	IMOD2	–	–	.1	.0	W	No	Yes	No
FB7H	SCMOD	.3	–	–	.1	W	Yes	No	No
FB8H		IE4	IRQ4	IEB	IRQB	R/W	Yes	Yes	No
FB9H									
FBAH		–	–	IEW	IRQW	R/W	Yes	Yes	No
FBBH		IEIF	IRQIF	IECE	IRQCE				
FBCH		–	–	IET0	IRQT0				
FBDH		–	–	IES	IRQS				
FBEH		IE1	IRQ1	IE0	IRQ0				
FBFH		–	–	IE2	IRQ2				
FC0H	BSC0	.3	.2	.1	.0	R/W	Yes	Yes	Yes
FC1H	BSC1	.7	.6	.5	.4				
FC2H	BSC2	.11	.10	.9	.8				Yes
FC3H	BSC3	.15	.14	.13	.12				
FC4H	PLLD0	.3	.2	.1	.0	W	No	Yes	Yes
FC5H	PLLD1	.7	.6	.5	.4				
FC6H	PLLD2	.11	.10	.9	.8	W	No	Yes	Yes
FC7H	PLLD3	.15	.14	.13	.12				
FC8H	PLMOD	.3	.2	.1	.0	W	No	Yes	No
FC9H	PLLREF	.3	.2	.1	.0				
FCAH–FCFH									

Table 3. I/O Map for Memory Bank 15 (Continued)

Memory Bank 15							Addressing Mode		
Address	**Register**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**R/W**	**1-Bit**	**4-Bit**	**8-Bit**
FD0H	CLMOD	.3	.2	.1	.0	W	No	Yes	No
FD1H									
FD2H–FD7H									
FD8H	ADATA	.3	.2	.1	.0	R	No	No	Yes
FD9H		.7	.6	.5	.4				
FDAH	ADMOD	–	.2	.1	.0	R/W	Yes	Yes	No
FDBH	AFLAG	ADSTR	EOC	–	–	(3)	Yes	Yes	No
FDCH	PUMOD	PUR3	PUR2	PUR1	PUR0	R/W	No	No	Yes
FDDH		–	PUR6	PUR5	PUR4				
FDEH–FDFH									
FE0H	SMOD	.3	.2	.1	.0	W	.3	No	Yes
FE1H		.7	.6	.5	–				
FE2H									
FE3H									
FE4H	SBUF	.3	.2	.1	.0	R/W	No	No	Yes
FE5H		.7	.6	.5	.4				
FE6H	PMG0	PM0.3	PM0.2	PM0.1	PM0.0	R/W	No	No	Yes
FE7H		–	–	–	–				
FE8H	PMG1	PM2.3	PM2.2	PM2.1	PM2.0				
FE9H		PM3.3	PM3.2	PM3.1	PM3.0				
FEAH	PMG2	PM4.3	PM4.2	PM4.1	PM4.0				
FEBH		PM5.3	PM5.2	PM5.1	PM5.0				
FECH	PMG3	PM6.3	PM6.2	PM6.1	PM6.0				
FEDH		–	–	–	–				
FEEH–FEFH									

Table 3. I/O Map for Memory Bank 15 (Concluded)

Memory Bank 15							Addressing Mode		
Address	**Register**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**R/W**	**1-Bit**	**4-Bit**	**8-Bit**
FF0H	Port 0	.3	.2	.1	.0	R/W	Yes	Yes	No
FF1H	Port 1	.3	.2	.1	.0	R			
FF2H	Port 2	.3	.2	.1	.0	R/W			
FF3H	Port 3	.3	.2	.1	.0	R/W			
FF4H	Port 4	.3	.2	.1	.0	R/W			
FF5H	Port 5	.3	.2	.1	.0	R/W			
FF6H	Port 6	.3	.2	.1	.0	R/W	Yes	Yes	No
FF7H	Port 7	.3	.2	.1	.0	W			
FF8H	Port 8	.3	.2	.1	.0	W			
FF9H	Port 9	.3	.2	.1	.0	W			
FFAH	Port 10	–	.2	.1	.0	W			
FFBH	Port 11	.3	.2	.1	.0	W			
FFCH	Port 12	.3	.2	.1	.0	W			
FFDH	Port 13	.3	.2	.1	.0	W			
FFEH	POD	POD10	POD9	POD8	POD7	W	No	No	Yes
FFFH		–	POD13	POD12	POD11	W			

NOTES:
1. Bit 3 in the WMOD register is read only.
2. The carry flag can be read or written by specific bit manipulation instructions only.
3. The ADSTR bit of the AFLAG register is 1- or 4-bit write-only; the EOC bit is 1- or 4-bit read-only.

OSCILLATOR CIRCUITS

The KS57C3108 microcontroller has two oscillator circuits: a main system clock circuit, and a subsystem clock circuit. The main system clock frequencies can be divided by 4, 8, or 64 by manipulating PCON bits 1 and 0. The system clock mode control register, SCMOD, lets you select the main system clock (fx) or a subsystem clock (fxt) as the CPU clock and to start (or stop) main system clock oscillation.

The watch timer, buzzer and LCD display operate normally with a subsystem clock source because they operate at very slow speeds and with very low power consumption (as low as 122 µs at 32.768 kHz).

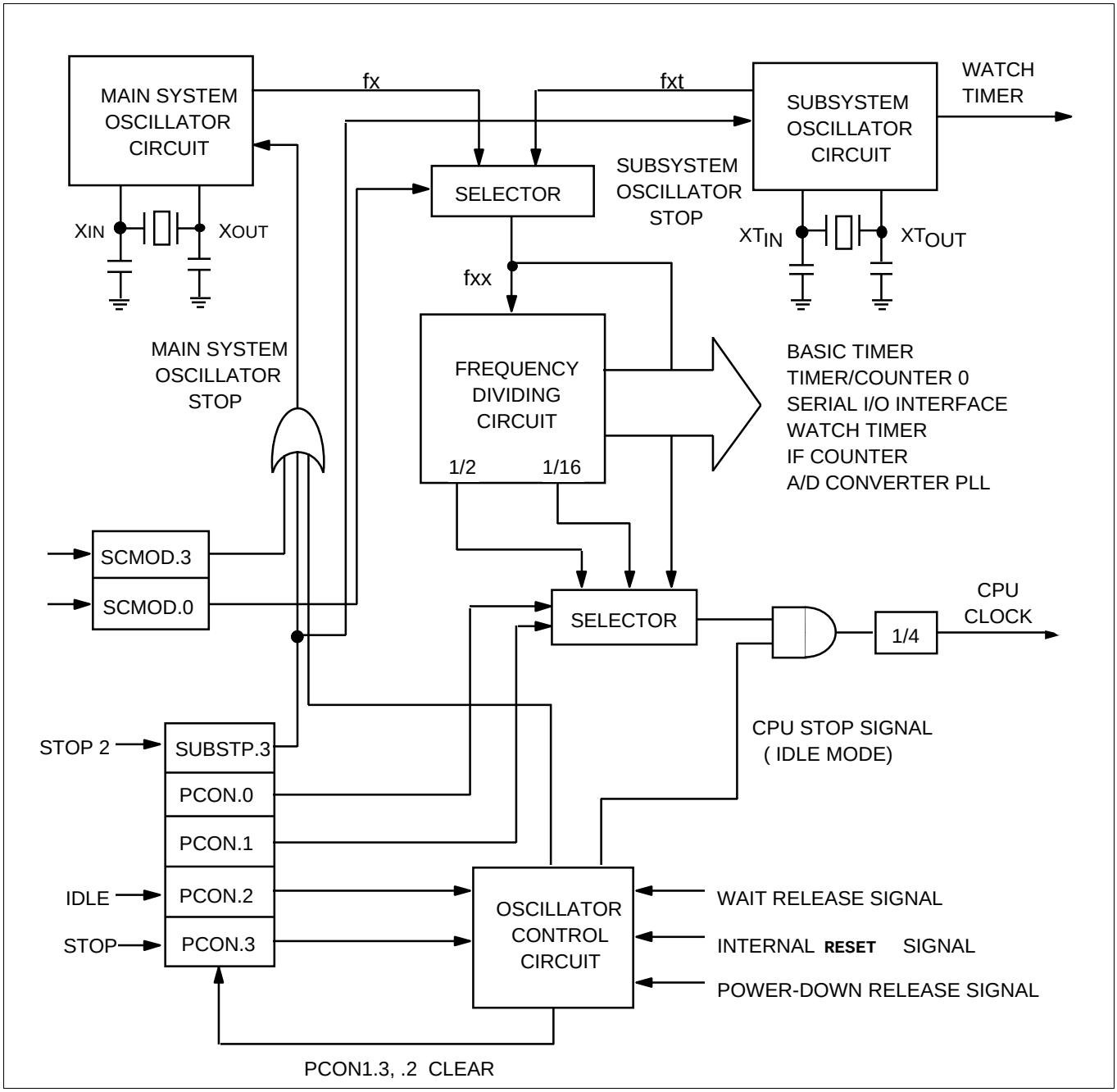


Figure 20. Clock Circuit Diagram

MAIN SYSTEM OSCILLATOR CIRCUITS

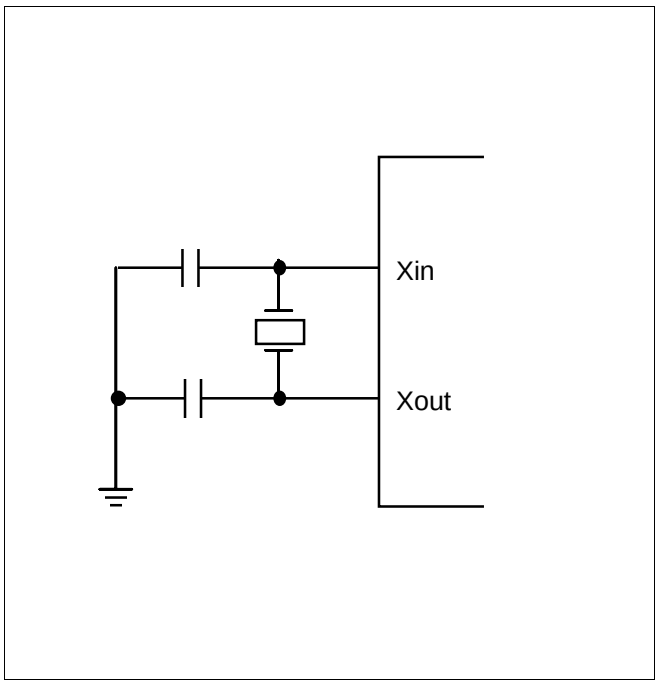


Figure 21. Crystal/Ceramic Oscillator (fx)

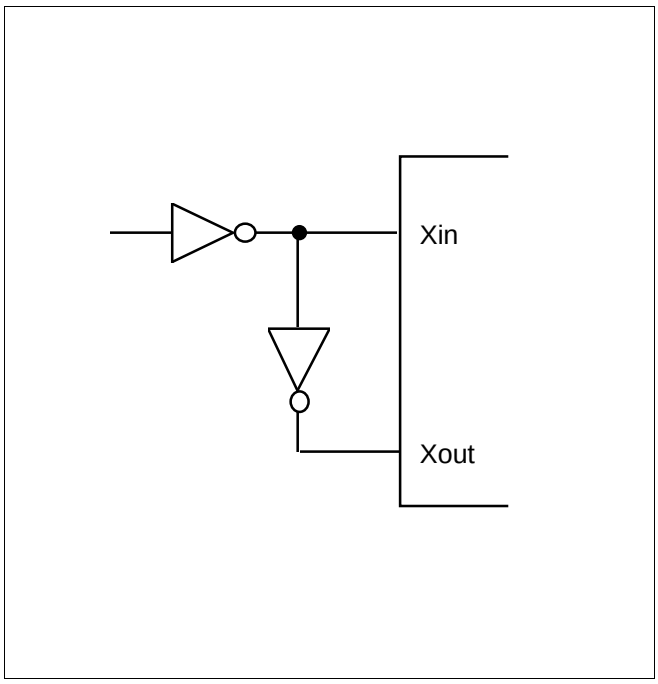


Figure 22. External Oscillator (fx)

SUBSYSTEM OSCILLATOR CIRCUITS

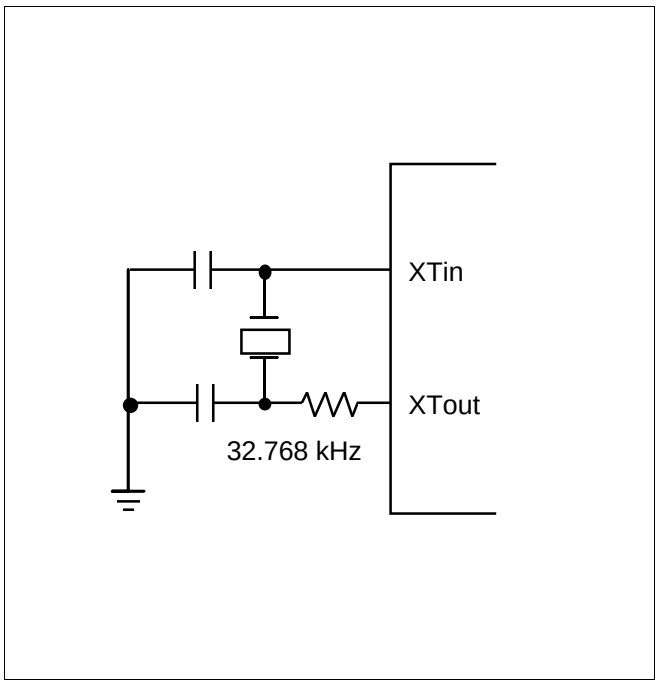


Figure 23. Crystal/Ceramic Oscillator (fxt)

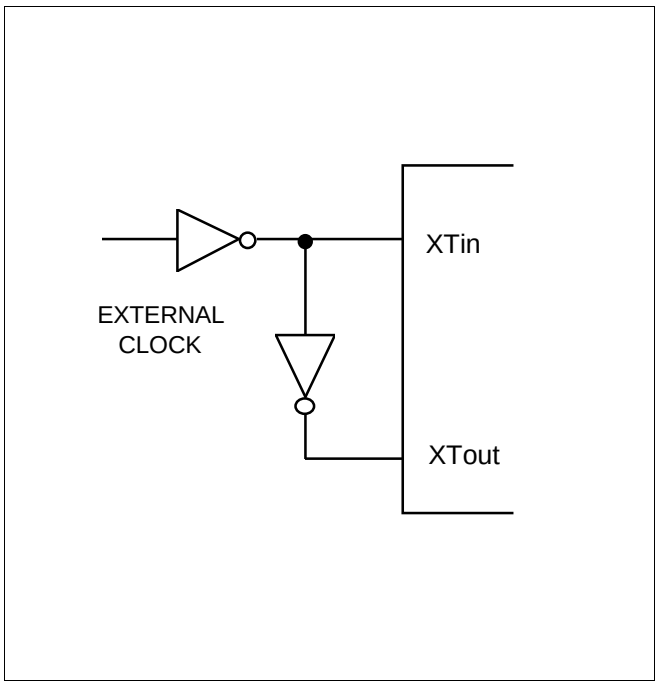


Figure 24. External Oscillator (fxt)

POWER CONTROL REGISTER (PCON) AND SUBCLOCK STOP CONTROL REGISTER (SUBSTP)

The power control register (PCON) and subclock stop control register (SUBSTP) are used to select the CPU clock frequency and to control CPU operating and power-down modes. They can be addressed directly by 4-bit write instructions or indirectly by the instructions IDLE and STOP.

PCON.3 and PCON.2 can also be addressed by the STOP and IDLE instructions, respectively, to engage the Idle and Stop power-down modes. Idle and Stop1 modes can be initiated by these instruction despite the current value of the enable memory bank flag (EMB). Stop2 mode is entered by setting bit SUBSTP.3 in the SUBSTP register. PCON bits 1 and 0 are used to select a specific system clock frequency. There are two basic choices:

— Main system clock (fx) or subsystem clock (fxt);

— Divided fx clock frequency of 4, 8, or 64.

— Divided fxt clock frequency of 4.

PCON.1 and PCON.0 settings are also connected with the system clock mode control register, SCMOD. If SCMOD.0 = "0", the main system clock is always selected by the PCON.1 and PCON.0 setting; if SCMOD.0 = "1" the subsystem clock is selected. A reset sets PCON and SUBSTP register values (and SCMOD) to logic zero.

Table 4. PCON and SUBSTP Register Organization

PCON and SUBSTP Bit Settings			Resulting CPU Operating Mode
SUBSTP.3	PCON.3	PCON.2	
0	0	0	Normal CPU operating mode
0	0	1	Idle power-down mode
0	1	0	Stop1 power-down mode (only main system clock stops)
1	X	X	Stop2 power-down mode (main and subsystem clock stop)

PCON Bit Settings		Resulting CPU Clock Frequency	
PCON .1	PCON.0	If SCMOD.0 = "0"	If SCMOD.0 = "1"
0	0	fx/64	—
1	0	fx/8	—
1	1	fx/4	fxt/4

☞ PROGRAMMING TIP — Setting the CPU Clock

To set the CPU clock to 0.889 µs at 4.5 MHz:

```
BITS    EMB
SMB     15
LD      A,#3H
LD      PCON,A
```

INSTRUCTION CYCLE TIMES

The unit of time that equals one machine cycle varies depending on whether the main system clock (fx) or a subsystem clock (fxt) is used, and on how the oscillator clock signal is divided (by 4, 8, or 64). Table 16 shows corresponding cycle times in microseconds.

Table 5. Instruction Cycle Times for CPU Clock Rates

Selected CPU Clock	Resulting Frequency	Oscillation Source	Cycle Time (μsec)
fx/64	65.5 kHz	fx = 4.5 MHz	14.2
fx/8	524.0 kHz		1.77
fx/4	1.05 MHz		0.889
fxt/4	8.19 kHz	fxt = 32.768 kHz	122.0

SYSTEM CLOCK MODE REGISTER (SCMOD)

The system clock mode register, SCMOD, is a 4-bit register that is used to select the CPU clock and to control main system clock oscillation. RESET clears all SCMOD values to logic zero, selecting the main system clock (fx) as the CPU clock and starting clock oscillation. The least significant and most significant SCMOD bits can be manipulated using 1-bit write instructions. (In other words, SCMOD.0 and SCMOD.3 cannot be modified simultaneously using a 4-bit write operation.) Bits 2 and 1 are always logic zero.

A subsystem clock (fxt) can be selected as the system clock by manipulating the SCMOD.3 and SCMOD.0 bit settings. If SCMOD.3 = "0" and SCMOD.0 = "1", the subsystem clock is selected and main system clock oscillation continues. If SCMOD.3 = "1" and SCMOD.0 = "0", fxt is selected, but the main system clock stops.

Subsystem clock oscillation can be stopped only by setting bit SUBSTP.3. If you have selected fx as the CPU clock, setting SCMOD.3 to "1" will not stop the main system clock. This can be done only by a STOP instruction.

Table 6. System Clock Mode Register (SCMOD) Organization

SCMOD Register Bit Settings		Resulting Clock Selection	
SCMOD.3	SCMOD.0	CPU Clock	fx Oscillation
0	0	fx	On
0	1	fxt	On
1	1	fxt	Off

SWITCHING THE CPU CLOCK

Together, bit settings in the power control register, PCON1, and the system clock mode register, SCMOD, determine whether a main system or a subsystem clock is selected as the CPU clock, and also how this frequency is to be divided. This makes it possible to switch dynamically between main and subsystem clocks and to modify operating frequencies.

SCMOD.3 and SCMOD.0 select the main system clock (fx) or a subsystem clock (fxt) and start or stop main system clock oscillation. PCON1.1 and PCON1.0 control the frequency divider circuit, and divide the selected fx or fxt clock by 4, 8, or 64.

NOTE

A clock switch operation does not go into effect immediately when you make the SCMOD and PCON register modifications — the previously selected clock continues to run for a certain number of machine cycles.

For example, you are using the default CPU clock (normal operating mode and a main system clock of fx/64) and you want to switch from the fx clock to a subsystem clock and to stop the main system clock. To do this, you first need to set SCMOD.0 to "1". This switches the clock from fx to fxt but allows main system clock oscillation to continue. Before the switch actually goes into effect, a certain number of machine cycles must elapse. After this time interval has elapsed, you can disable main system clock oscillation by setting SCMOD.3 to "1".

This same 'stepped' approach must be taken to switch from a subsystem clock to the main system clock: first, clear SCMOD.3 to "0" to enable main system clock oscillation. Then, after a certain number of machine cycles has elapsed, select the main system clock by clearing all SCMOD values to logic zero.

Following a RESET, CPU operation starts with the lowest main system clock frequency of 14.2 µsec at 4.5 MHz after the standard oscillation stabilization interval of 29.1 ms has elapsed. Table 18 details the number of machine cycles that must elapse before a CPU clock switch modification goes into effect.

Table 7. Elapsed Machine Cycles During CPU Clock Switch

BEFORE	AFTER	SCMOD.0 = 0						SCMOD.0 = 1
		PCON.1 = 0	PCON.0 = 0	PCON.1 = 1	PCON.0 = 0	PCON.1 = 1	PCON.0 = 1	
SCMOD.0 = 0	PCON.1 = 0	N/A		1 MACHINE CYCLE		1 MACHINE CYCLE		N/A
	PCON.0 = 0							
	PCON.1 = 1	8 MACHINE CYCLES		N/A		8 MACHINE CYCLES		N/A
	PCON.0 = 0							
	PCON.1 = 1	16 MACHINE CYCLES		16 MACHINE CYCLES		N/A		fx / 4fxt
	PCON.0 = 1							
SCMOD.0 = 1		N/A		N/A		fx / 4fxt (M/C)		N/A

NOTES:
1. Even if oscillation is stopped by setting SCMOD.3 during main system clock operation, Stop mode is not entered.
2. Because the X_{IN} input is connected internally to V_{SS} to avoid current leakage due to the crystal oscillator in Stop mode, do not set SCMOD.3 to "1" when an external clock is used as the main system clock.
3. When the system clock is switched to the subsystem clock, you must disable any interrupts that may occur during the time intervals shown in Table 7.
4. 'N/A' means 'not available'.

☞ PROGRAMMING TIP — Switching Between Main System and Subsystem Clock

1. Switch from the main system clock to the subsystem clock:

```
MA2SUB   BITS   SCMOD.0    ;  Switches to subsystem clock
         CALL   DLY80      ;  Delay 80 machine cycles
         BITS   SCMOD.3    ;  Stop the main system clock
         RET
DLY80    LD     A,#0FH
DEL1     NOP
         NOP
         DECS   A
         JR     DEL1
         RET
```

2. Switch from the subsystem clock to the main system clock:

```
SUB2MA   BITR   SCMOD.3    ;  Start main system clock oscillation
         CALL   DLY80      ;  Delay 80 machine cycles
         BITR   SCMOD.0    ;  Switch to main system clock
         RET
```

CLOCK OUTPUT MODE REGISTER (CLMOD)

The clock output mode register, CLMOD, is a 4-bit register that is used to enable or disable clock output to the CLO pin and to select the CPU clock source and frequency. CLMOD is addressable by 4-bit write instructions only.

RESET clears CLMOD to logic zero, which automatically selects the CPU clock as the clock source (without initiating clock oscillation), and disables clock output.

CLMOD.3 is the enable/disable clock output control bit; CLMOD.1 and CLMOD.0 are used to select one of four possible clock sources and frequencies: normal CPU clock, fxx/8, fxx/16, or fxx/64.

Table 8. Clock Output Mode Register (CLMOD) Organization

CLMOD Bit Settings		Resulting Clock Output	
CLMOD.1	**CLMOD.0**	**Clock Source**	**Frequency**
0	0	CPU clock (fx/4, fx/8, fx/64, fxt/4)	1.125 MHz, 562.5 kHz, 70.312kHz, 8.19 kHz
0	1	fxx/8	562.5 kHz, 4.096 kHz
1	0	fxx/16	281.25 kHz, 2.048 kHz
1	1	fxx/64	70.312 kHz, 0.512 kHz

CLMOD.3	Result of CLMOD.3 Setting
0	Clock output is disabled
1	Clock output is enabled

NOTE: Frequencies assume that fx is 4.5 MHz and fxt is 32.768 kHz.

CLOCK OUTPUT CIRCUIT

The clock output circuit, used to output clock pulses to the CLO pin, has the following components:

— 4-bit clock output mode register (CLMOD)
— Clock selector
— Output latch
— Port mode flag
— CLO output pin (P4.3)

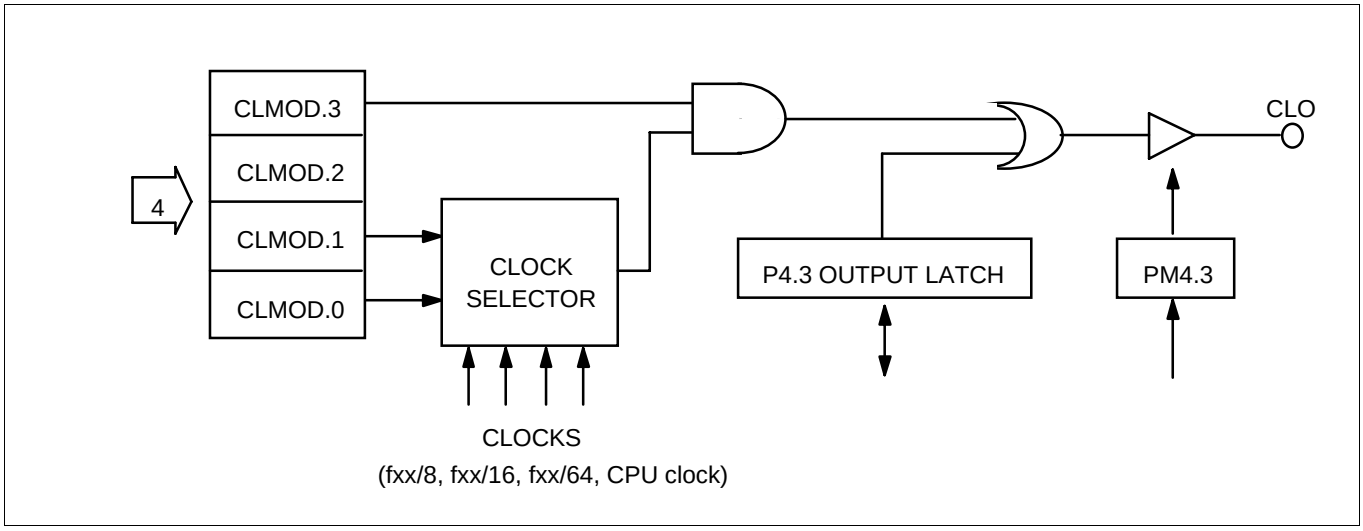


Figure 25. CLO Output Pin Circuit Diagram

CLOCK OUTPUT PROCEDURE

The procedure for outputting clock pulses to the CLO pin may be summarized as follows:

1. Disable clock output by clearing CLMOD.3 to logic zero.
2. Set the clock output frequency (CLMOD.1, CLMOD.0).
3. Load a "0" to the output latch of the CLO pin (P4.3).
4. Set the P4.3 mode flag (PM4.3) to output mode.
5. Enable clock output by setting CLMOD.3 to logic one.

PROGRAMMING TIP — CPU Clock Output to the CLO Pin

To output the CPU clock to the CLO pin:

```
BITS    EMB
SMB     15
LD      EA,#40H
LD      PMG2,EA          ;  P4.3 ← Output mode
BITR    P4.3             ;  Clear P4.3 output latch
LD      A,#9H
LD      CLMOD,A
```

INTERRUPTS

The KS57C3016 has four external interrupts, four internal interrupts and two quasi-interrupts. The request flags that allow these interrupts to be generated are cleared by hardware when the service routine is vectored. The quasi-interrupt's request flags must be cleared by software.

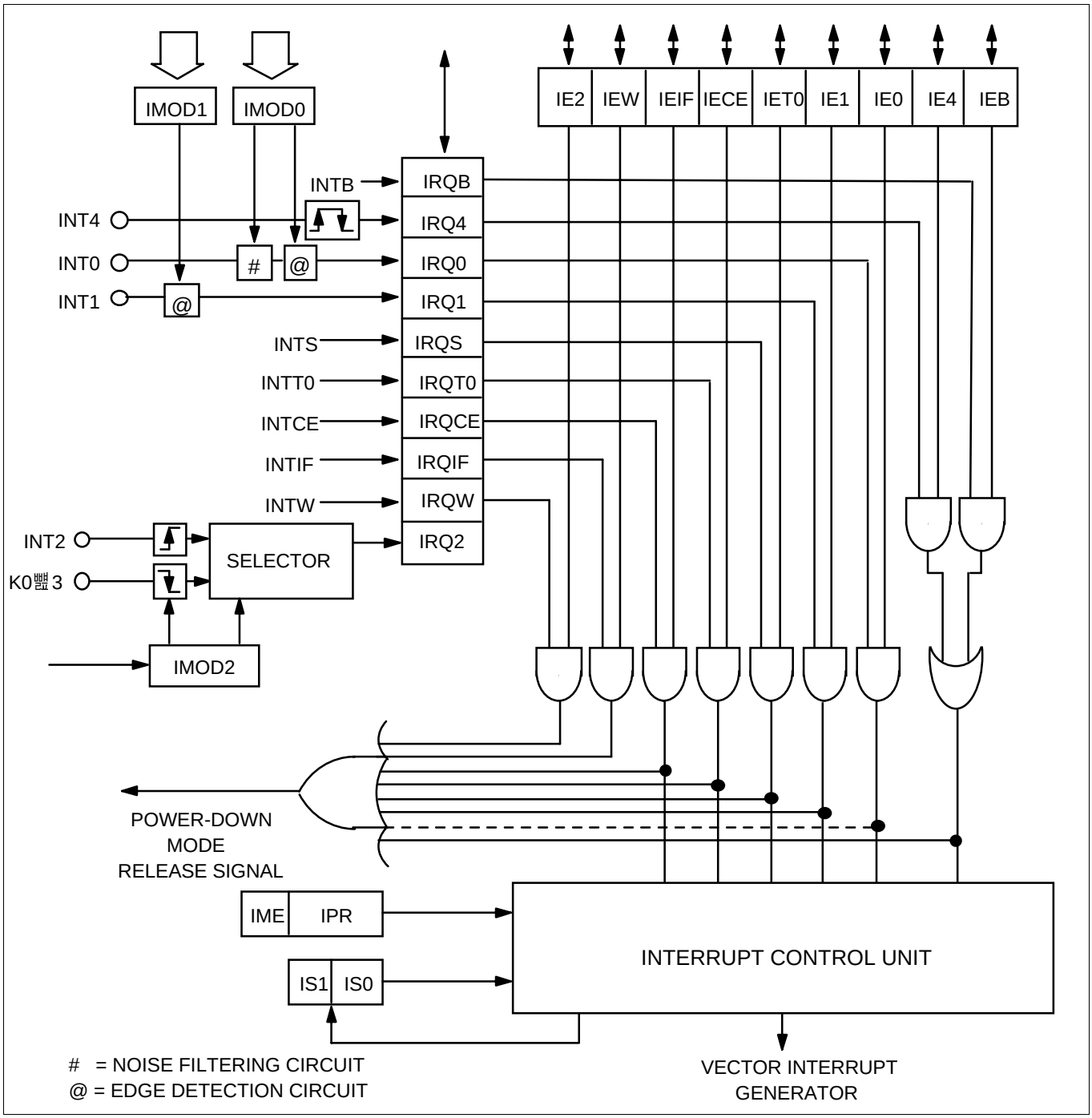


Figure 27. Interrupt Control Circuit Diagram

INTERRUPT PRIORITY REGISTER (IPR)

The 4-bit interrupt priority register (IPR) is used to control multi-level interrupt handling and its reset value is logic zero. Before the IPR can be modified, all interrupts must first be disabled by a DI instruction.

By manipulating the IPR settings, you can choose to process all interrupt requests with the same priority level, or you can select one type of interrupt for high-priority processing. A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

Table 9. Standard Interrupt Priorities

Interrupt	Default Priority
INTB, INT4	1
INT0	2
INT1	3
INTS	4
INTT0	5
INTCE	6
INTIF	7

The MSB of the IPR, the interrupt master enable flag (IME), enables and disables all interrupt processing. Even if an interrupt request flag and its corresponding enable flag are set, a service routine cannot be executed until the IME flag is set to logic one. The IME flag can be directly manipulated by EI and DI instructions, regardless of the current enable memory bank (EMB) value.

Table 10. Interrupt Priority Register Settings

IPR.2	IPR.1	IPR.0	Result of IPR Bit Setting
0	0	0	Process all interrupt requests at low priority (see Note)
0	0	1	Process INTB and INT4 interrupts only
0	1	0	Process INT0 interrupts only
0	1	1	Process INT1 interrupts only
1	0	0	Process INTS interrupts only
1	0	1	Process INTT0 interrupts only
1	1	0	Process INTCE interrupts only
1	1	1	Process INTIF interrupts only

☞ PROGRAMMING TIP — Setting the INT Interrupt Priority

The following instruction sequence sets the INT1 interrupt to high priority:

```
BITS    EMB
SMB     15
DI                      ;  IPR.3 (IME) ← 0
LD      A,#3H
LD      IPR,A
EI                      ;  IPR.3 (IME) ← 1
```

EXTERNAL INTERRUPT 0 and 1 MODE REGISTERS (IMOD0, IMOD1)

The following components are used to process external interrupts at the INT0 and INT1 pin:

— Noise filtering circuit for INT0

— Edge detection circuit

— Two mode registers, IMOD0 and IMOD1

The mode registers are used to control the triggering edge of the input signal. IMOD0 and IMOD1 settings let you choose either the rising or falling edge of the incoming signal as the interrupt request trigger. The INT4 interrupt is an exception since its input signal generates an interrupt request on both rising and falling edges.

IMOD0 and IMOD1 are addressable by 4-bit write instructions. RESET clears all IMOD values to logic zero, selecting rising edges as the trigger for incoming interrupt requests.

Table 11. IMOD0 and IMOD1 Register Organization

IMOD0	**IMOD0.3**	**0**	**IMOD0.1**	**IMOD0.0**	**Effect of IMOD0 Settings**
	0				Select CPU clock for sampling
	1				Select fxx/64 sampling clock
			0	0	Rising edge detection
			0	1	Falling edge detection
			1	0	Both rising and falling edge detection
			1	1	IRQ0 flag cannot be set to "1"

IMOD1	**0**	**0**	**0**	**IMOD1.0**	**Effect of IMOD1 Settings**
				0	Rising edge detection
				1	Falling edge detection

When a sampling clock rate of fxx/64 is used for INT0, an interrupt request flag must be cleared before 16 machine cycles have elapsed. Because the INT0 pin has a clock-driven noise filtering circuit built into it, please take the following precautions when you use it:

— To trigger an interrupt, the input signal width at INT0 must be at least two times wider than the pulse width of the clock selected by IMOD0. This is true even when the INT0 pin is used for general-purpose input.

— Because the INT0 input sampling clock does not operate during Stop or Idle mode, you cannot use INT0 to release power-down mode.

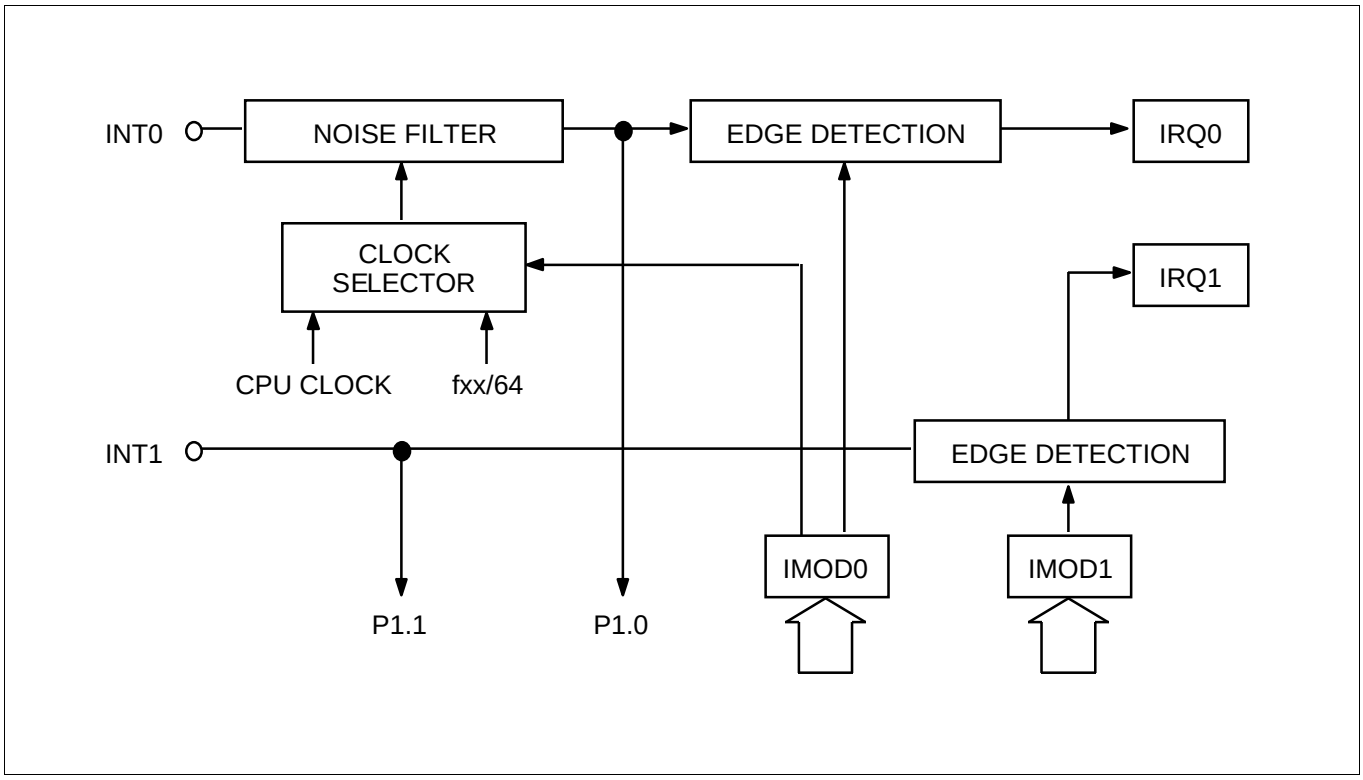


Figure 28. Circuit Diagram for INT0 and INT1 Pins

When modifying the IMOD0 and IMOD1 registers, it is possible to accidentally set an interrupt request flag. To avoid unwanted interrupts, take these precautions when writing your programs:

1. Disable all interrupts with a DI instruction.
2. Modify the IMOD0 or IMOD1 register.
3. Clear all relevant interrupt request flags.
4. Enable the interrupt by setting the appropriate IEx flag.
5. Enable all interrupts with an EI instructions.

EXTERNAL INTERRUPT 2 MODE REGISTER (IMOD2)

The mode register for external interrupts at the INT2 pin, IMOD2, is addressable only by 4-bit write instructions. RESET clears all IMOD2 bits to logic zero.

When IMOD2 is cleared to logic zero, INT2 uses the rising edge of an incoming signal as the interrupt request trigger. If a rising edge is detected at the INT2 pin, or when a falling edge is detected at any one of the pins KS0–KS3, the IRQ2 flag is set to logic one and a release signal for power-down mode is generated.

Table 12. IMOD2 Register Bit Settings

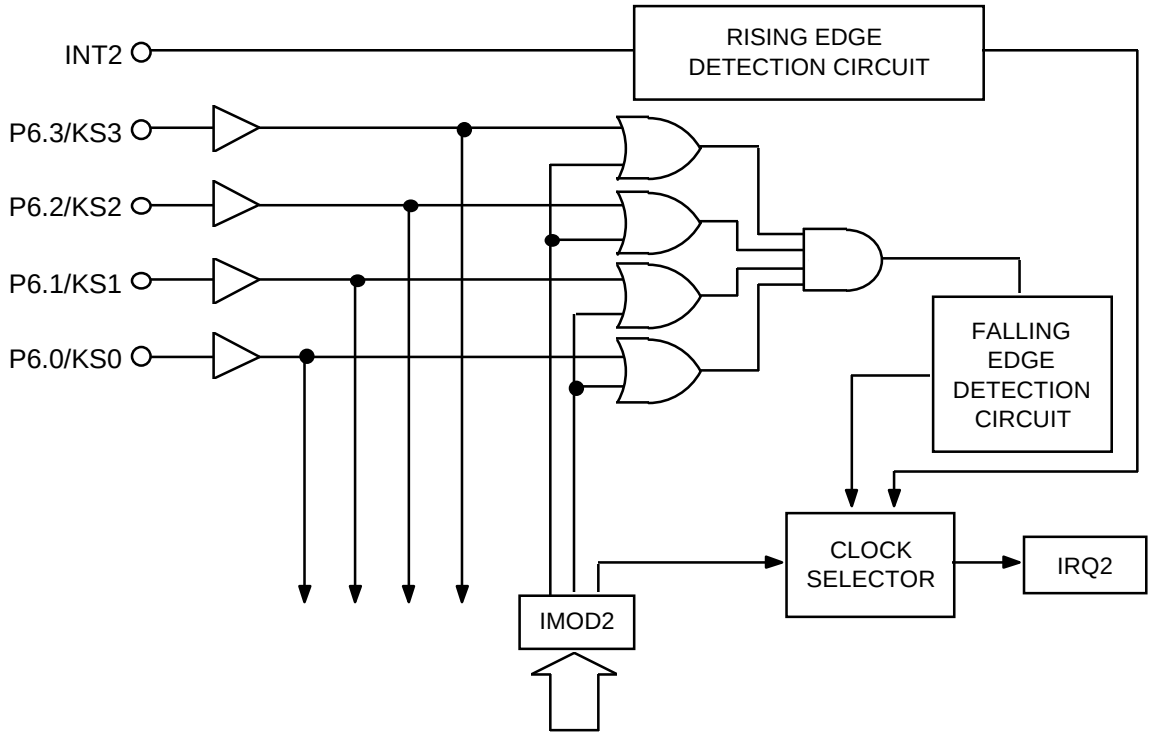
IMOD2	0	0	IMOD2.1	IMOD2.0	Effect of IMOD2 Settings
			0	0	Select rising edge at INT2 pin
			1	0	Select falling edge at KS2–KS3
			1	1	Select falling edge at KS0–KS3

☞ Programming Tip — Using INT2 as a Key Input Interrupt

When you use the INT2 interrupt as a key entry interrupt, the selected key interrupt source pin must be set to input mode:

1. When KS0–KS3 are selected (four pins):

```
BITS    EMB
SMB     15
LD      A,#3H
LD      IMOD2,A        ;  (IMOD2) ← #3H, KS0–KS3 falling edge select
LD      EA,#00H
LD      PMG3,EA        ;  P6 ← input mode
LD      EA,#40H
LD      PUMOD,EA       ;  Enable P6 pull-up resistors
```



NOTE: To generate a key interrupt on a falling edge at KS0–S3, all KS0–S3 pins must be configured to input mode.

Figure 29. Circuit Diagram for INT2

INTERRUPT FLAGS

There are three types of interrupt flags: interrupt request and interrupt enable flags that correspond to each interrupt, the interrupt master enable flag, which enables or disables all interrupt processing.

Interrupt Master Enable Flag (IME)

The interrupt master enable flag, IME, enables or disables all interrupt processing. Therefore, even when an IRQx flag is set and its corresponding IEx flag is enabled, the interrupt service routine is not executed until the IME flag is set to logic one.

The IME flag is located in the IPR register (IPR.3). It can be directly be manipulated by EI and DI instructions, regardless of the current value of the enable memory bank flag (EMB).

IME	IPR.2	IPR.1	IPR.0	Effect of Bit Settings
0				Inhibit all interrupts
1				Enable all interrupts

POWER-DOWN

The KS57C3108 microcontroller has four power-down modes to reduce power consumption: Idle, Stop1, Stop2, and CE Low modes.

Idle mode is initiated by the IDLE instruction and Stop1 mode by the instruction STOP (Several NOP instructions must always follow an IDLE or STOP instruction in a program). In Idle mode, the CPU clock stops while peripherals and the oscillation source continue to operate normally.

In Stop1 mode, main system clock oscillation is halted (assuming it is currently operating), and all peripheral hardware components are powered-down.

You initiate Stop2 mode by setting bit 3 of the SUBSTP register to "1". In Stop2 mode, both the main clock and the subsystem clock are stopped. In CE Low mode, only the PLL module is disabled; all other peripherals operate normally.

The effects of power-down modes on specific peripheral components are detailed in Table 16.

NOTE

Do not use Stop mode if you are using an external clock source because X_{IN} input must be restricted internally to V_{SS} in order to reduce current leakage.

Idle or Stop1 modes are released either by a RESET, or by an interrupt which is enabled by the corresponding interrupt enable flag, IEx (An exception is INT0). Stop2 mode can only be released by RESET. When a power-down mode is released by RESET, a normal reset operation is executed.

Assuming that both the interrupt enable flag and the interrupt request flag are set to "1", a power-down mode is released immediately after entering the power-down mode.

Table 13. Hardware Operation During Power-Down Modes

Operation	Stop1 Mode	Stop2 Mode	Idle Mode	CE Low Mode
Instruction	STOP	BITS SUBSTP.3	IDLE	CE pin low input
System clock status	Can be changed only if the main system clock is used	Can not be changed	Can be changed if the main system clock or subsystem clock is used	Can be changed only when clock is used
Clock oscillator	Main system clock oscillation stops	Both main system clock and subsystem clock oscillation stop	Only CPU clock oscillation stops (main and subsystem clock oscillation continues)	Clock oscillation is not stopped
Basic timer	Basic timer stops	Basic timer stops	Basic timer operates (with IRQB set at each reference interval)	Basic timer operates (with IRQB set at each reference interval)
Serial interface	Operates only if external SCK input is selected as the serial I/O clock	Operation stops	Operates if a clock other than the CPU clock is selected as the serial I/O clock	Serial I/O interface operates
Timer/counter 0	Operates only if TCL is selected as the counter clock	Operation stops	Timer/counter 0 operates	Timer/counter 0 operates
Watch timer	Operates only if subsystem clock (fxt) is selected as the counter clock	Operation stops	Watch timer operates	Watch timer operates
LCD controller	Operates only if a subsystem clock is selected as LCDCK	Operation stops	LCD controller operates	LCD controller operates
External interrupts	INT1, INT2, and INT4 are acknowledged; INT0 is not serviced	External interrupt are not acknowledged	INT1, INT2, and INT4 are acknowledged; INT0 is not serviced	All external interrupts are acknowledged
CPU	All CPU operations are disabled	All CPU operations are disabled	All CPU operations are disabled	CPU operates normally
PLL	PLL stops	PLL stops	PLL operates	PLL stops
A/D converter	A/D converter is disabled	A/D converter is disabled	A/D converter operates	A/D converter operates
Mode release signal	Interrupt request signals (except INT0) or RESET input	Only RESET input	Interrupt request signals (except INT0) or RESET input	CE pin high
IFC	IFC stops	IFC stops	IFC operates	IFC operates

NOTE: CE mode is not controlled by instruction. It can only be changed by external CE pin state.

STOP MODE TIMING DIAGRAMS

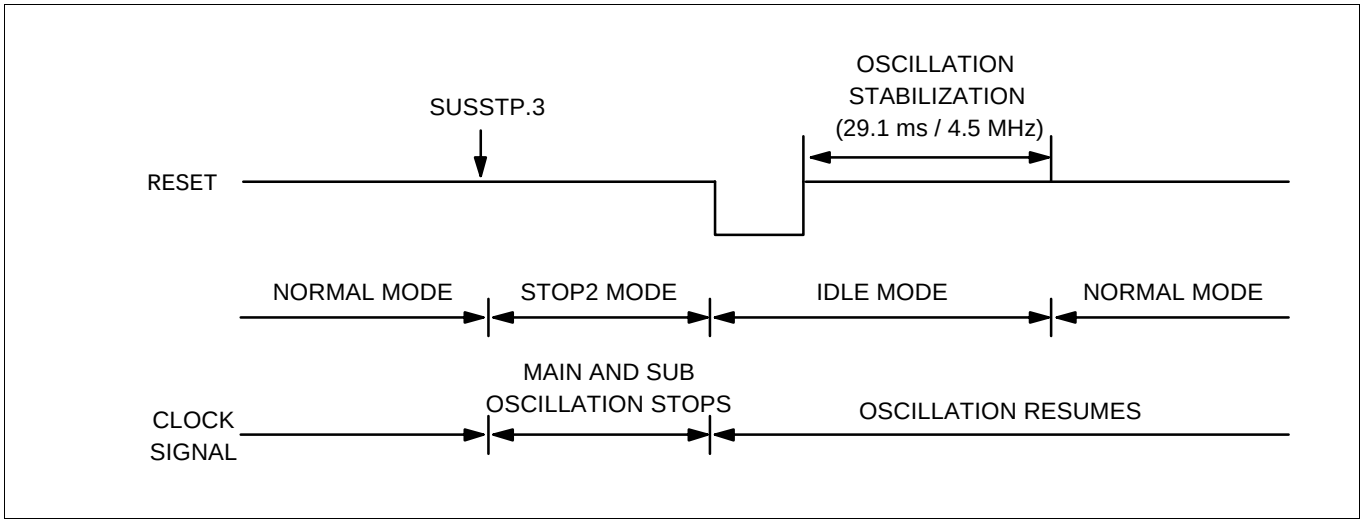


Figure 30. Timing When Stop2 Mode is Release by RESET

CE LOW MODE TIMING DIAGRAMS

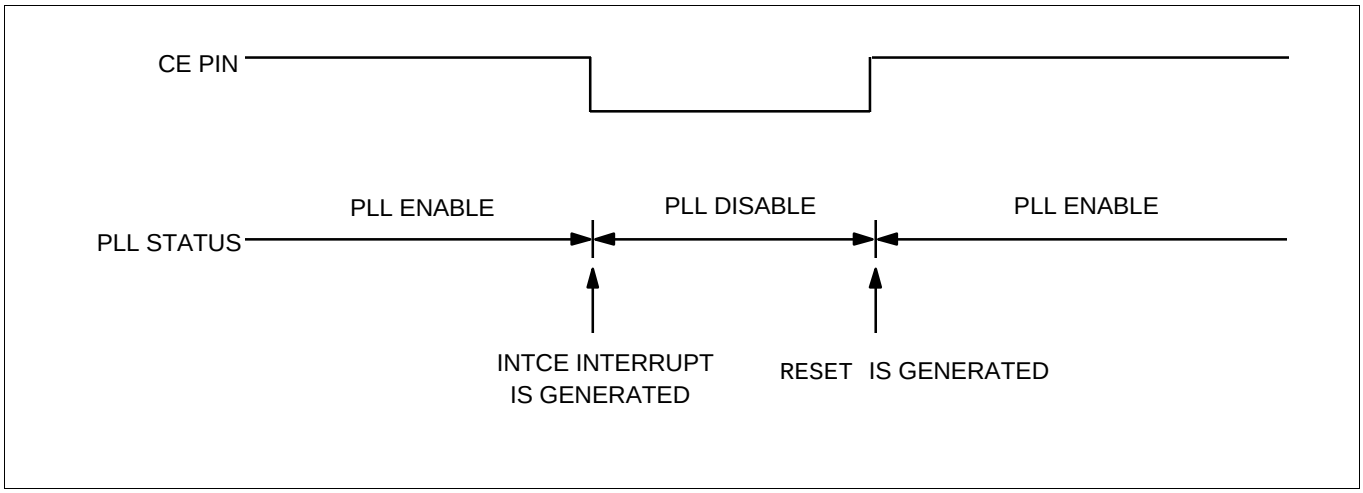


Figure 31. Timing When CE Low Mode is Released by RESET

☞ PROGRAMMING TIP — Reducing Power Consumption for Key Input Interrupt Processing

The following code shows real-time clock and interrupt processing for key inputs for reduced power consumption. In this example, the system clock source is switched from the main system clock to a subsystem clock and the LCD display is turned on:

```
KEYCLK  DI
        CALL    MA2SUB      ; Main system clock → subsystem clock switch
                            subroutine
        SMB     15
        LD      EA,#00H
        LD      P4,EA       ; All key strobe outputs go to Low level
        LD      A,#3H
        LD      IMOD2,A     ; Select KS0–KS3 enable
        SMB     0
        BITR    IRQW
        BITR    IRQ2
        BITS    IEW
        BITS    IE2
CLKS1   CALL    WATDIS      ; Execute clock and display changing subroutine
        BTSTZ   IRQ2
        JR      CIDLE
        CALL    SUB2MA      ; Subsystem clock → main system clock switch
                            subroutine
        EI
        RET
CIDLE   IDLE                ; Enter Idle mode
        NOP
        NOP
        JPS     CLKS1
```

RECOMMENDED CONNECTIONS FOR UNUSED PINS

To reduce overall power consumption, please configure unused pins according to the guidelines described in Table 14.

Table 14. Unused Pin Connections for Reduced Power Consumption

Pin/Share Pin Names	Recommended Connection
P0.0 / BTCO P0.1 / TCLO P0.2 / TCL P0.3 / BUZ	Input mode: Connect to V_{DD} Output mode: Do not connect
P1.0 / INT0 – P1.2 / INT2	Connect to V_{DD}
P1.3 / INT4	Connect to V_{SS}
P2.0 –P2.3 P3.0 –P3.3 P4.0 / SCK P4.1 / SO P4.2 / SI P4.3 / CLO P5.0 / ADC0 – P5.3 / ADC3 P6.0 / KS0 – P6.3 / KS3	Input mode: Connect to V_{DD} Output mode: Do not connect AD mode: Do not connect
SEG0–SEG27 COM0–COM3	Do not connect
V_{LC0}–V_{LC2}	Connect to V_{SS}
BIAS	If all of the V_{LC0}–V_{LC2} pins are unused, connect BIAS to V_{SS}
XT_{IN}	Connect XT_{IN} to V_{SS} or to V_{DD}
XT_{OUT}	Do not connect
AMIF, FMIF	Connect to V_{SS}
TEST	Connect to V_{SS}

RESET

System reset operation can be initiated by RESET or CE pin. When a reset operation occurs, the system is initialized and the program is executed from reset vector address. The CE reset occurs when the CE pin rises from Low to High. CE reset is also used to initialize a system. However, RESET signal is not generated automatically.

Table 15. Hardware Register Values After RESET

Hardware Component or Subcomponent	If RESET Occurs During Operating Mode	If RESET Occurs After Power-On
	RESET Pin	RESET Pin
Program counter (PC)	Lower six bits of address 0000H are transferred to PC12–8, and the contents of 0001H to PC7–0.	Lower six bits of address 0000H are transferred to PC12–8, and the contents of 0001H to PC7–0.
Bank selection registers (SMB, SRB)	0, 0	0, 0
BSC register (BSC0–BSC3)	0	0
Program Status Word (PSW):		
Carry flag (C)	Retained	Undefined
Skip flag (SC0–SC2)	0	0
Interrupt status flags (IS0, IS1)	0	0
Bank enable flags (EMB, ERB)	Bit 6 of address 0000H in program memory is transferred to the ERB flag, and bit 7 of the address to the EMB flag.	Bit 6 of address 0000H in program memory is transferred to the ERB flag, and bit 7 of the address to the EMB flag.
Stack pointer (SP)	Undefined	Undefined
Data Memory (RAM):		
Working registers E, A, L, H, X, W, Z, Y	Values retained	Undefined
General-purpose registers	Retained (note)	Undefined
Clocks:		
Power control register (PCON)	0	0
Subclock stop control register (SUBSTP)	0	0
Clock output mode register (CLMOD)	0	0
System clock mode register (SCMOD)	0	0

NOTE: The values of the 0F8H–0FDH are not retained when a RESET signal is input.

Table 15. Hardware Register Values After RESET (Continued)

Hardware Component or Subcomponent	If RESET Occurs During Operating Mode	If RESET Occurs After Power-On
	RESET Pin	RESET Pin
Interrupts:		
Interrupt request flags (IRQx)	0	0
Interrupt enable flags (IEx)	0	0
Interrupt priority flag (IPR)	0	0
Interrupt master enable flag (IME)	0	0
INT0 mode register (IMOD0)	0	0
INT1 mode register (IMOD1)	0	0
INT2 mode register (IMOD2)	0	0
I/O Ports:		
Output buffers	Off	Off
Output latches	0	0
Port mode flags (PM)	0	0
Pull-up resistor mode reg (PUMOD1/2)	0	0
Basic Timer:		
Count register (BCNT)	Undefined	Undefined
Mode register (BMOD)	0	0
Output enable flag (BOE)	0	0
Timer/Counter 0:		
Count registers (TCNT0)	0	0
Reference registers (TREF0)	FFH	FFH
Mode registers (TMOD0)	0	0
Output enable flags (TOE0)	0	0
Watch Timer:		
Watch timer mode register (WMOD)	0	0
LCD Driver/Controller:		
LCD mode register (LMOD)	0	0
LCD control register (LCON)	0	0
Display data memory	Values retained	Undefined
Output buffers	Off	Off

Table 15. Hardware Register Values After RESET (Concluded)

Hardware Component or Subcomponent	If RESET Occurs During Operating Mode	If RESET Occurs After Power-On
	RESET Pin	RESET Pin
Serial I/O Interface:		
SIO mode register (SMOD)	0	0
SIO interface buffer (SBUF)	Values retained	Undefined
PLL		
PLL mode register (PLMOD)	Values retained	Undefined
PLL data registers (PLLD0–PLLD3)	Values retained	Undefined
PLL flag register (PLLREG)	(see Note 1)	(see Note 2)
IF Counter		
IF counter mode register (IFMOD)	0	0
IF counter (IFCNT0, IFCNT1)	0	0
A/D Converter		
A/D converter mode register (ADMOD)	0	0
A/D converter data register (ADATA)	0	0

NOTES:
1. The value of ULFG is undefined, CEFG = current state of CE pin, and CFG = "0".
2. The value of ULFG is undefined, CEFG = current state of CE pin, and IFCFG is undefined.

I/O PORTS

The KS57C3108 has 14 ports. There are 4 input pins, 28 output pins, and 16 configurable I/O pins for a total number of 56 I/O pins.

Pin addresses for all ports are mapped in bank 15 of the RAM. The contents of I/O port pin latches can be read, written, or tested at the corresponding address using bit manipulation instructions.

Table 16. I/O Port Overview

Port	I/O	Pins	Pin Names	Address	Function Description
0	I/O	4	P0.0–P0.3	FF0H	4-bit I/O port. 1-bit or 4-bit read/write and test is possible. 4-bit unit pull-up resistor selectable by program
1	I	4	P1.0–P1.3	FF1H	4-bit input port. 1-bit and 4-bit read/test is possible. 3-bit pull-up resistors are assignable by software to pins P1.0, P1.1, and P1.2.
2–6	I/O	20	P2.0–P2.3 P3.0–P3.3 P4.0–P4.3 P5.0–P5.3 P6.0–P6.3	FF2H–FF6H	Same as P0
7-13	O	28	P7.0–P7.3 P8.0–P8.3 P9.0–P9.3 P10.0–P10.3 P11.0–P11.3 P12.0–P12.3 P13.0–P13.3	FF7H–FFDH	1–bit or 4-bit I/O port. LCD segment output port.

Table 17. Port Pin Status During Instruction Execution

Instruction Type	Example	Input Mode Status	Output Mode Status

1-bit test
1-bit input
4-bit input
8-bit input
	BTST P0.1
	LDB C,P1.3
	LD A,P5
	LD EA,P4
		Input or test data at each pin	Input or test data at output latch

1-bit output	BITR P2.3	Output latch contents undefined	Output pin status is modified
4-bit output 8-bit output	LD P2,A LD P4,EA	Transfer accumulator data to the output latch	Transfer accumulator data to the output pin

PORT MODE FLAGS (PM FLAGS)

You use port mode flags (PM) to configure I/O ports to input or output mode by setting or clearing the corresponding port I/O buffer. PM flags are stored in three 8-bit registers and are addressable by 8-bit write instructions only.

For convenient program reference, PM flags are organized into four groups — PMG0, PMG1, PMG2, and PMG3, as shown in Table 33. PMG0 contains settings for port 0, PMG1 for ports 2 and 3, PMG2 for ports 4 and 5, and PMG3 for port 6.

When a PM flag is "0", the port is set to input mode; when it is "1", the port is enabled for output. A reset operation clears all port mode flags to logical zero, automatically configuring the corresponding I/O ports to input mode.

Table 18. Port Mode Group Flags

PM Group ID	Address	Bit 3	Bit 2	Bit 1	Bit 0
PMG0	FE6H	PM0.3	PM0.2	PM0.1	PM0.0
	FE7H	"0"	"0"	"0"	"0"
PMG1	FE8H	PM2.3	PM2.2	PM2.1	PM2.0
	FE9H	PM3.3	PM3.2	PM3.1	PM3.0
PMG2	FEAH	PM4.3	PM4.2	PM4.1	PM4.0
	FEBH	PM5.3	PM5.2	PM5.1	PM5.0
PMG3	FECH	PM6.3	PM6.2	PM6.1	PM6.0
	FEDH	"0"	"0"	"0"	"0"

NOTE: If a PMGn bit = "0", the corresponding I/O pin is set to input mode. If the PMGn bit = "1", the pin is set to output mode. All flags are cleared to "0" by a reset operation.

☞ PROGRAMMING TIP — Configuring I/O Ports to Input or Output

Configure the port 0 pins to output mode:

```
BITS    EMB
SMB     15
LD      EA,#0FH
LD      PMG0,EA          ;  P0 ← Output mode
```

PULL-UP RESISTOR MODE REGISTER (PUMOD)

he pull-up resistor mode register (PUMOD) is an 8-bit register used to assign internal pull-up resistors by software to specific I/O ports.

When a configurable I/O port pin is used as an output pin, its assigned pull-up resistor is automatically disabled, even though the pin's pull-up is enabled by a corresponding PUMOD bit setting.

PUMOD is addressable by 8-bit write instructions only. A reset operation clears PUMOD register values to logic zero, automatically disconnecting all software-assignable port pull-up resistors.

Table 19. Pull-Up Resistor Mode Register (PUMOD) Organization

PUMOD ID	Address	Bit 3	Bit 2	Bit 1	Bit 0
PUMOD	FDCH	PUR3	PUR2	PUR1	PUR0
	FDDH	"0"	PUR6	PUR5	PUR4

NOTE: When a PURn bit = "1", a pull-up resistor is assigned to the corresponding I/O port: PUR3 is for port 3, PUR2 for port 2, and so on.

☞ PROGRAMMING TIP — Enabling and Disabling I/O Port Pull-Up Resistors

P6 enable pull-up resistors.

```
BITS    EMB
SMB     15
LD      EA,#40H
LD      PUMOD,EA        ;  P6 enable
```

PIN ADDRESSING FOR OUTPUT PORTS 7–13

The addresses for the ports 7–13 1-bit output pin latches are located in bank 15 of data memory. To address ports 7–13 output pins, use the settings EMB = 1 and SMB = 15. The LCD port register, LPOT is used to control whether the pin address is used for LCD data output or for normal data output.

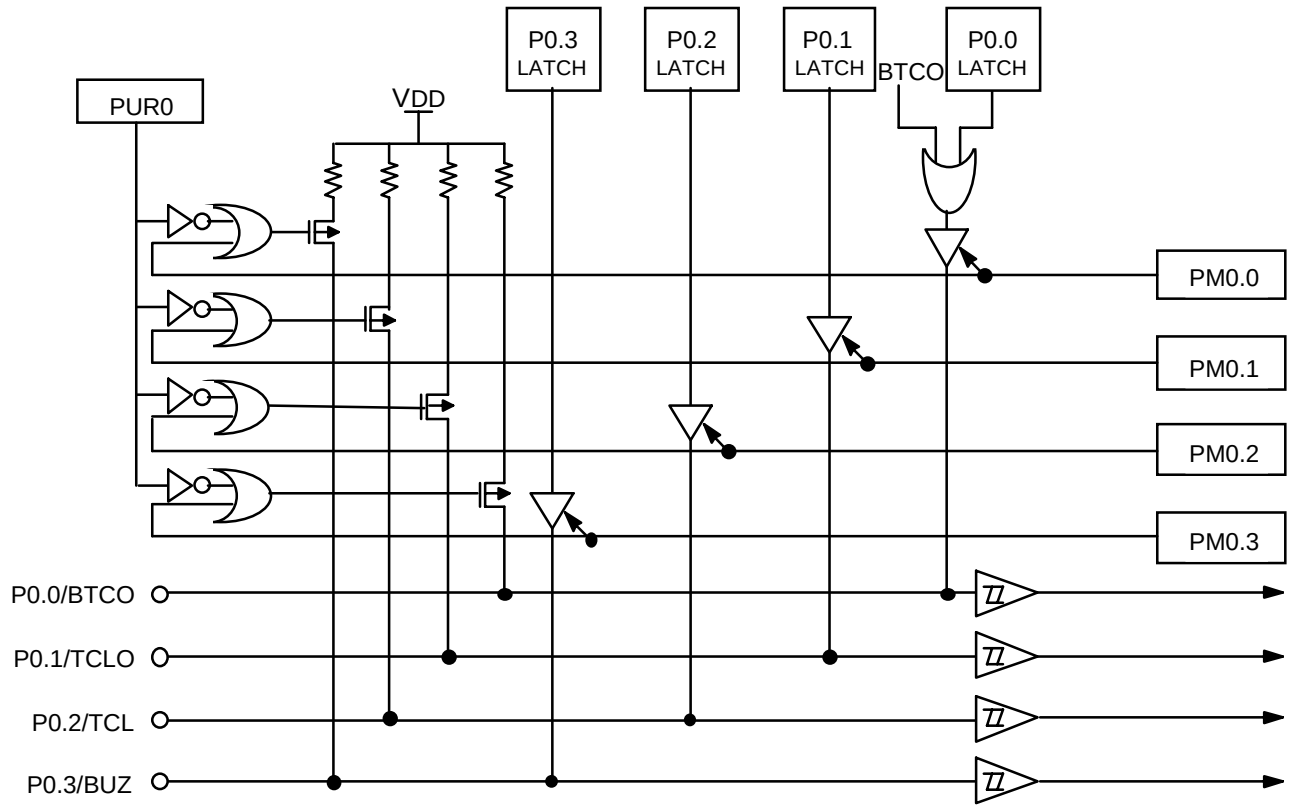
Each address in RAM bank 15 corresponds to a 4-bit register location. After a RESET, the values contained in the ports 7–13 output buffer are left undetermined.

Table 35 shows ports 7–13 pin addresses and also the corresponding LCD segment names if the pins are used to output LCD segment data. Pin addresses that are not used for LCD segment output can be used for normal 1-bit output.

Table 20. Ports 7–13 Pin Addresses and LCD Segment Correspondence

Ports 7–13 Pin Number	RAM Address	LCD Segment
P7.0–P7.3	FF7H	SEG0–SEG3
P8.0–P8.3	FF8H	SEG4–SEG7
P9.0–P9.3	FF9H	SEG8–SEG11
P10.0–P10.3	FFAH	SEG12–SEG15
P11.0–P11.3	FFBH	SEG16–SEG19
P12.0–P12.3	FFCH	SEG20–SEG23
P13.0–P13.3	FFDH	SEG24–SEG27

PORT 0 CIRCUIT DIAGRAM



NOTE: When a port pin acts as an output, its pull-up resistor is automatically disabled, even though the port's pull-up resistor is enabled by bit settings to the pull-up resistor mode register (PUMOD).

Figure 32. Port 0 Circuit Diagram

PORT 1 CIRCUIT DIAGRAM

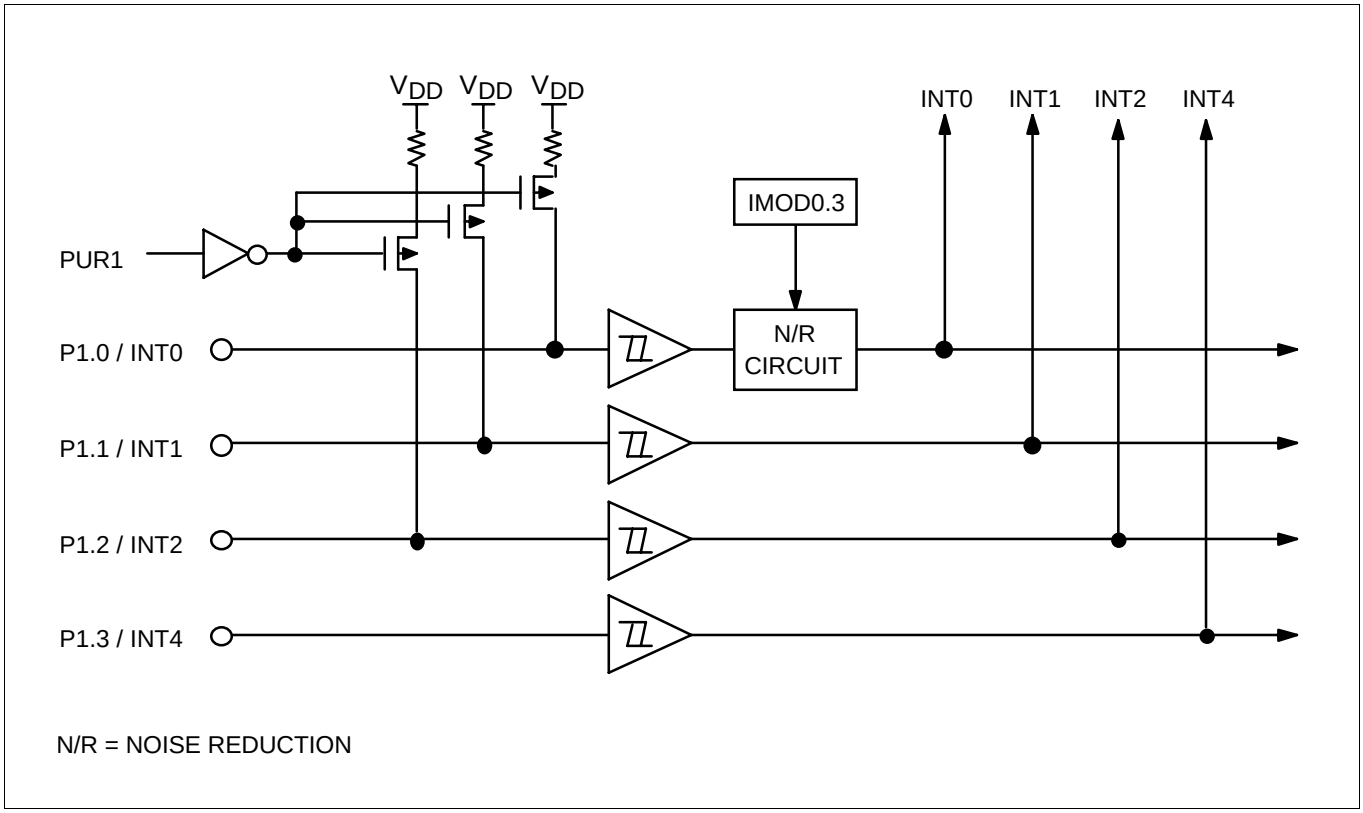
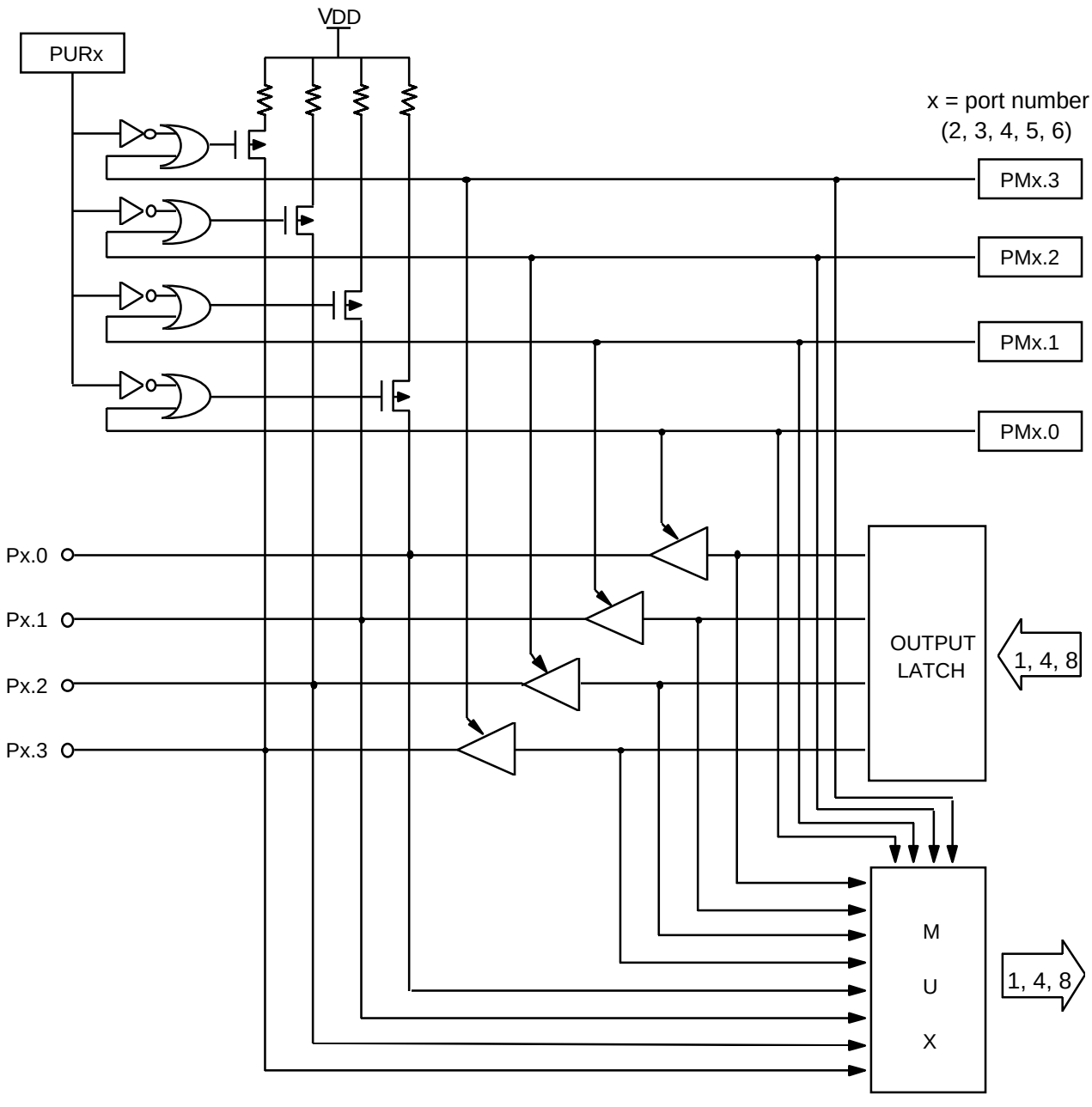


Figure 33. Port 1 Circuit Diagram

PORTS 2, 3, 4, 5, 6 CIRCUIT DIAGRAM



NOTE: When a port pin acts as an output, its pull-up resistor is automatically disabled, even though the port's pull-up resistor is enabled by bit settings to the pull-up resistor mode register (PUMOD).

Figure 34. Ports 2, 3, 4, 5, 6 Circuit Diagram

TIMERS and TIMER/COUNTER

The KS57C3108 microcontroller has the following timer and timer/counter modules:

— 8-bit basic timer (BT)
— 8-bit timer/counter (TC0)
— Watch timer (WT)

The 8-bit basic timer (BT) is the microcontroller's main interval timer. It generates an interrupt request at time interval specified by loading values to the basic timer mode register. When the value of the basic timer counter register BCNT overflows, a pulse is output to the basic timer output pin, BTCO. The basic timer also serves as a 'watchdog timer' and can be used to program the clock oscillation stabilization time whenever Stop mode is released by an interrupt and after a reset.

The 8-bit timer/counter (TC0) is a programmable timer/counter that is used primarily for event counting and for clock frequency modification and output. In addition, TC0 generates a clock signal that can be used by the serial I/O interface.

The watch timer (WT) consists of an 8-bit watch timer mode register, a clock selector, and a frequency divider circuit. Watch timer functions include real-time and watch-time measurement, main and subsystem clock interval timing, and buzzer output generation. It also generates a clock signal for the LCD controller/driver.

BASIC TIMER (BT)

The basic timer generates interrupt requests at precise intervals, based on the frequency of the system clock. Timer pulses are output from the basic timer's counter register BCNT to the output pin BTCO when an overflow occurs in the counter register BCNT.

You can use the basic timer as a watchdog timer for monitoring system events or use BT output to stabilize clock oscillation whenever Stop mode is released by an interrupt or by RESET. Bit settings in the basic timer mode register BMOD settings enable or disable the basic timer, select input clock frequency, and control interrupt or stabilization intervals.

Interval Timer Function

The basic timer's primary function is to measure elapsed time intervals. The standard time interval is equal to 256 basic timer clock pulses.

To restart the basic timer, one bit setting is required: BMOD.3 is set to "1". The input clock frequency and the interrupt and stabilization interval are selected by loading the appropriate bit values to BMOD.2–BMOD.0.

The 8-bit counter register, BCNT, is incremented each time a clock signal is detected that corresponds to the frequency selected by BMOD. BCNT continues incrementing as it counts BT clocks until an overflow occurs. The overflow sets the BT interrupt request flag (IRQB) to "1" to signal that the time interval has elapsed. An interrupt request is then generated, BCNT is cleared to "0", and counting continues from 00H.

Watchdog Timer Function

You can use the basic timer as a watchdog timer to signal the occurrence of specific system events. Each time BCNT overflows, an overflow signal is sent to the basic timer clock output pin, BTCO.

When the IRQB flag is "1" and the interrupt is requested, a BCNT overflow signal is sent to the P0.0 latch to be output at the BTCO pin.

Oscillation Stabilization Interval Control

BMOD.0–BMOD.2 are used to select the input clock frequency for the basic timer. This setting also determines the time interval (also referred to as 'wait time') that is required to stabilize clock signal oscillation when power-down mode is released by an interrupt. When RESET is input, the standard stabilization interval for system clock oscillation is 29.1 ms at 4.5 MHz.

Table 21. Basic Timer Register Overview

Register Name	Type	Description	Size	RAM Address	Addressing Mode	Reset Value
BMOD	Control	Controls the clock frequency (mode) of the basic timer; also, the oscillation stabilization interval after power-down mode release or RESET	4-bit	F85H	4-bit write-only; BMOD.3: 1-bit writeable	"0"
BCNT	Counter	Counts clock pulses matching the BMOD frequency setting	8-bit	F86H–F87H	8-bit read-only	'x'
BOE	Flag	Controls output of basic timer output latch to the BTCO pin	1-bit	F92H.1	1-bit read/write	"0"

NOTE: x' means the value is undetermined after RESET.

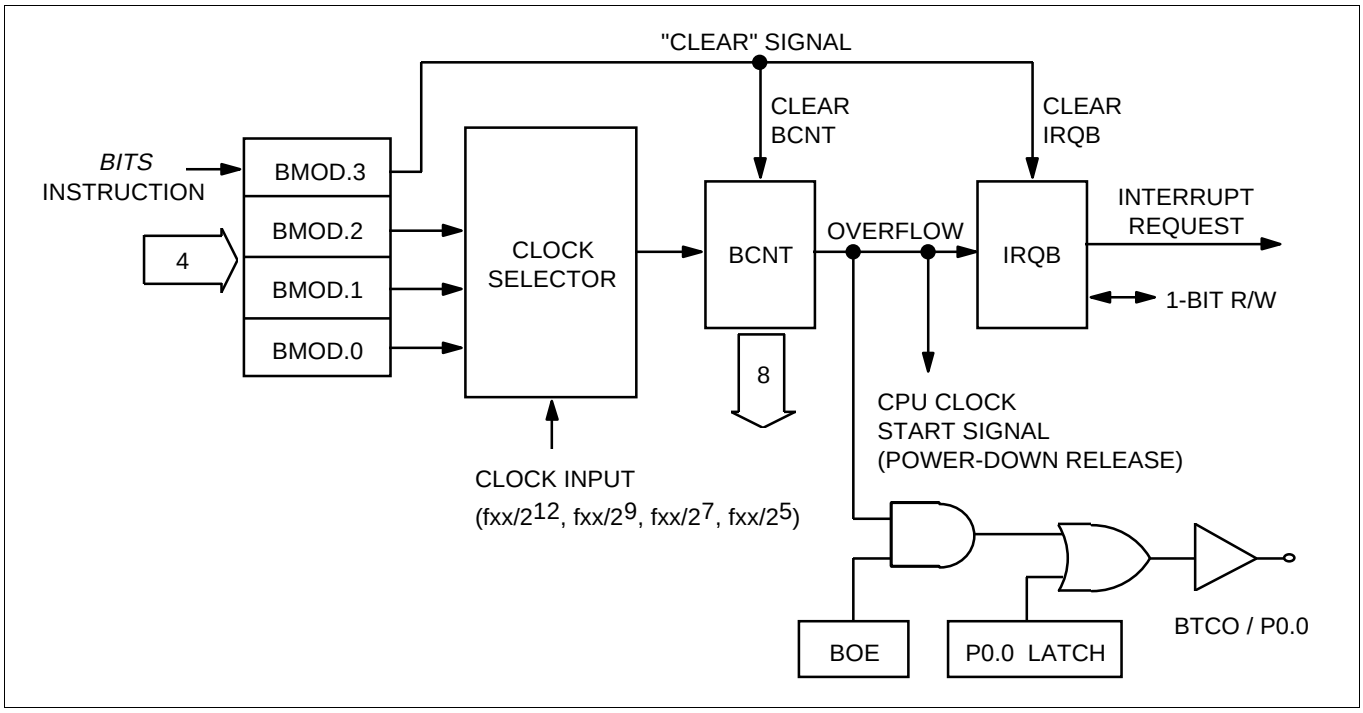


Figure 35. Basic Timer Circuit Diagram

BASIC TIMER MODE REGISTER (BMOD)

The basic timer mode register, BMOD, is a 4-bit write-only register. BMOD.3, the basic timer start control bit, is also 1-bit addressable. A reset operation clears BMOD to '00H', setting interrupt request signal generation to the longest interval (Please note that the BT counter runs continuously).

BMOD settings control the following functions:

— Restart the basic timer

— Control the frequency of clock signal input to the basic timer

— Set the time interval for clock oscillation stabilization after the release of Stop mode by an interrupt

BASIC TIMER MODE REGISTER (BMOD)

By loading different values into the BMOD register, you can dynamically modify the basic timer clock frequency during program execution. Four BT frequencies, ranging from fxx/2^{12} to fxx/2^5, are selectable. Because BMOD's reset value is "0", the default clock frequency setting is fxx/2^{12}.

The most significant bit of the BMOD register, BMOD.3, is used to restart the basic timer. When BMOD.3 is set to "1" (enabled) by a 1-bit write instruction, the contents of the BT counter register (BCNT) and the BT interrupt request flag (IRQB) are both cleared to "0", and the timer is restarted.

The combination of bit settings in the remaining three registers — BMOD.2, BMOD.1, and BMOD.0 — determine the clock input frequency and oscillation stabilization interval.

Table 22. Basic Timer Mode Register (BMOD) Organization

BMOD.3	Basic Timer Enable/Disable Control Bit
1	Restart basic timer; clear IRQB, BCNT, and BMOD.3 to "0"

BMOD.2	BMOD.1	BMOD.0	Basic Timer Input Clock	Oscillation Stabilization
0	0	0	fxx/2^{12} (1.098 kHz)	2^{20}/fxx (233 ms)
0	1	1	fxx/2^9 (8.789 kHz)	2^{17}/fxx (29.1 ms)
1	0	1	fxx/2^7 (35.15 kHz)	2^{15}/fxx (7.28 ms)
1	1	1	fxx/2^5 (140.6 kHz)	2^{13}/fxx (1.82 ms)

NOTES:
1. Clock frequencies and stabilization intervals assume a system oscillator clock frequency (fxx) of 4.5 MHz.
2. fxx = selected system clock frequency.
3. Oscillation stabilization time is the time required to stabilize clock signal oscillation after Stop mode is released. The data in the column 'Oscillation Stabilization' are also called the "Interrupt Interval Times."
4. The standard stabilization time for system clock oscillation following a RESET is 29.1 ms at 4.5 MHz.

BASIC TIMER COUNTER (BCNT)

BCNT is an 8-bit counter for the basic timer. It can be addressed using 8-bit read instructions. The reset operation leaves the BCNT counter value undetermined. BCNT is automatically cleared to "0" whenever the BMOD register control bit (BMOD.3) is set to "1" to restart the basic timer. It is incremented each time a clock pulse of the frequency determined by the current BMOD bit settings is detected.

When BCNT has incremented to hexadecimal 'FFH' (255 clock pulses), it is cleared to '00H' and an overflow is generated. The overflow causes the interrupt request flag, IRQB, to be set to "1". When the interrupt request is generated, BCNT immediately resumes counting incoming clock signals.

NOTE

Always execute a BCNT read operation twice to eliminate the possibility of reading unstable data while the counter is incrementing. If, after two consecutive read operations, the BCNT values match, you can select the latter value as valid data. Until the results of the consecutive reads match, however, the read operation must be repeated until the validation condition is met.

BASIC TIMER OUTPUT ENABLE FLAG (BOE)

The BOE flag value enables and disables basic timer output to the BTCO pin. When BOE is "0", basic timer output to the BTCO pin is disabled; when it is "1", BT output to the BTCO pin is enabled.

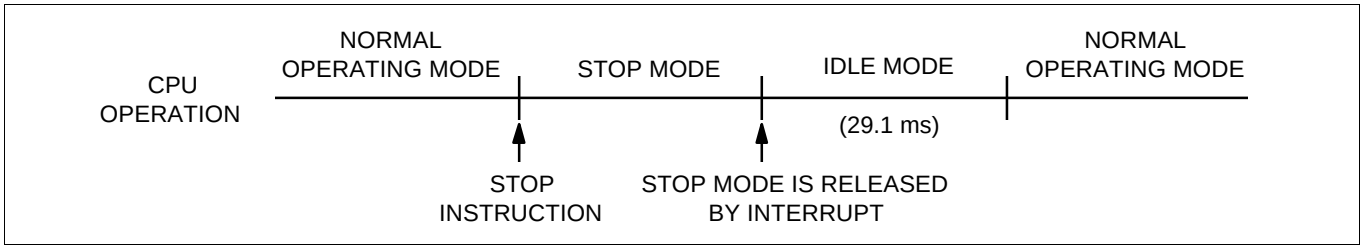
☞ PROGRAMMING TIP — Using the Basic Timer

1. To read the basic timer count register (BCNT):

```
        BITS    EMB
        SMB     15
BCNTR   LD      EA,BCNT
        LD      YZ,EA
        LD      EA,BCNT
        CPSE    EA,YZ
        JR      BCNTR
```

2. Whenever Stop mode is released by an interrupt, set the oscillation stabilization interval to 29.1 ms:

```
        BITS    EMB
        SMB     15
        LD      A,#0BH
        LD      BMOD,A          ;  Wait time is 29.1 ms
        STOP                    ;  Enter Stop power-down mode
        NOP
        NOP
```



3. To set the basic timer interrupt interval time to 1.82 ms (at 4.5 MHz):

```
        BITS    EMB
        SMB     15
        LD      A,#0FH
        LD      BMOD,A
        EI
        BITS    IEB             ;  Basic timer interrupt enable flag is set to "1"
```

4. Clear BCNT and the IRQB flag and restart the basic timer:

```
        BITS    EMB
        SMB     15
        BITS    BMOD.3
```

8-BIT TIMER/COUNTER 0 (TC0)

Timer/counter 0 (TC0) is used to count system 'events' by identifying the transition (High-to-Low or Low-to-High) of incoming square wave signals. To indicate that an event has occurred, or that a specified time interval has elapsed, TC0 generates an interrupt request. By counting signal transitions and comparing the current counter value with the reference register value, TC0 can be used to measure specific time intervals.

TC0 has a reloadable counter that consists of two parts: an 8-bit reference register (TREF0) into which you write the counter reference value, and an 8-bit counter register (TCNT0) whose value is automatically incremented by counter logic.

An 8-bit mode register, TMOD0, is used to activate the timer/counter and to select the basic clock frequency to be used for timer/counter operations. To dynamically modify the basic frequency, you can load new values into the TMOD0 register during program execution.

TC0 FUNCTION SUMMARY

8-bit programmable timer	Generates interrupts at specific time intervals based on the selected clock frequency.
External event counter	Counts various system events based on edge detection of external clock signals at the TC0 input pin, TCL. To start the event counting operation, TMOD0.2 is set to "1" and TMOD0.6 is cleared to "0".
Arbitrary frequency output	Outputs selectable clock frequencies to the TC0 output pin, TCLO.
External signal divider	Divides the frequency of an incoming external clock signal according to a modifiable reference value (TREF0), and outputs the modified frequency to the TCLO pin.
Serial I/O clock source	Outputs a modifiable clock signal for use as the SCK clock source.

TC0 COMPONENT SUMMARY

Mode register (TMOD0)	Activates the timer/counter and selects the internal clock frequency or the external clock source at the TCL pin.
Reference register (TREF0)	Stores the reference value for the desired number of clock pulses between interrupt requests.
Counter register (TCNT0)	Counts internal or external clock pulses based on the bit settings in TMOD0 and TREF0.
Clock selector circuit	Together with the mode register (TMOD0), lets you select one of four internal clock frequencies or an external clock.
8-bit comparator	Determines when to generate an interrupt by comparing the current value of the counter register (TCNT0) with the reference value previously programmed into the reference register (TREF0).
Output latch (TOL0)	Where a clock pulse is stored pending output to the serial I/O circuit or to the TC0 output pin, TCLO. When the contents of the TCNT0 and TREF0 registers coincide, the timer/counter interrupt request flag (IRQT0) is set to "1", the status of TOL0 is inverted, and an interrupt is generated.
Output enable flag (TOE0)	Must be set to "1" before the contents of the TOL0 latch can be output to TCLO.
Interrupt request flag (IRQT0)	Cleared when TC0 operation starts and the TC0 interrupt service routine is executed and enabled whenever the counter value and reference value coincide.
Interrupt enable flag (IET0)	Must be set to "1" before the interrupt requests generated by timer/counter 0 can be processed.

Table 23. TC0 Register Overview

Register Name	Type	Description	Size	RAM Address	Addressing Mode	Reset Value
TMOD0	Control	Controls TC0 enable/disable (bit 2); clears and resumes counting operation (bit 3); sets input clock and clock frequency (bits 6–4)	8-bit	F90H–F91H	8-bit write-only; (TMOD0.3 is also 1-bit writeable)	"0"
TCNT0	Counter	Counts clock pulses matching the TMOD0 frequency setting	8-bit	F94H–F95H	8-bit read-only	"0"
TREF0	Reference	Stores reference value for the timer/counter 0 interval setting	8-bit	F96H–F97H	8-bit write-only	FFH
TOE0	Flag	Controls timer/counter 0 output to the TCLO pin	1-bit	F92H.2	1-bit write-only	"0"

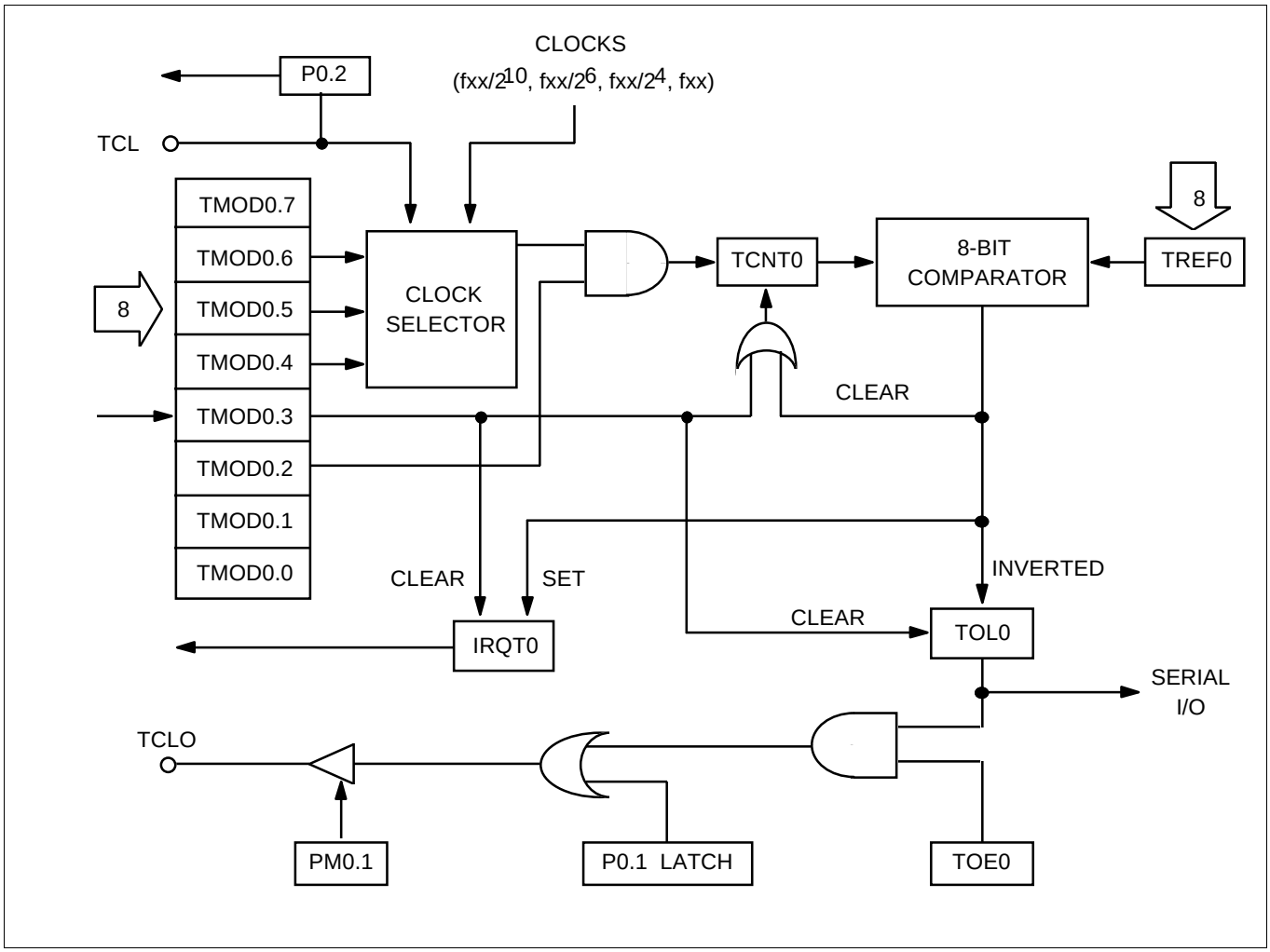


Figure 36. TC0 Circuit Diagram

TC0 PROGRAMMABLE TIMER/COUNTER FUNCTION

You can program timer/counter 0 to generate interrupt requests at various intervals based on the selected system clock frequency. Its 8-bit TC0 mode register TMOD0 is used to activate the timer/counter and to select the clock frequency.

The reference register TREF0 stores the value for the number of clock pulses to be generated between interrupt requests. The counter register, TCNT0, counts the incoming clock pulses, which are compared to the TREF0 value as TCNT0 is incremented. When there is a match (TREF0 = TCNT0), an interrupt request is generated.

To program timer/counter 0 to generate interrupt requests at specific intervals, choose one of four internal clock frequencies (divisions of the system clock, fxx) and load a counter reference value into the TREF0 register. TCNT0 is incremented each time an internal counter pulse is detected with the reference clock frequency specified by TMOD0.4–TMOD0.6 settings.

To generate an interrupt request, the TC0 interrupt request flag (IRQT0) is set to "1", the status of TOL0 is inverted, and the interrupt is generated. The content of TCNT0 is then cleared to 00H and TC0 continues counting. The interrupt request mechanism for TC0 includes an interrupt enable flag (IET0) and an interrupt request flag (IRQT0).

TC0 OPERATION SEQUENCE

The general sequence of operations for using TC0 can be summarized as follows:

1. Set TMOD0.2 to "1" to enable TC0
2. Set TMOD0.6 to "1" to enable the system clock (fxx) input
3. Set TMOD0.5 and TMOD0.4 bits to desired internal frequency ($fxx/2^n$)
4. Load a value to TREF0 to specify the interval between interrupt requests
5. Set the TC0 interrupt enable flag (IET0) to "1"
6. Set TMOD0.3 bit to "1" to clear TCNT0, IRQT0, and TOL0, and start counting
7. TCNT0 increments with each internal clock pulse
8. When the comparator shows TCNT0 = TREF0, the IRQT0 flag is set to "1"
9. Output latch (TOL0) logic toggles high or low
10. Interrupt request is generated
11. TCNT0 is cleared to 00H and counting resumes
12. Programmable timer/counter operation continues until TMOD0.2 is cleared to "0".

TC0 EVENT COUNTER FUNCTION

Timer/counter 0 can monitor or detect system 'events' by using the external clock input at the TCL pin (I/O port 0.2) as the counter source. The TC0 mode register selects rising or falling edge detection for incoming clock signals. The counter register TCNT0 is incremented each time the selected state transition of the external clock signal occurs.

With the exception of the different TMOD0.4–TMOD0.6 settings, the operation sequence for TC0's event counter function is identical to its programmable timer/counter function. To activate the TC0 event counter function,

— Set TMOD0.2 to "1" to enable TC0;

— Clear TMOD0.6 to "0" to select the external clock source at the TCL pin;

— Select TCL edge detection for rising or falling signal edges by loading the appropriate values to TMOD0.5 and TMOD0.4.

— P0.2 must be set to input mode.

Table 24. TMOD0 Settings for TCL Edge Detection

TMOD0.5	TMOD0.4	TCL Edge Detection
0	0	Rising edges
0	1	Falling edges

TC0 CLOCK FREQUENCY OUTPUT

Using timer/counter 0, you can output a modifiable clock frequency to the TC0 clock output pin, TCLO. To select the clock frequency, you load the appropriate value to the TC0 mode register, TMOD0. The clock interval is selected by loading the desired reference value into the reference register TREF0. To enable the output to the TCLO pin at I/O port 0.1, the following conditions must be met:

— TC0 output enable flag TOE0 must be set to "1"

— I/O mode flag for P0.1 (PM0.1) must be set to output mode ("1")

— Output latch value for P0.1 must be cleared to "0"

Each time TCNT0 overflows and an interrupt request is generated, the state of the output latch TOL0 is inverted and the TC0-generated clock signal is output to the TCLO pin.

☞ PROGRAMMING TIP — TC0 Signal Output to the TCLO Pin

Output a 30-ms pulse width signal to the TCLO pin:

```
BITS    EMB
SMB     15
LD      EA,#79H
LD      TREF0,EA
LD      EA,#4CH
LD      TMOD0,EA
LD      EA,#01H
LD      PMG0,EA          ;  P0.1 ← output mode
BITR    P0.1             ;  P0.1 clear
BITS    TOE0
```

TC0 SERIAL I/O CLOCK GENERATION

Timer/counter 0 can supply a clock signal to the clock selector circuit of the serial I/O interface for data shifter and clock counter operations (These internal SIO operations are controlled in turn by the SIO mode register, SMOD). This clock generation function enables you to adjust data transmission rates across the serial interface.

Use TMOD0 and TREF0 register settings to select the frequency and interval of the TC0 clock signals to be used as SCK input to the serial interface. The generated clock signal is then sent directly to the serial I/O clock selector circuit — not through the port 0.1 latch and TCLO pin (the TOE0 flag may be disabled).

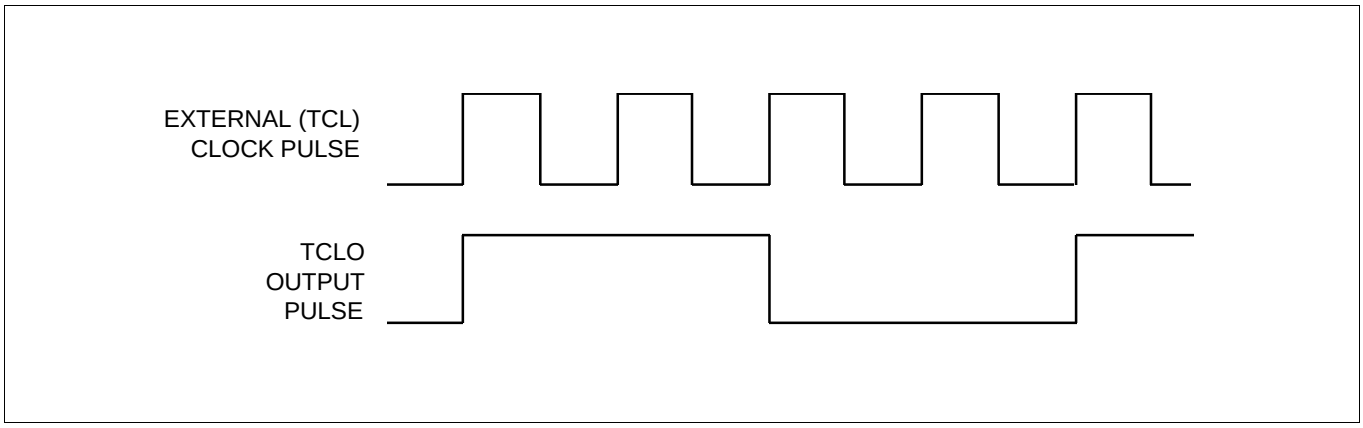
TC0 EXTERNAL INPUT SIGNAL DIVIDER

By selecting an external clock source and loading a reference value into the TC0 reference register, TREF0, you can divide the incoming clock signal by the TREF0 value and then output this modified clock frequency to the TCLO pin. The sequence of operations used to divide external clock input can be summarized as follows:

1. Load a signal divider value to the TREF0 register
2. Clear TMOD0.6 to "0" to enable external clock input at the TCL pin
3. Set TMOD0.5 and TMOD0.4 to desired TCL signal edge detection
4. Set port 0.1 mode flag (PM0.1) to output ("1")
5. Set P0.1 output latch to "0"
6. Set TOE0 flag to "1" to enable output of the divided frequency to the TCLO pin

☞ PROGRAMMING TIP — External TCL Clock Output to the TCLO Pin

Output external TCL clock pulse to the TCLO pin (divided by four):




```
BITS    EMB
SMB     15
LD      EA,#01H
LD      TREF0,EA
LD      EA,#0CH
LD      TMOD0,EA
LD      EA,#02H
LD      PMG0,EA          ;  P0.1 ← output mode
BITR    P0.1             ;  P0.1 clear
BITS    TOE0
```

TC0 MODE REGISTER (TMOD0)

TMOD0 is the 8-bit mode control register for timer/counter 0. It is addressable by 8-bit write instructions. One bit, TMOD0.3, is also 1-bit writeable. RESET clears TMOD0 to '00H' and disables TC0 operations.

F90H	TMOD0.3	TMOD0.2	"0"	"0"
F91H	"0"	TMOD0.6	TMOD0.5	TMOD0.4

TMOD0.2 is the enable/disable bit for timer/counter 0. When TMOD0.3 is set to "1", the contents of TCNT0, IRQT0, and TOL0 are cleared, counting starts from '00H', and TMOD0.3 is automatically reset to "0" for normal TC0 operation. When TC0 operation stops (TMOD0.2 = "0"), the contents of the TC0 counter register TCNT0 are retained until TC0 is re-enabled.

The TMOD0.6, TMOD0.5, and TMOD0.4 bit settings are used together to select the TC0 clock source. This selection involves two variables:

— Synchronization of timer/counter operations with either the rising edge or the falling edge of the clock signal input at the TCL0 pin, and

— Choosing of one of four frequencies, based on division of the incoming system clock frequency, for use in internal TC0 operation.

Table 25. TC0 Mode Register (TMOD0) Organization

Bit Name	Setting	Resulting TC0 Function	Address
TMOD0.7	0	Always logic zero	F91H
TMOD0.6 TMOD0.5 TMOD0.4	0,1	Specify input clock edge and internal frequency	
TMOD0.3	1	Clear TCNT0, IRQT0, and TOL0 and resume counting immediately (This bit is automatically cleared to "0" after counting resumes.)	F90H
TMOD0.2	0	Disable timer/counter 0; retain TCNT0 contents	
	1	Enable timer/counter 0	
TMOD0.1	0	Always "0"	
TMOD0.0	0	Always "0"	

Table 26. TMOD0.6, TMOD0.5, and TMOD0.4 Bit Settings

TMOD0.6	TMOD0.5	TMOD0.4	Resulting Counter Source and Clock Frequency
0	0	0	External clock input (TCL0) on rising edges
0	0	1	External clock input (TCL0) on falling edges
1	0	0	$fxx/2^{10}$ (4.39 kHz)
1	0	1	$fxx/2^{6}$ (70.3 kHz)
1	1	0	$fxx/2^{4}$ (281 kHz)
1	1	1	fxx = 4.5 MHz

NOTE: 'fxx' = selected system clock of 4.5 MHz.

☞ PROGRAMMING TIP — Restarting TC0 Counting Operation

1. Set TC0 timer interval to 4.39 kHz:

```
BITS    EMB
SMB     15
LD      EA,#4CH
LD      TMOD0,EA
EI
BITS    IET0
```

2. Clear TCNT0, IRQT0, and TOL0 and restart TC0 counting operation:

```
BITS    EMB
SMB     15
BITS    TMOD0.3
```

TC0 COUNTER REGISTER (TCNT0)

The 8-bit counter register for timer/counter 0, TCNT0, is read-only and can be addressed by 8-bit RAM control instructions. RESET sets TCNT0 to '00H'.

Whenever TMOD0.3 is enabled, TCNT0 is cleared to '00H' and counting resumes. The TCNT0 register value is incremented each time an incoming clock signal is detected that matches the signal edge and frequency setting of the TMOD0 register (specifically, TMOD0.6, TMOD0.5, and TMOD0.4).

Each time TCNT0 is incremented, the new value is compared to the reference value stored in the TC0 reference buffer, TREF0. When TCNT0 = TREF0, an overflow occurs in the TCNT0 register, the interrupt request flag, IRQT0, is set to "1", and an interrupt request is generated to indicate that the programmed timer/counter interval has elapsed.

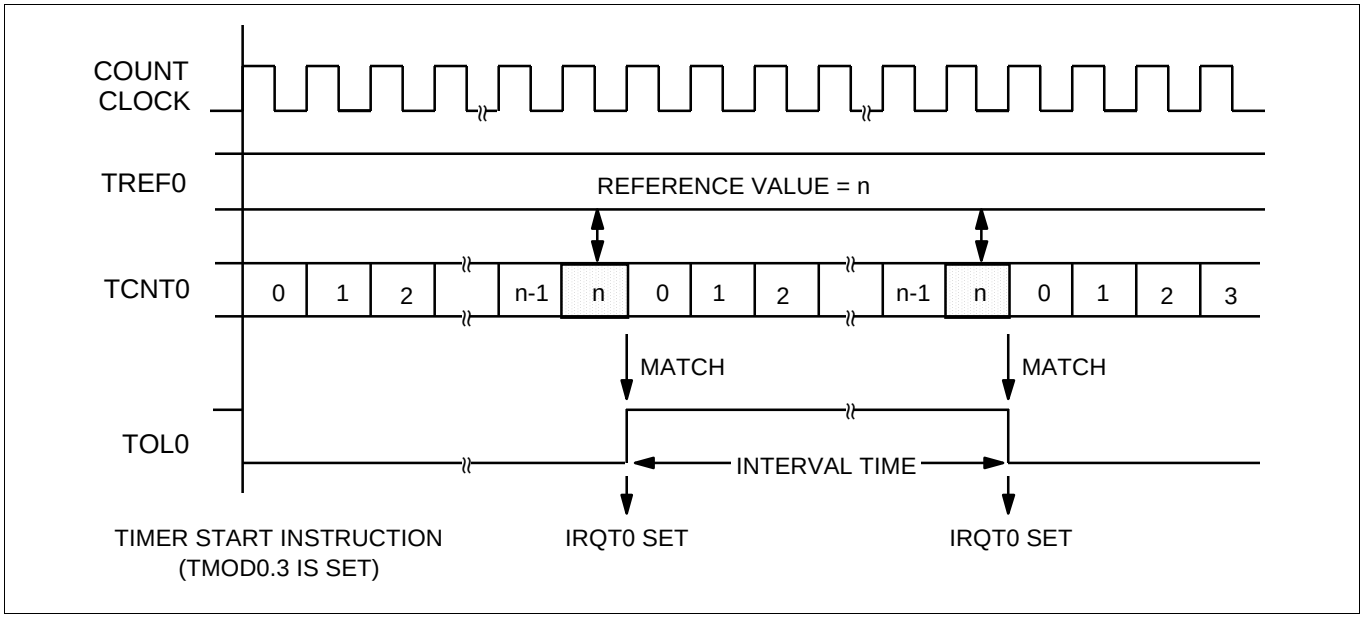


Figure 37. TC0 Timing Diagram

TC0 REFERENCE REGISTER (TREF0)

The TC0 reference register TREF0 is an 8-bit write-only register. It is addressable by 8-bit RAM control instructions. RESET initializes the TREF0 value to 'FFH'.

TREF0 is used to store a reference value to be compared to the incrementing TCNT0 register in order to identify an elapsed time interval. Reference values will differ depending upon the specific function that TC0 is being used to perform — as a programmable timer/counter, event counter, clock signal divider, or arbitrary frequency output source.

During timer/counter operation, the value loaded into the reference register is compared to the TCNT0 value. When TCNT0 = TREF0, the TC0 output latch (TOL0) is inverted and an interrupt request is generated to signal the interval or event. The TREF0 value, together with the TMOD0 clock frequency selection, determines the specific TC0 timer interval. Use the following formula to calculate the correct value to load to the TREF0 reference register:

$$\text{TC0 timer interval} = (\text{TREF0 value} + 1) \times \frac{1}{\text{TMOD0 frequency setting}}$$

$$(\text{TREF0 value} \bullet 0)$$

TC0 OUTPUT ENABLE FLAG (TOE0)

The 1-bit timer/counter 0 output enable flag TOE0 controls output from timer/counter 0 to the TCLO0 pin. TOE0 is addressable by 1-bit read and write instructions.

	(MSB)			(LSB)
F92H	"0"	**TOE0**	BOE	"0"

When you set the TOE0 flag to "1", the contents of TOL0 can be output to the TCLO0 pin. Whenever a RESET occurs, TOE0 is automatically cleared to "0", disabling all TC0 output. Even when the TOE0 flag is disabled, timer/counter 0 can continue to output an internally-generated clock frequency, at TOL0, to the serial I/O clock selector circuit.

TC0 OUTPUT LATCH (TOL0)

TOL0 is the output latch for timer/counter 0. When the 8-bit comparator detects a correspondence between the value of the counter register TCNT0 and the reference value stored in the TREF0 register, the TOL0 value is inverted — the latch toggles High-to-Low or Low-to-High. Whenever the state of TOL0 is switched, the TC0 signal is output.

TC0 output may be directed to the TCLO0 pin, or it can be output directly to the serial I/O clock selector circuit as the SCK signal.

Assuming TC0 is enabled, when TMOD0.3 is "1", the TOL0 latch is cleared to "0", along with the counter register TCNT0 and the interrupt request flag, IRQT0. Counting then resumes immediately. When TC0 is disabled (TMOD0.2 = "0"), the contents of the TOL0 latch are retained and can be read, if necessary.

☞ PROGRAMMING TIP — Setting a TC0 Timer Interval

To set a 30 ms timer interval for TC0, given fxx = 4.5 MHz, follow these steps.

1. Select the timer/counter 0 mode register with a maximum setup time of 58.3 ms (assume the TC0 counter clock = fxx/2^{10}, and TREF0 is set to FFH):

2. Calculate the TREF0 value:

$$30\ \text{ms} = \frac{\text{TREF0 value} + 1}{4.39\ \text{kHz}}$$

$$\text{TREF0} + 1 = \frac{30\ \text{ms}}{227\ \mu\text{s}} = 132.15 = 84\text{H}$$

$$\text{TREF0 value} = 84\text{H} - 1 = 83\text{H}$$

3. Load the value 79H to the TREF0 register:

```
BITS    EMB
SMB     15
LD      EA,#83H
LD      TREF0,EA
LD      EA,#4CH
LD      TMOD0,EA
```

WATCH TIMER

The watch timer is a multi-purpose timer which consists of three basic components:

— 8-bit watch timer mode register (WMOD)

— Clock selector

— Frequency divider circuit

Watch timer functions include real-time and watch-time measurement and interval timing for the main and subsystem clock. It is also used as a clock source for the LCD controller and for generating buzzer (BUZ) output.

Real-Time and Watch-Time Measurement

To start the watch timer, you set bit 2 of the watch timer mode register (WMOD.2) to "1". The interrupt request flag IRQW is then automatically set to "1" and interrupt requests commence in 0.5-second intervals.

Because the watch timer interrupt is a quasi-interrupt instead of a vectored interrupt, the IRQW flag must be cleared to "0" by program software whenever a requested interrupt service routine is executed.

Using a Main System or Subsystem Clock Source

The watch timer can generate interrupts based on the main system clock frequency or on the subsystem clock. When the zero bit of the WMOD register is set to "1", the watch timer uses the subsystem clock signal (fxt) as its source; if WMOD.0 = "0", the main system clock (fx) is used as the signal source, according to the following formula:

$$\text{Watch timer clock (fw)} = \frac{\text{Main system clock (fx)}}{128}$$

= 32.768 kHz (fx = 4.19 MHz)

This feature is useful for controlling timer-related operations during stop mode. When stop mode is engaged, the main system clock (fx) is halted, but the subsystem clock continues oscillating. By using the subsystem clock as the oscillation source during Stop mode, the watch timer can set the interrupt request flag IRQW to "1", thereby releasing Stop mode.

Clock Source Generation for LCD Controller

The watch timer supplies the clock frequency for the LCD controller (f_{LCD}). Therefore, if the watch timer is disabled, the LCD controller cannot not operate.

Buzzer Output Frequency Generator

The watch timer can generate a steady 2-kHz, 4-kHz, 8-kHz, or 16-kHz signal to the BUZ pin (given an oscillation clock of 4.5 MHz). To select the desired BUZ frequency , load the appropriate value to the WMOD register. You can then use this output to actuate an external buzzer sound. To generate a BUZ signal, three conditions must be met:

— The WMOD.7 register bit is set to "1"

— The output latch for I/O port 0.3 is cleared to "0"

— The port 0.3 output mode flag (PM0.3) set to 'output' mode

Timing Tests in High-Speed Mode

When you set WMOD.1 to "1", the watch timer functions in high-speed mode, generating an interrupt every 3.91 milliseconds with an oscillation clock of 4.19 MHz. At its normal speed (WMOD.1 = '0'), the watch timer generates an interrupt request every 0.5 seconds. High-speed mode is useful for timing events for program debugging sequences.

Check Subsystem Clock Level Feature

The watch timer can also check the input level of the subsystem clock by testing WMOD.3. If WMOD.3 is "1", the input level at the XT_{IN} pin is High; if WMOD.3 is "0", the input level at the XT_{IN} pin is Low.

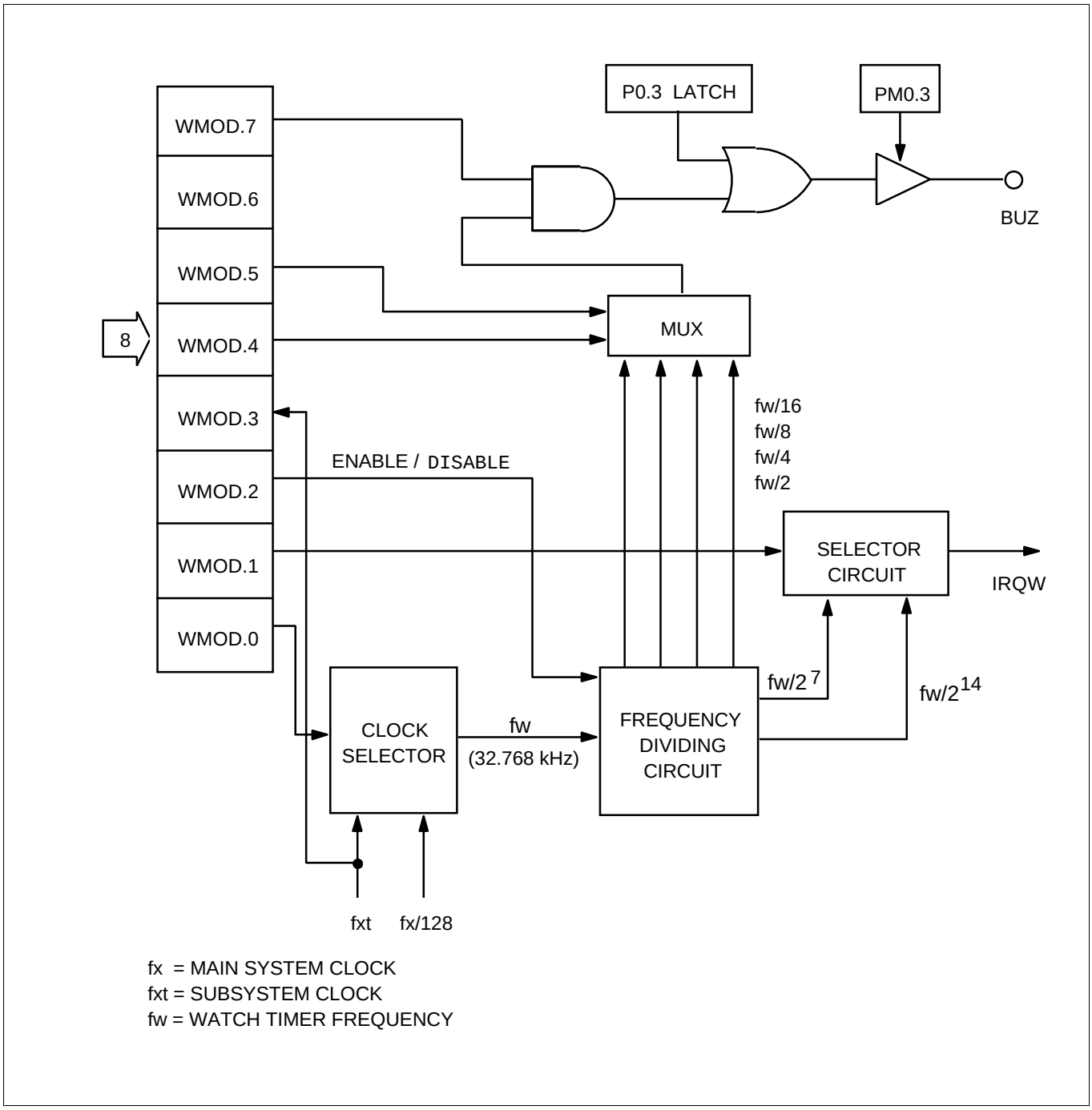


Figure 38. Watch Timer Circuit Diagram

WATCH TIMER MODE REGISTER (WMOD)

The watch timer mode register WMOD is used to select specific watch timer operations. It is 8-bit write-only addressable. An exception is WMOD bit 3 (the XT_{IN} input level control bit) which is 1-bit read-only addressable.

A RESET automatically sets WMOD.3 to the current input level of the subsystem clock, XT_{IN} (High, if "1"; Low, if "0"), and all other WMOD bits to "0".

Table 27. Watch Timer Mode Register (WMOD) Organization

Bit Name	Values		Function	Address
WMOD.7	0		Disable buzzer (BUZ) signal output	F89H
	1		Enable buzzer (BUZ) signal output	
WMOD.6	0		Always logic zero	
WMOD.5 – .4	0	0	2 kHz buzzer (BUZ) signal output	
	0	1	4 kHz buzzer (BUZ) signal output	
	1	0	8 kHz buzzer (BUZ) signal output	
	1	1	16 kHz buzzer (BUZ) signal output	
WMOD.3	0		Input level to XT_{in} pin is low	F88H
	1		Input level to XT_{in} pin is high	
WMOD.2	0		Disable watch timer; clear frequency dividing circuits	
	1		Enable watch timer	
WMOD.1	0		Normal mode; sets IRQW to 0.5 seconds	
	1		High-speed mode; sets IRQW to 3.91 ms	
WMOD.0	0		Select (fx/128) as the watch timer clock (fw)	
	1		Select subsystem clock as watch timer clock (fw)	

NOTE: Main system clock frequency (fx) is assumed to be 4.19 MHz; subsystem clock (fxx) is assumed to be 32.768 kHz.

☞ PROGRAMMING TIP — Using the Watch Timer

1. Select a subsystem clock as the LCD display clock, a 0.5 second interrupt, and 2 kHz buzzer enable:

```
BITS    EMB
SMB     15
LD      EA,#08H
LD      PMG0,EA          ;  P0.3 ← output mode
BITR    P0.3
LD      EA,#85H
LD      WMOD,EA
BITS    IEW
```

2. Sample real-time clock processing method:

```
CLOCK   BTSTZ   IRQW      ;  0.5 second check
        RET               ;  No, return
        •                 ;  Yes, 0.5 second interrupt generation
        •
        •                 ;  Increment HOUR, MINUTE, SECOND
```

A/D CONVERTER

The 8-bit analog-to-digital converter (ADC) has the following components:

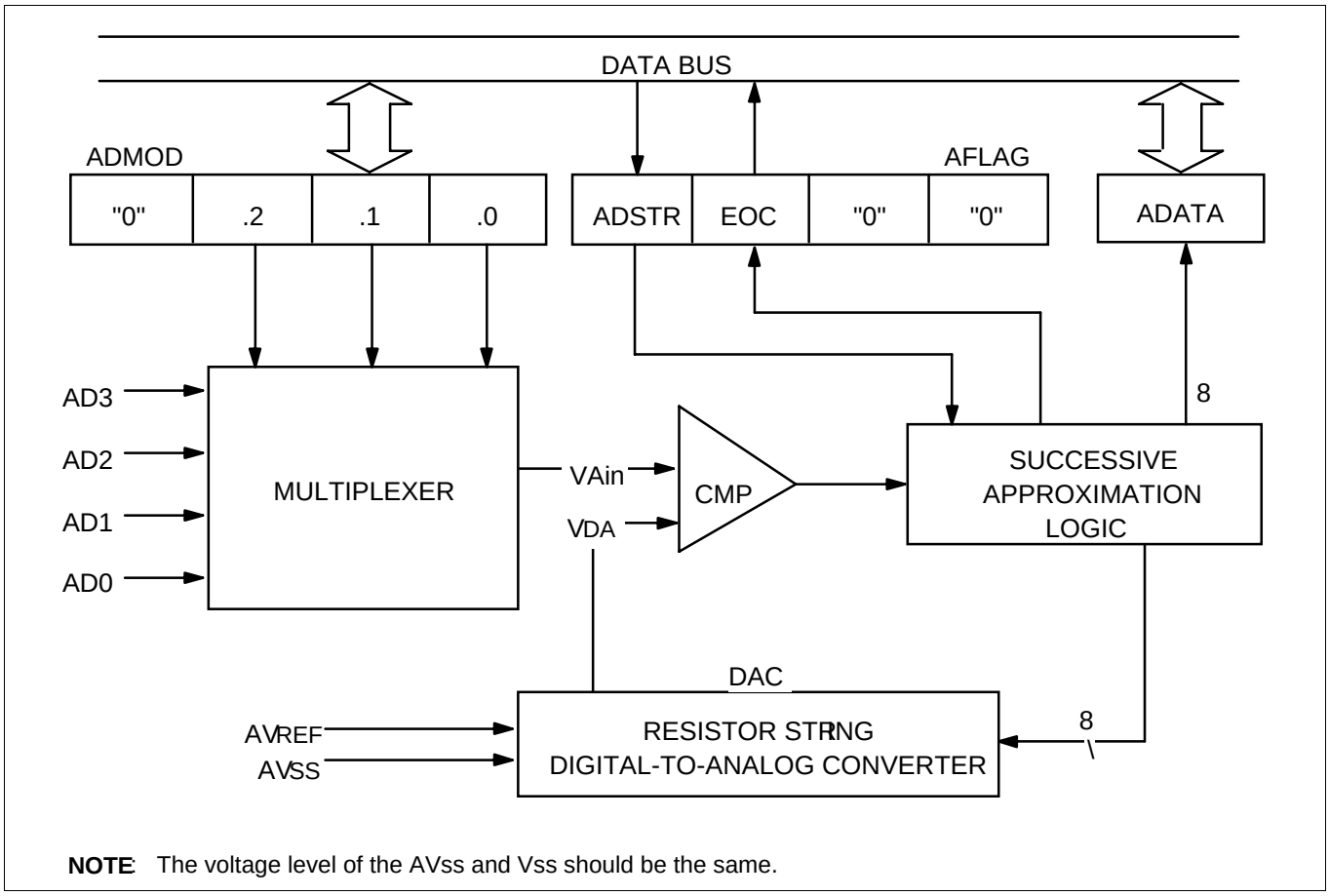
— Digital-to-analog converter
— Comparator
— ADC data register (ADATA)
— ADC mode register (ADMOD)
— ADC control register (AFLAG)
— Successive approximation logic

To operate the A/D converter, you select one of the four analog input channels by writing the appropriate value to the ADC mode register. To start the converter, you set the ADSTR flag in the control register AFLAG to "1". Conversion speed is determined by the oscillator frequency and the CPU clock.

When the A/D operation is complete, the EOC flag must be tested in order to verify that the conversion was successful. When the EOC value is "0", the converted digital values stored in the data register ADATA can be read.

Table 28. A/D Converter Component Overview

ADC Function	Mnemonic	Description
Digital-to-analog converter	DAC	Uses successive approximation logic to convert digital input into the reference analog voltage, V_{DA}. These V_{DA} values are input to the comparator and then compared to the multiplexed external analog source voltage, VA_{IN}.
Comparator	CMP	Compares the applied external analog input voltage, VA_{IN}, to the analog reference voltage (V_{DA}) that is generated by the DAC and writes the corresponding digital value to the ADATA register.
Digital data register	ADATA	Stores digital values as analog-to-digital conversion is completed.
ADC mode register	ADMOD	Used to select one of four analog channels as the input source for the analog data to be converted.
ADC control register	AFLAG	Contains the control flags used to start A/D converter operation and to monitor operational status.
Successive approximation logic	–	Control blocks in the A/D converter contain the successive approximation logic required to generate the analog reference voltage.



NOTE: The voltage level of the AVss and Vss should be the same.

Figure 39. A/D Converter Circuit Diagram

ADC PROCEDURE DESCRIPTION

Use these steps as a general guideline for writing A/D converter programs:

1. Select one of the four analog channels, ADC0–ADC3, as the analog input source. To do this, write the appropriate value to the ADMOD register, bits ADMOD.2–ADMOD.0.
2. Start the A/D converter by setting the ADSTR flag of the AFLAG register to "1".
3. When the converter starts, the EOC (End Of Conversion) flag in the AFLAG register is automatically set to "1", and the ADSTR flag is cleared to "0".
4. The analog-to-digital conversion speed is determined by the oscillator frequency and the CPU clock, as follows:

$$t_{CONV} = \frac{1}{fx} \times 80$$

For example, with a 4.5 MHz oscillator clock, the t_{conv} value is 17.77 µs. The 'tinit' value is determined by the instruction type and the speed of the CPU clock.

5. When conversion has been completed, the EOC flag is cleared automatically so that a check can be made to verify that the conversion was successful.
6. Converted digital values that have been stored in the 8-bit ADATA register can now be read. Conversion values are retained until the next A/D conversion operation starts.

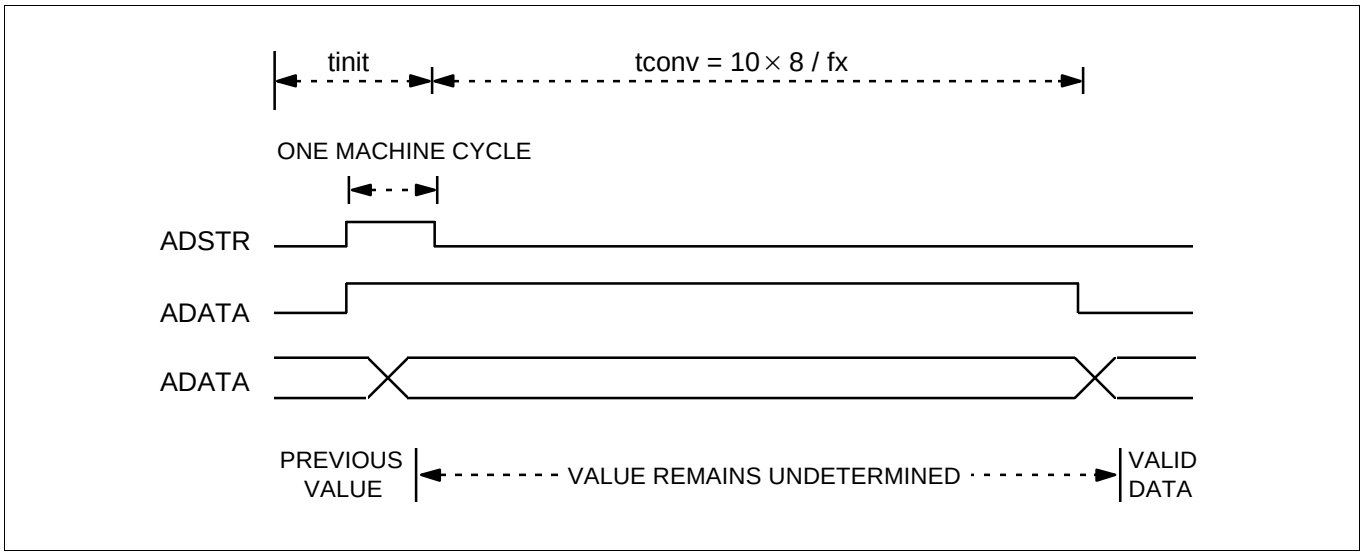


Figure 40. A/D Converter Timing Diagram

ADC DIGITAL-TO-ANALOG CONVERTER (DAC)

The 8-bit digital-to-analog converter (DAC) generates analog voltage reference values for the comparator. The DAC is a 256-step resistor string type digital-to-analog converter that uses successive approximation logic to convert digital input into the reference analog voltage, V_{DA}.

V_{DA} values are input from the DAC to the comparator where they are compared to the multiplexed external analog source voltage, VA_{in}. Since the DAC has 8-bit resolution, it generates the 256-step analog reference voltage as follows:

$$V_{DA} = V_{REF} \times \left(\frac{n}{256} \pm \frac{1}{512}\right) \text{ (1/2 LSB compensation)}$$

(n = 0 – 256, as determined by successive approximation logic)

Input channels ADC0–ADC3 may be used either for analog input to the A/D converter, or as normal input ports. Since only one of the four ports can be selected at one time as external source of analog data, the three remaining input ports are always available for other inputs.

ADC DATA REGISTER (ADATA)

The A/D converter data register, ADATA, is an 8-bit register in which digital data values are stored as an A/D conversion operation is completed. Digital values stored in ADATA are retained until another conversion operation is initiated. ADATA is addressable by 8-bit read instructions only.

ADC MODE REGISTER (ADMOD)

The analog-to-digital converter mode register ADMOD is a 4-bit register that is used to select one of four analog channels as the analog data input source. ADMOD is addressable by 1-bit or 4-bit read or write instructions.

RESET clears the ADMOD register to logic zero. Table 44 shows ADMOD.2–ADMOD.0 bit settings for selecting a channel for analog input:

Table 29. Disable A/D Converter Mode Register Settings

ADMOD.2	ADMOD.1	ADMOD.0	Effect of ADMOD Bit Setting
0	x	x	Disable A/D converter
1	0	0	Select input channel AD0
1	0	1	Select input channel AD1
1	1	0	Select input channel AD2
1	1	1	Select input channel AD3

NOTE: If ADMOD.2–ADMOD.0 = "0", the analog input channel selection is disabled.

ADC CONTROL REGISTER (AFLAG)

The A/D converter control register, AFLAG, is a 4-bit register that contains the control flags used to start the A/D converter and to monitor its operational status.

To start a conversion operation, you set the ADSTR bit in the AFLAG register to "1". ADSTR is write-only and is 1-bit and 4-bit addressable. You can read the status of the EOC flag (End Of Conversion) at any time to check the current status of the conversion operation. When a conversion is in-progress, EOC is "0". When a conversion is completed, the EOC flagset to "1" and the result can be read. Whenever the ADSTR bit is set to "1", the EOC flag is cleared to "0". This prevents the start of a new conversion operation until the current operation has been completed. The EOC flag is 1-bit or 4-bit read-only addressable.

☞ PROGRAMMING TIP — Configuring A/D Converter Input Pins

In this A/D converter program sample, the ADC0, ADC1 and ADC2 pins are used as A/D input pins and the P5.3/ADC3 is used as normal input pin:

```
        BITR    EMB
        BITR    IEAD        ;  Disable INTAD interrupt
        DI                  ;  Disable all interrupts during A/D conversion
        LD      A,#4H
        LD      ADMOD,A     ;  ADC0 pin select for A/D conversion
        BITS    ADSTR       ;  A/D conversion start
AD0CK   BTSF    EOC         ;  A/D conversion end check
        JR      AD0CK       ;  A/D conversion not completed
        LD      EA,ADATA    ;  A/D conversion end
        LD      ADC0BUF,EA  ;  ADC0BUF ← ADC0 conversion data
        LD      A,#1H
        LD      ADMOD,A     ;  ADC1 pin select for A/D conversion
        BITS    ADSTR       ;  A/D conversion start
AD1CK   BTSF    EOC         ;  A/D conversion end check
        JR      ADC1CK      ;  A/D conversion not completed
        LD      EA,ADATA    ;  AD1 conversion end
        LD      ADC1BUF,EA  ;  ADC1BUF ← ADC1 conversion data
        LD      A,#2H
        LD      ADMOD,A     ;  ADC2 pin select for A/D conversion
        BITS    ADSTR       ;  AD conversion start
AD2CK   BTSF    EOC         ;  AD conversion end check
        JR      AD2CK       ;  AD conversion not completed
        LD      EA,ADATA    ;  AD conversion end
        LD      ADC2BUF,EA  ;  ADC2BUF ← ADC2 conversion data
```

PLL FREQUENCY SYNTHESIZER

The phase locked loop (PLL) frequency synthesizer locks medium frequency (MF), high frequency (HF), and very high frequency (VHF) signals to a fixed frequency using a phase difference comparison system. As shown in Figure 41, the PLL frequency synthesizer consists of an input selection circuit, programmable divider, phase detector, reference frequency generator, and a charge pump.

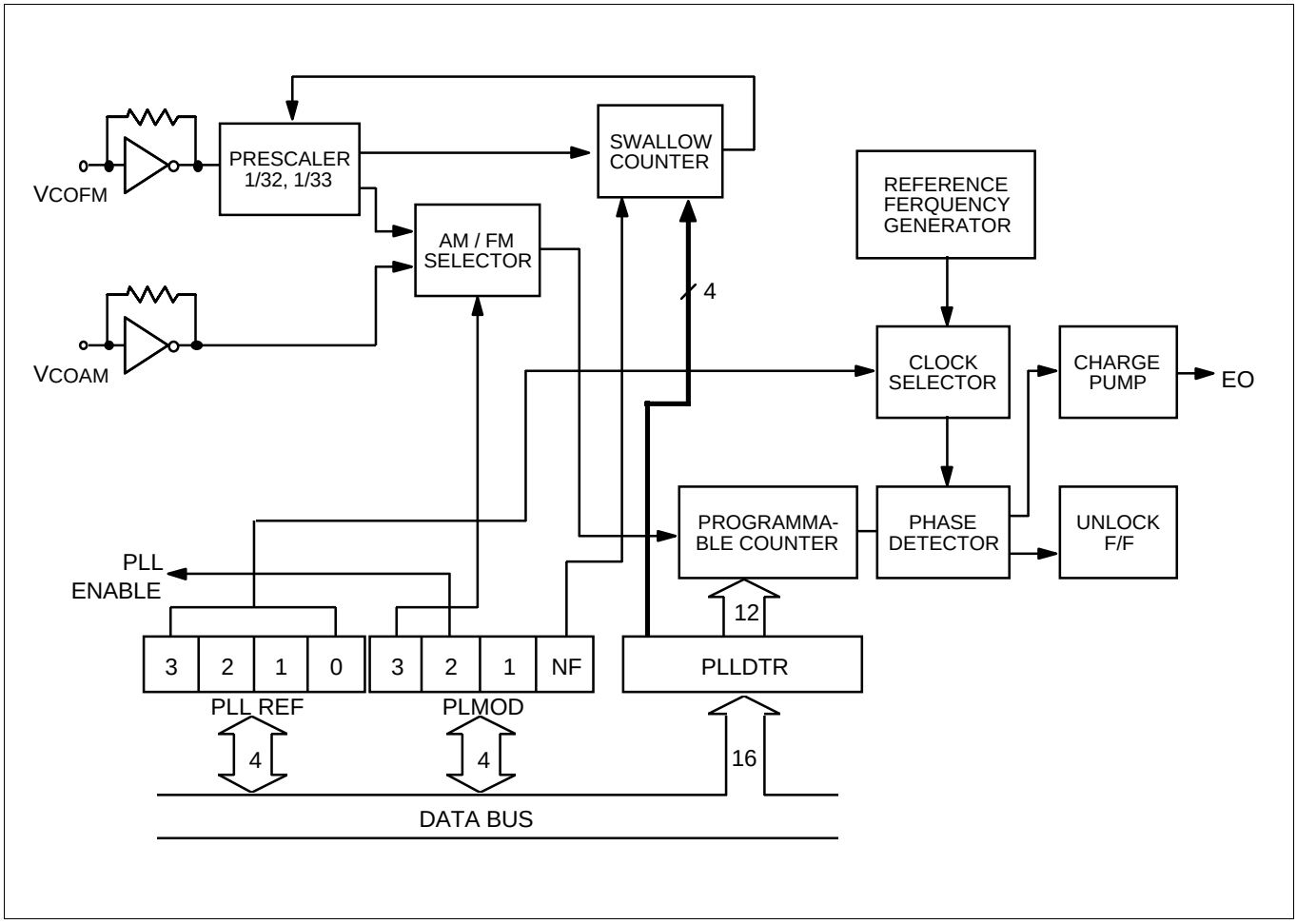


Figure 41. The Block Diagram of the PLL Frequency Synthesizer

PLL FREQUENCY SYNTHESIZER FUNCTIONS

The PLL frequency synthesizer divides the signal frequency at the V_{COAM} or V_{COFM} pin using the programmable divider. It then outputs the phase difference between the divided frequency and reference frequency at the EO pin.

NOTE

The PLL frequency synthesizer operates only when the CE pin is High level. When the CE pin is Low leve, the synthesizer is disabled.

Input Selection Circuit

The input selection circuit consists of the V_{COAM} pin and V_{COFM} pins, an FM/AM selector, and two amplifiers. The input selection circuit selects the frequency division method and the input pin of the PLL frequency synthesizer.

You can choose one of two frequency division methods using the PLL mode register: 1) direct frequency division method, or 2) pulse swallow method. The PLL mode register is also used to select the V_{COAM} or V_{COFM} pin as the frequency input pin.

Programmable Divider

The programmable divider divides the frequency of the signal from the V_{COAM} and V_{COFM} pins in accordance with the values contained in the swallow counter and programmable counter. The programmable divider consists of prescalers, a swallow counter, and a programmable counter.

When the PLL operation starts, the contents of the PLL data registers (PLLD0–PLLD3) and the NF bit in the PLMOD register are automatically loaded into the 12-bit programmable counter and the 5-bit swallow counter.

When the 12-bit programmable down counter reaches zero, the contents of the data register are automatically reloaded into the programmable counter and the swallow counter for the next counting operation.

If you modify the data register value while the PLL is operating, the new values are not immediately loaded into the two counters; the new data are loaded into the two counters when the current count operation has been completed.

The contents of the data register are undetermined after an initial power-on. However, the data register retains its current value when the reset operation is initiated by an external reset or a change in level at the CE pin.

The swallow counter is a 5-bit binary down counter; the programmable counter is a 12-bit binary down counter. The swallow counter is for FM mode only. The swallow counter and programmable counter start counting down simultaneously. When the swallow counter starts counting down, the 1/33 prescaler is selected. When the swallow counter reaches zero, it stops operation and selects the 1/32 prescaler.

PLL DATA REGISTER (PLLD)

The frequency division value of the swallow counter and programmable counter is set in the PLL data register (PLLD0–PLLD3). Figure 48 shows the PLL data register configuration. The PLLD register can be manipulated using 4-bit and 8-bit RAM control instructions.

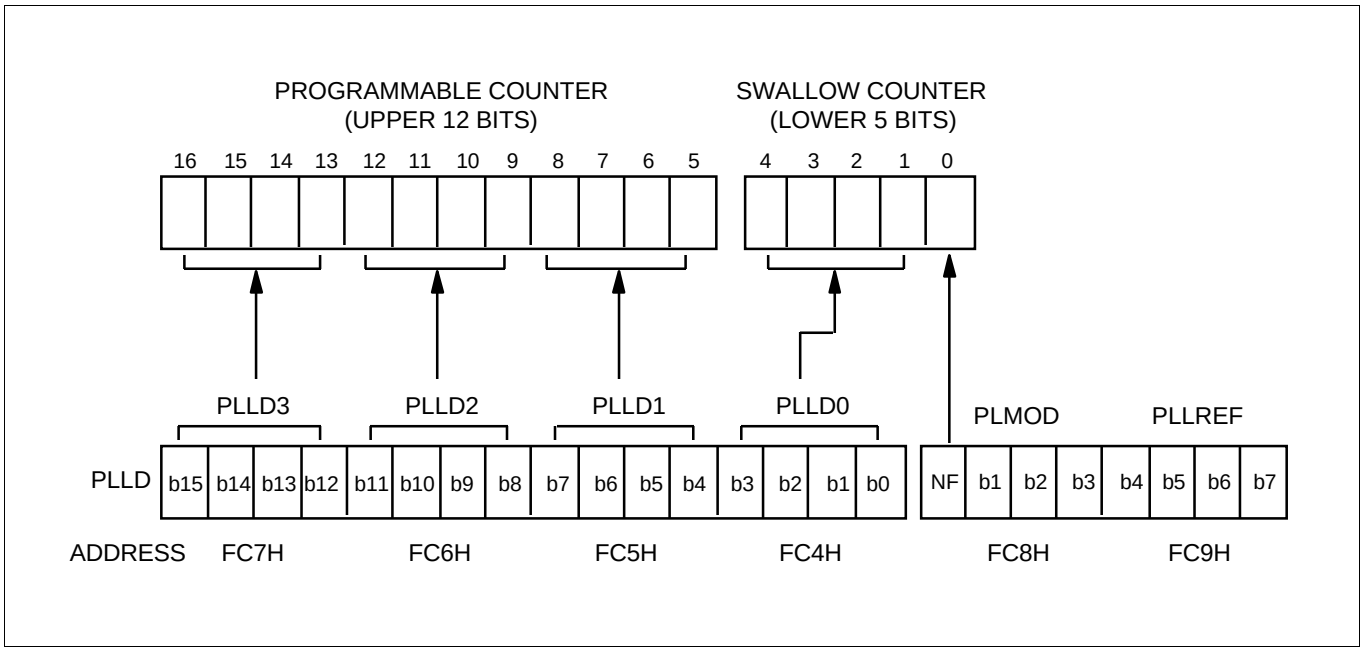


Figure 42. PLLD Register Configuration

Direct Frequency Division and Pulse Swallow Formulas

In the direct frequency division method, the upper 12 bits are valid. In the pulse swallow method, all 16 bits are valid. The upper 12 bits are set in the programmable counter and the lower 4 bits and the NF bit are set in the swallow counter. The frequency division formulas for both methods, as set in the PLL data register, are shown below:

— Direct frequency division (AM) is

$$fr = \frac{fV_{COAM}}{N}$$

where the frequency division value (N) is 12 bits; fV_{COAM} = input frequency at the V_{COAM} pin

— Pulse swallow system (FM) is

$$fr = \frac{fV_{COFM}}{N}$$

where the frequency division value (N) is 16 bits; fV_{COFM} = input frequency at the V_{COFM} pin

REFERENCE FREQUENCY GENERATOR

The reference frequency generator produces reference frequencies which are then compared by the phase comparator. As shown in Figure 43, the reference frequency generator divides a crystal oscillation frequency of 4.5 MHz and generates the reference frequency (fr) for the PLL frequency synthesizer. Using the PLLREF register, you can select from ten different reference frequencies.

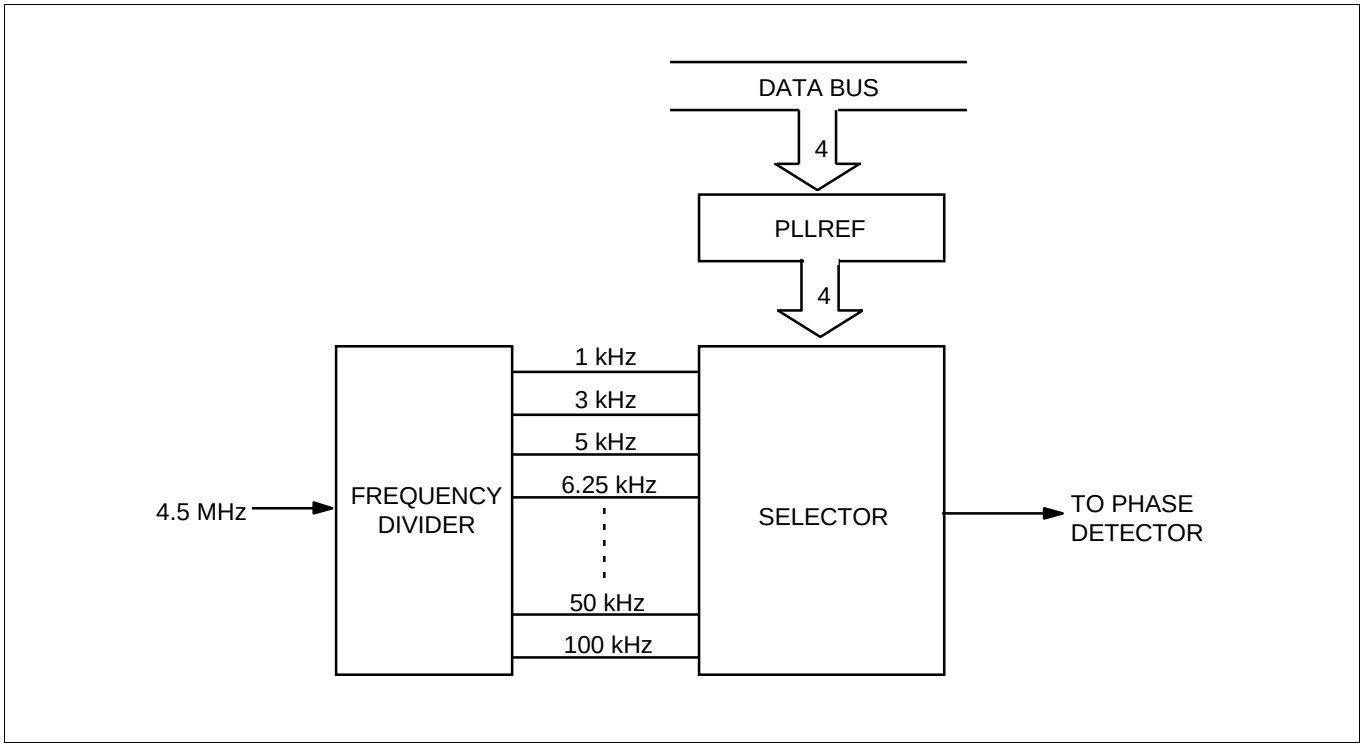


Figure 43. Reference Frequency Generator

PLL MODE REGISTER (PLMOD)

The PLL mode register (PLMOD) is used to start and stop PLL operation. PLMOD values also determine the frequency dividing method.

PLMOD.3 selects the frequency dividing method. The basic configuration for the two frequency dividing methods are as follows:

Direct Method

— Used for AM mode

— Swallow counter is not used

— V_{COAM} pin is selected for input

Pulse Swallow Method

— Used for FM mode

— Swallow counter is used

— V_{COFM} pin is selected for input

The input frequency at the V_{COAM} or V_{COFM} pin is divided by the programmable divider.The frequency division value of the programmable divider is written to the PLL data register. When the pulse swallow method is selected by setting PLMOD.3, the input signal is first divided by a1/32 or 1/33 prescaler and the divided frequency is input to the programmable divider. PLMOD can be written by 4-bit RAM control instruction. Table 33 shows PLMOD organization.

Table 30. PLMOD Organization

PLL Enable Bit

PLMOD.2	0	PLL disable
	1	PLL enable

Frequency Division Method Selection Bit

PLMOD.3	Frequency Division Method	Selected Pin	Input Voltage	Input Frequency	Division Value
0	Direct method for AM	V_{COAM} selected; V_{COFM} pulled down	300 mV_{PP}	0.5 – 30 MHz	16 to (2^{12} – 1)
1	Pulse swallow method for FM	V_{COFM} selected; V_{COAM} pulled down	300 mV_{PP}	15 – 150 MHz	2^{10} to (2^{17} – 2)

NOTE: The NF bit, a one-bit frequency division value, is written to bit 0 in the swallow counter.

PLL REFERENCE FREQUENCY SELECTION REGISTER (PLLREF)

The PLL reference frequency selection register (PLLREF) used to determine the reference frequency. You can select one of ten reference frequencies by setting bits PLLREF.3–PLLREF.0 to the appropriate value.

You can select one of ten reference frequencies by setting bits PLLREF.3–PLLREF.0.

Table 31. PLLREF Register Organization

PLLREF.3	PLLREF.2	PLLREF.1	PLLREF.0	Reference frequency selection
0	0	0	0	Select 1 kHz as reference frequency
0	0	0	1	Select 3 kHz as reference frequency
0	0	1	0	Select 5 kHz as reference frequency
0	0	1	1	Select 6.25 kHz as reference frequency
0	1	0	0	Select 9 kHz as reference frequency
0	1	0	1	Select 10 kHz as reference frequency
0	1	1	0	Select 12.5 kHz as reference frequency
0	1	1	1	Select 25 kHz as reference frequency
1	0	0	0	Select 50 kHz as reference frequency
1	0	0	1	Select 100 kHz as reference frequency

PHASE DETECTOR, CHARGE PUMP, AND UNLOCK DETECTOR

The phase comparator compares the phase difference between divided frequency (f_N) output from the programmable divider and the reference frequency (fr) output from the reference frequency generator.

The charge pump outputs the phase comparator's output from error output pins EO. The relation between the error output pin output, divided frequency f_N, and reference frequency f_R is shown below:

fr	>	f_N	=	Low level output
fr	<	f_N	=	High level output
fr	=	f_N	=	Floating level

A PLL operation starts when a value is loaded to the PLMOD register. The PLL unlock flag (ULFG) in the PLL flag register, PLLREG, provides status information regarding the reference frequency and divided frequency.

The unlock detector detects the unlock state of the PLL frequency synthesizer. The unlock flag in the PLLREG register is set to "1" in an unlocked state. When ULFG = "0", the PLL locked state is selected.

PLLREG	**ULFG**	**CEFG**	IFCFG	0	F9DH

The ULFG flag is set continuously at a period of reference frequency f_R by the unlock detector. You must therefore read the ULFG flag in the PLLREG register at periods longer than $1/f_R$ of the reference frequency. ULFG is reset whenever it is read. The PLLREG register can be read using 1-bit or 4-bit RAM control register instructions.

PLL operation is controlled by the state of the CE (chip enable) pin. The PLL frequency synthesizer is disabled and the error output pin is set to floating state whenever the CE pin is Low. When CE pin is hHgh level, the PLL operates nomally.

The chip enable flag in the PLLREG register, CEFC, provides the status of the current level of the CE pin. Whenever the state of the CE pin goes from Low to High, the CEFG flag is set to "1" and a CE reset operation occurs. When the CE pin goes from High to Low, the CEFG flag is cleared to "0" and a CE interrupt is generated.

USING THE PLL FREQUENCY SYNTHESIZER

This section describes the steps you should follow when using the PLL direct frequency division method and the pulse swallow method. In each case, you must make the following selections in this order:

1. Frequency division method: Direct frequency division (AM) or pulse swallow (FM)
2. Output pin: V_{COAM} or V_{COFM}
3. Reference frequency: f_R
4. Frequency division value: N

Direct Frequency Division Method

1. Select the direct frequency division method by writing a "0" to PLMOD.3.
2. The V_{COAM} pin is configured for output when you select the direct frequency division method.
3. Select the reference frequency by writing the appropriate values to the PLMOD register.
4. The frequency division value is

$$N = \frac{fV_{COAM}}{f_R}$$

where fV_{COAM} is the input frequency at the V_{COAM} pin, and f_R is the reference frequency.

Example:

The following data are used to receive an AM-band broadcasting station:

Receive frequency:1422 kHz

Reference freqency: 9 kHz

Intermediate frequency: + 450 kHz

The frequency division value N is calculated as follows:

$$N = \frac{fV_{COAM}}{fr} = \frac{1422 + 450}{9} = 208 \text{ (decimal)}$$

= 0D0H (hexadecimal)

You would modify the PLL data register and PLMOD register as follows:

PLLD3	PLLD2	PLLD1	PLLD0	PLMOD	NF
0 0 0 0	1 1 0 1	0 0 0 0	X X X X	0 1 0	X

NOTE: In the direct method, the contents of PLDTR0 and NF are not evaluated.

Pulse Swallow Method

1. Select the pulse swallow method by writing a "1" to PLMOD.3.
2. The V_{COFM} pin is configured for output when you select the pulse swallow method.
3. Select the reference frequency by writing the appropriate values to the PLMOD register.
4. Calculate the frequency division value as follows:

$$N = \frac{fV_{COFM}}{f_R}$$

Where fV_{COFM} is the input frequency at the V_{COFM} pin, and f_R is the reference frequency

Example:

The following data are used to receive an FM-band broadcasting station:

Receive frequency: 100.0 MHz

Reference frequency: 25 kHz

Intermediate frequency: 10.7 MHz

The frequency division value N is calculated as follows:

$$N = \frac{fV_{COFM}}{f_R} = \frac{(100.0 + 10.7) \infty 10^{6}}{25 \infty 10^{3}}$$

= 4428 (decimal)
= 114CH (hexadecimal)

You would modify the PLL data register and PLMOD register as follows:

PLLD3	PLLD2	PLLD1	PLLD0	NF	PLMOD	NF
0 0 0 0	1 0 0 0	1 0 1 0	0 1 1 0	0	1 1 0	0
1	1	4	C		PLL Enable	
					Pulse Swallow Method	

In the above example, each time NF bit value (LSB) is inverted, the V_{CO} oscillation frequency varies by 25 kHz. To simplify programming, it is therefore better not to use the NF bit.

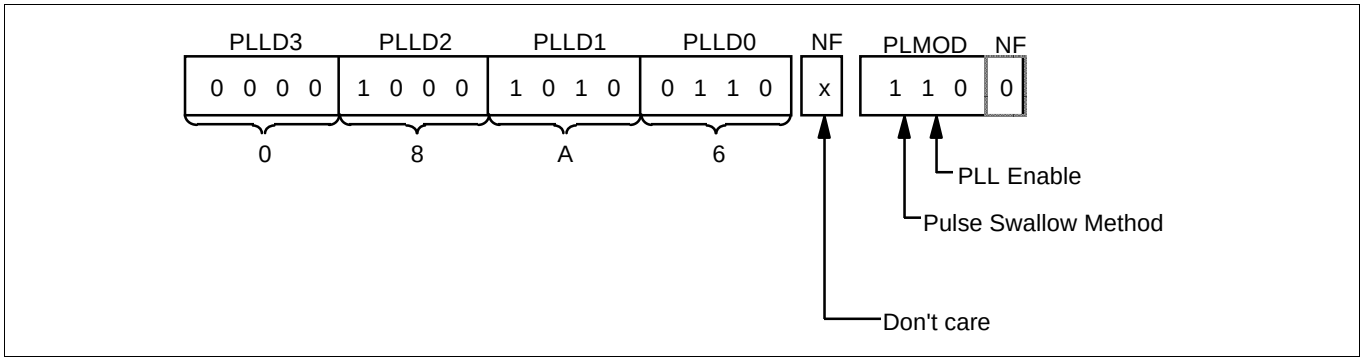
In the next example, the reference frequency is calculated in multiples of 25 kHz and the NF bit is not used.

Example:

$$N = \frac{fV_{COFM}}{fr}$$

$$= \frac{(100.0 + 10.7) \infty 10^6}{2 \infty 25 \infty 10^3}$$

= 2214 (decimal)
= 8A6H (hexadecimal)



As these examples show, all 16 bits (the 16 PLLD bits except the NF bit) are used for the pulse swallowing method. When the direct method is used, only the most significant 12 bits of the PLLD value (PLLD3, PLLD2, and PLLD1) are evaluated.

INTERMEDIATE FREQUENCY COUNTER

The KS57C3108 uses an intermediate frequency counter (IFC) to count the frequency of the AM or FM signal at FMIF or AMIF pin. The IFC block consists of a 1/2 divider, gate control circuit, IFC mode register (IFMOD) and a 16-bit binary counter. The gate control circuit, which controls the frequency counting time, is programmed using the IFMOD register. Four different gate times can be selected using IFMOD register settings.

During gate time, the 16-bit IFC counts the input frequency at the FMIF or AMIF pins. The FMIF or AMIF pin input signal for the 16-bit counter is selected by IFMOD register.

The 16-bit binary counter (IFCNT1–IFCNT0) can be read by 8-bit RAM control instructions only. When the FMIF pin input signal is selected, the signal is divided by two. When the AMIF pin input signal is directly connected to the IFC, it is not divided.

By setting IFMOD register, the gate is opened for 1-ms, 4-ms, or 8-ms periods. During the open period of the gate, input frequency is counted by the 16-bit counter. When the gate is closed, the counting operation is complete, and an interrupt is generated.

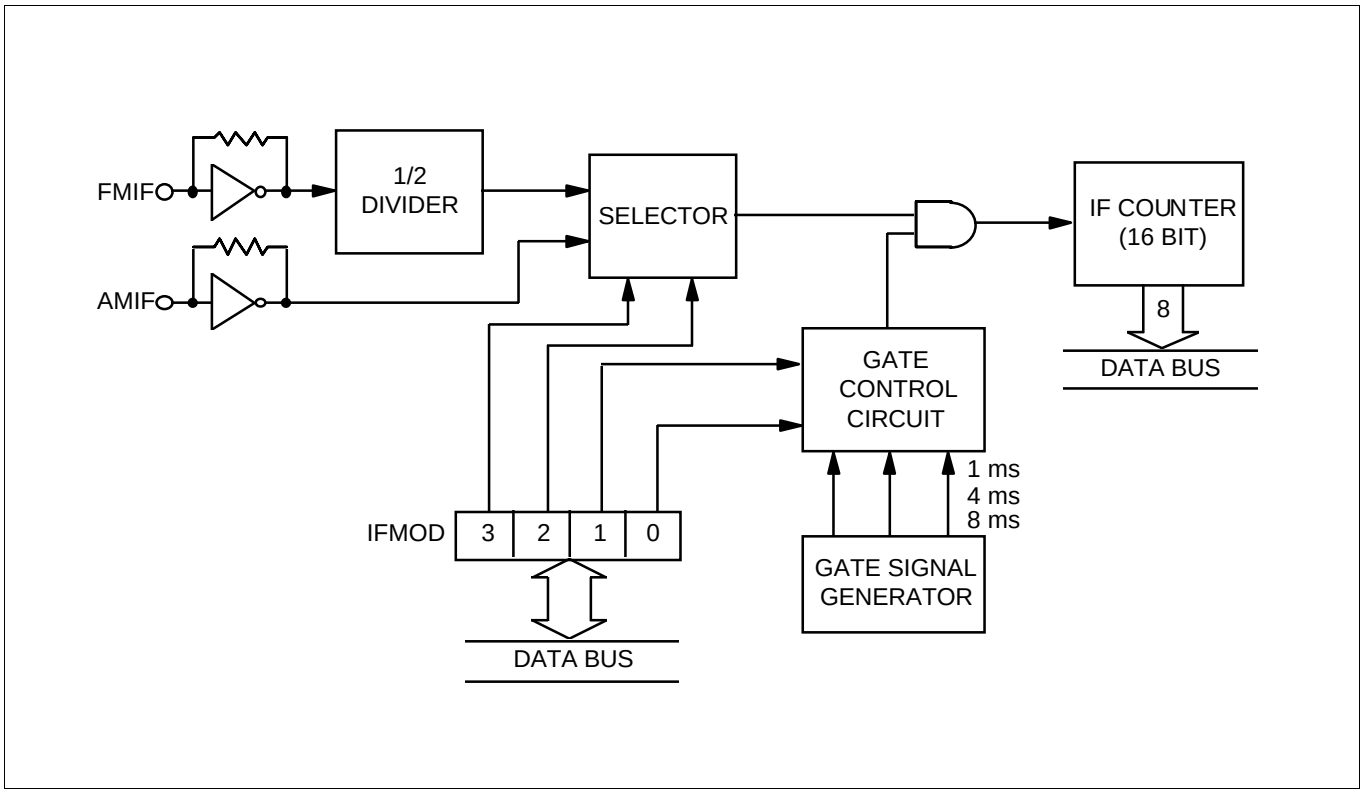


Figure 44. IF Counter Block Diagram

IFC MODE REGISTER (IFMOD)

The IFC mode register (IFMOD) is a 4-bit register that is used to select the input pin and gate time. Setting IFMOD register reset IFC value and IFC gate flag value, and starts IFC operation. You use the IFMOD register to select the AMIF or FMIF input pin and the gate time.

IFC operation starts when you select AMIF or FMIF as the IFC input pin. The IFMOD register can be read or written by 4-bit RAM control instructions. A reset operation clears all IFMOD values to "0"..

Table 32. IFMOD Organization

Pin Selection Bits

IFMOD.3	IFMOD.2	Effect of Control Setting
0	0	Disable IFC
0	1	Enable IFC operation; select AMIF pin
1	0	Enable IFC operation; select FMIF pin
1	1	Enable IFC operation; select both AMIF and FMIF pins

Gate Time Selection Bits

IFMOD.1	IFMOD.0	Selected Gate Time
0	0	1 millisecond
0	1	4 milliseconds
1	0	8 milliseconds
1	1	Gate remains open

PLL FLAG REGISTER (PLLREG)

The PLL flag register (PLLREG) is a 4-bit read-only register.

When IFC operation is started by setting IFMOD, the IFC gate flag (IFCFG) is cleared to "0". After a specified gate time has elapsed, the IFCFG bit is automatically set to "1". This lets you check whether a IFC counting operation has been completed or not.

The IFC interrupt can also be used to check whether or not a IFC counting operation is complete. The reset value of IFCFG is "0".

GATE TIME

When you write a value to IFMOD, the IFC gate is opened for a 1-millisecond, 4-millisecond, or 8-millisecond interval, starting with a rising clock edge. When the gate is open, the frequency at the AMIF or FMIF pin is counted by the 16-bit counter. When the gate closes, the IFC gate flag (IFCFG) is set to "1".

An interrupt is then generated and the IFC interrupt request flag (IRQIF) is set.

Figure 51 shows gate timings with a 1-kHz internal clock.

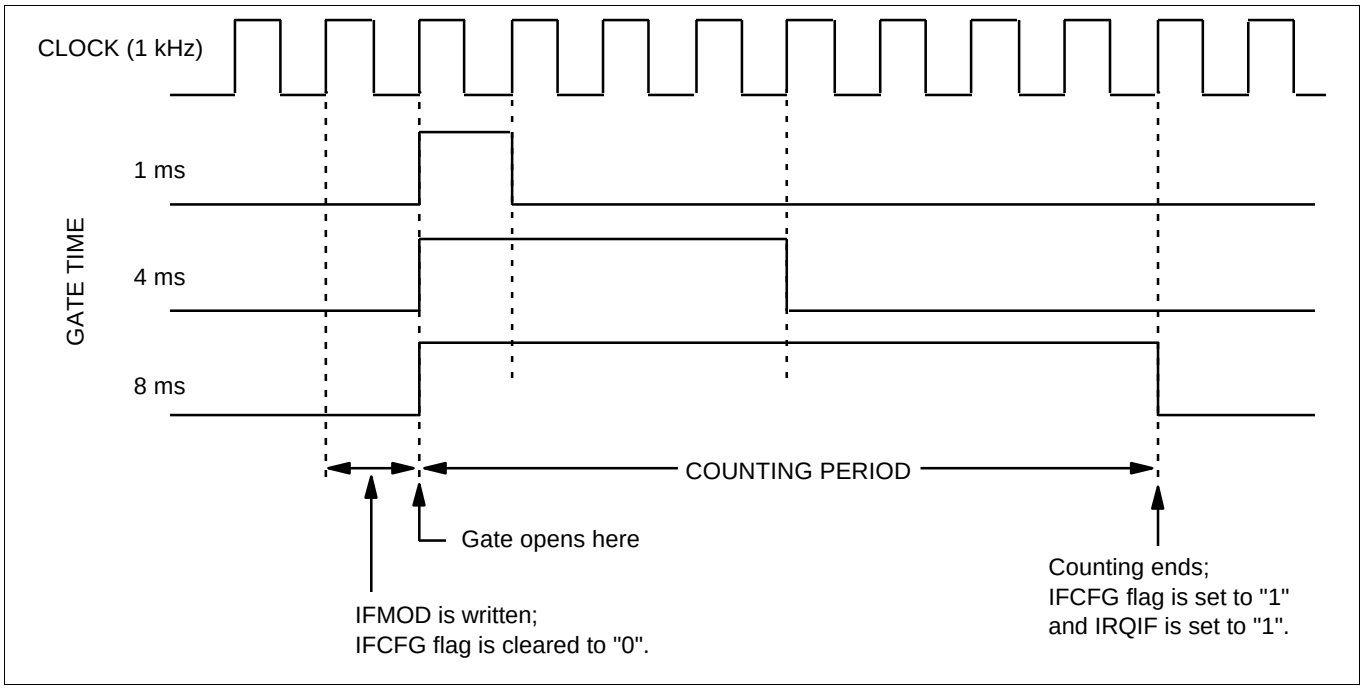


Figure 45. Gate Timing (1, 4, or 8 ms)

Selecting "Gate Remains Open"

If you select "gate remains open" (IFMOD.0 and IFMOD.1 = "1"), the IFC counts the input signal during the open period of the gate. The gate closes the next time a value is written to IFMOD.

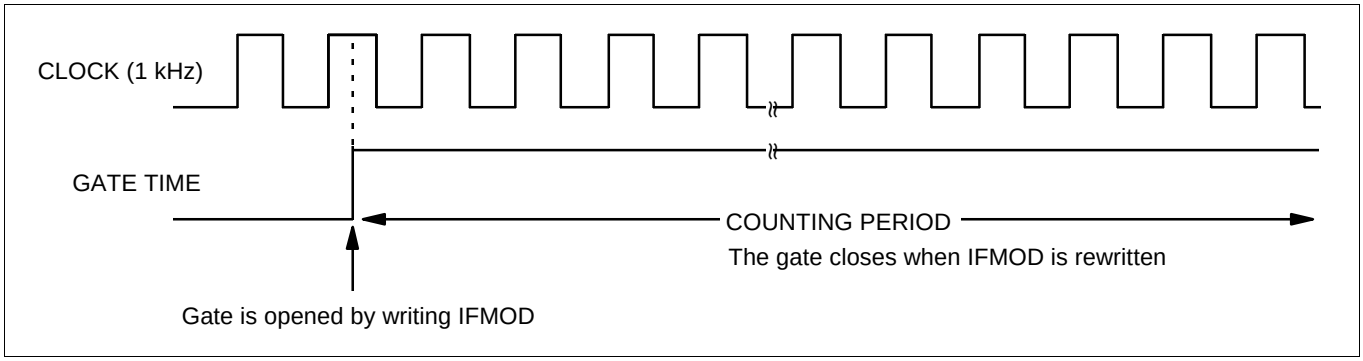
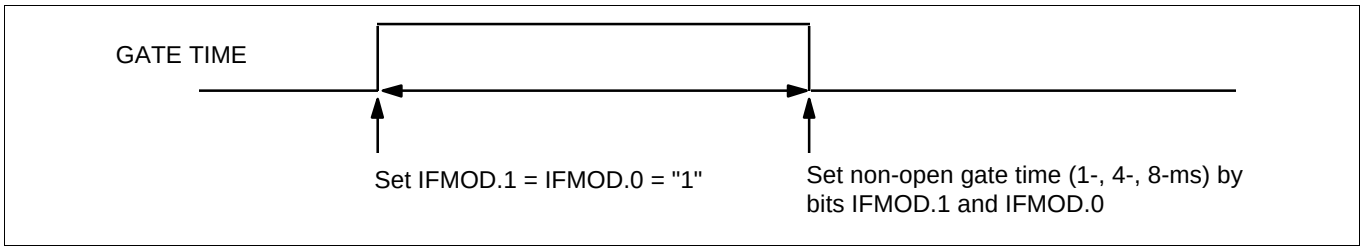


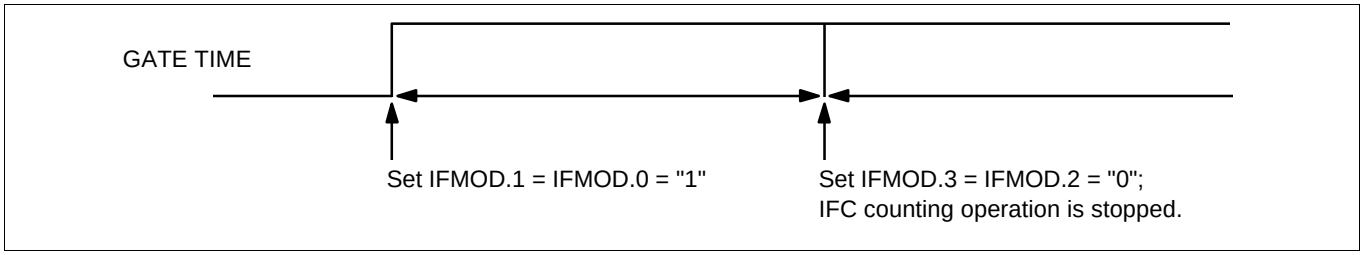
Figure 46. Gate Timing (When Open)

When you select "gate remains open" as the gating time, you can control the opening and closing of the gate in one of two ways:

— Set the gate time to a specific interval (1-ms, 4-ms, or 8-ms) by setting bits IFMOD.1 and IFMOD.0.



— Disable IFC operation by clearing bits IFMOD.3 and IFMOD.2 to "0". This method lets the gate remain open, and stops the counting operation.



Gate Time Errors

A gate time error occurs whenever the gate signals are not synchronized to the internal instruction clock. That is, the IFC does not start counter operation until a rising edge of the gate signal is detected, even though the counter start instruction (setting bits IFMOD.3 and IFMOD.2) has been executed.

Therefore, there is a maximum 1-ms timing error (see Figure 53).

After you have executed the IFC start instruction, you can check the gate state at any time. Please note, however that the IFC does not actually start its counting operation until the stabilization time for the gate control signal has elapsed.

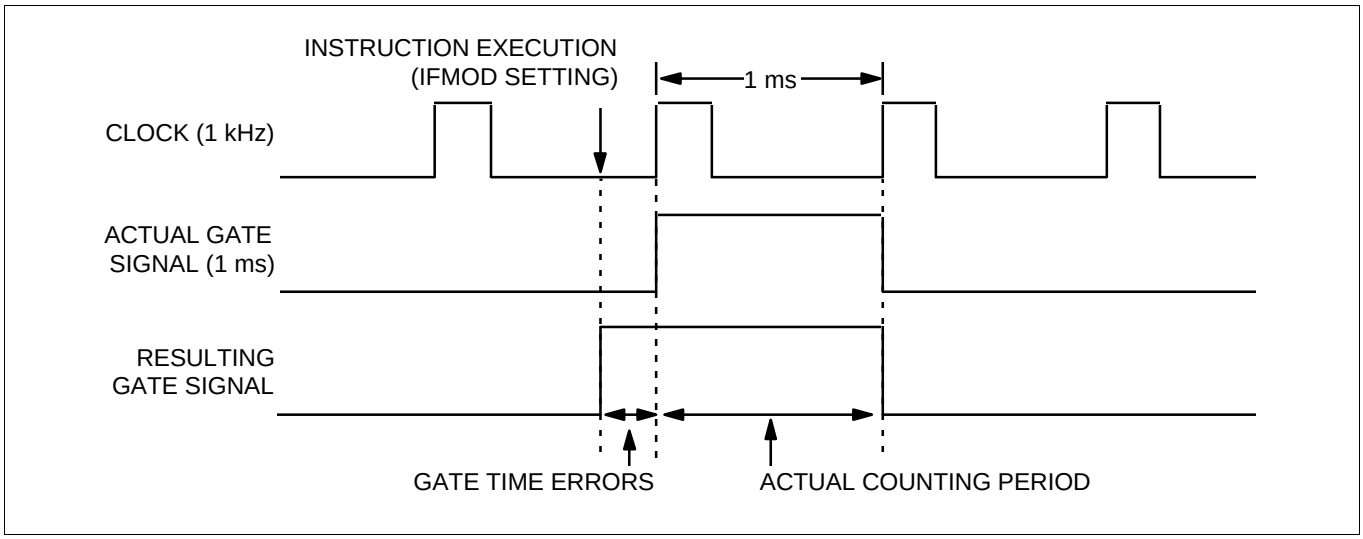


Figure 47. Resulting Gate Timing

Counting Errors

The IF counter counts the rising edges of the input signal to determine the frequency. If the input signal is high level when the gate is open, one additional pulse is counted. When the gate is close, however, counting is not affected by the input signal status. In other words, the counting error is "+1, 0".

IF COUNTER (IFC) OPERATION

IFMOD register bits 2 and 3 are used to select the input pin and to start or stop IFC counting operation.

You stop the counting operation by clearing IFMOD.2 and IFMOD.3 to "0". The IFC retains its previous value until IFMOD register values are specified.

Setting bits IFMOD.3 and IFMOD.2 starts the frequency counting operation. Counting continues as long as the gate is open. The 16-bit counter value is automatically cleared to 0000H after it overflows (at FFFFH), and continues counting from zero. The 16-bit count value (IFCNT1–IFCNT0) can be read by 8-bit RAM control instructions. A reset operation clears the counter to zero.

IFCNT0	IFCNT0.7	IFCNT0.6	IFCNT0.5	IFCNT0.4	IFCNT0.3	IFCNT0.2	IFCNT0.1	IFCNT0.0
IFCNT1	IFCNT1.7	IFCNT1.6	IFCNT1.5	IFCNT1.4	IFCNT1.3	IFCNT1.2	IFCNT1.1	IFCNT1.0

When the specified gate open time has elapsed, the gate closes in order to complete the counter operation. At this time, the IFC interrupt request flag (IRQIF) is automatically set to "1" and an interrupt is generated. The IRQIF flag is automatically cleared to "0" when the interrupt is serviced. The IFC gate flag (IFCFG) is set to "1" at the same time the gate is closed. Since the IFCFG flag is cleared to "0" when IFC operation start, you can check the IFCFG flag to determine when IFC operation stops (that is, when the specified gate open time has elapsed).

The frequency applied to FMIF or AMIF pin is counted while the gate is open. The frequency applied to FMIF pin is divided by 2 before counting. The relationship between the count value (N) and input frequencies f_{AMIF} and f_{FMIF} is shown below.

— FMIF pin input frequency is

$$f_{FMIF} = \frac{N\ (DEC)}{T_G} \times 2$$

when T_G = gate time (1 ms, 4 ms, 8 ms)

— AMIF pin input frequency is

$$f_{AMIF} = \frac{N\ (DEC)}{T_G}$$

when T_G = gate time (1 ms, 4 ms, 8 ms)

Table 36 shows the range of frequencies that you can apply to the AMIF and FMIF pins.

Table 33. IF Counter Frequency Characteristics

Pin	Voltage Level	Frequency Range
AMIF	300 m Vpp (min)	0.1 MHz to 1 MHz
FMIF	300 m Vpp (min)	5 MHz to 15 MHz

INPUT PIN CONFIGURATION

The AMIF and FMIF pins have built-in AC amplifiers (see Figure 54). The DC component of the input signal must be stripped off by the external capacitor.

When the AMIF or FMIF pin is selected for the IFC function and the switch is turned on voltage of each pin increases to approximately 1/2 V_{DD} after sufficiently long time. If the pin voltage does not increase to approximately 1/2 V_{DD}, the AC amplifier exceeds its operating range, possibly causing an IFC malfunction. To prevent this from occurring, you should program a sufficiently long time delay interval before starting the count operation.

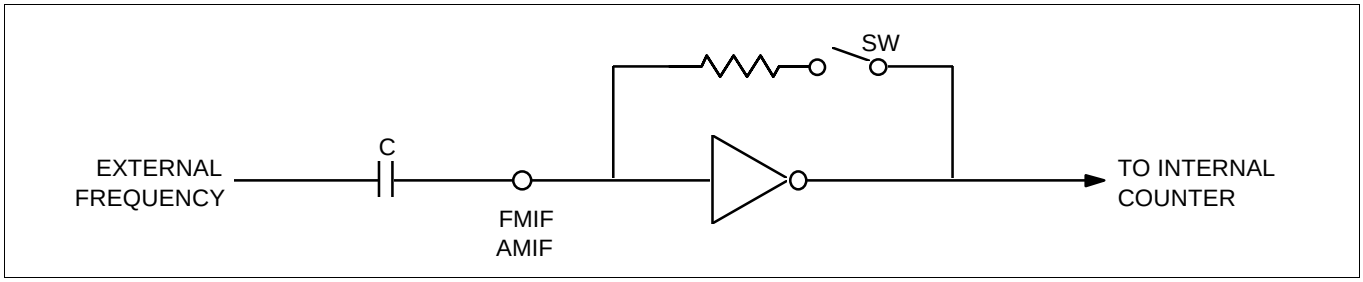


Figure 48. AMIF and FMIF Pin Configuration

☞ PROGRAMMING TIP — Counting the Frequency at the FMIF pin (8-ms Gate Time)

You must insert a time delay before starting an IF counter operation. This time delay ensures the normal operation of the built-in AC amplifier when each pin is selected as a IFC input pin.

```
        SMB     15
        (Time delay) ;          Built-in AC amplifier stabilization time
        LD      A,#0AH          ;
        LD      IFMOD,A         ;  FMIF pin is selected and gate time is set to 8 ms
                                ;  Start IFC operation
LOOP    BTSF    IFCG            ;  Check gate open and close
        JPS     READ            ;  Jump to READ if gate closes
        •
        •
        JPS     LOOP
READ    (Read IFCNT1, IFCNT0)
```

IFC DATA CALCULATION

Selecting the FMIF Pin for IFC Input

First, divide the signal at the FMIF pin by 2, and then apply this value to the IF counter. This means that the IF counter value is equal to one-half of the input signal fre quency.

FMIF input frequency (f_{FMIF}): 10.7 MHz
Gate time (T_G): 8 ms
IFC counter value (N):

$N = (f_{FMIF} / 2) \times T_G = 10.7 \times 10^6 / 2 \times 8 \times 0^{-3} = 42800 = A730H$

Bin	1	0	1	0	0	1	1	1	0	0	1	1	0	0	0	0
Dec	A				7				3				0			
IFCNT	IFCNT1								IFCNT0							

Selecting the AMIF Pin for IFC Input

The signal at AMIF pin is directly input to the IF counter.

AMIF input frequency (f_{AMIF}):450 kHz

Gate time (T_G): 8 ms

IFC counter value (N):

$N = (f_{AMIF}) \times T_G$

$= 450 \times 10^3 \times 4 \times 10^{-3}$

$= 3600$

$= E10H$

Bin	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0
Dec	0				E				1				0			
IFCNT	IFCNT1								IFCNT0							

LCD CONTROLLER/DRIVER

The KS57C3108 microcontroller can directly drive an up-to-14-digit LCD panel. Its LCD block has the following components:

— LCD controller/driver

— Display RAM (1E4H–1FFH) for storing display data

— 28 segment output pins (SEG0–SEG27)

— Four common output pins (COM0–COM3)

— Three LCD operating power supply pins (V_{LC0}–V_{LC2})

— Bias pin for controlling the driver and bias voltage

The frame frequency, duty and bias, and the segment pins used for display output, are determined by bit settings in the LCD mode register, LMOD.

The LCD control register, LCON, is used to turn the LCD display on and off, to switch current to the dividing resistors for the LCD display. Data written to the LCD display RAM can be transferred to the segment signal pins automatically without program control.

When a subsystem clock is selected as the LCD clock source, the LCD display is enabled even during Stop1 mode and Idle mode.

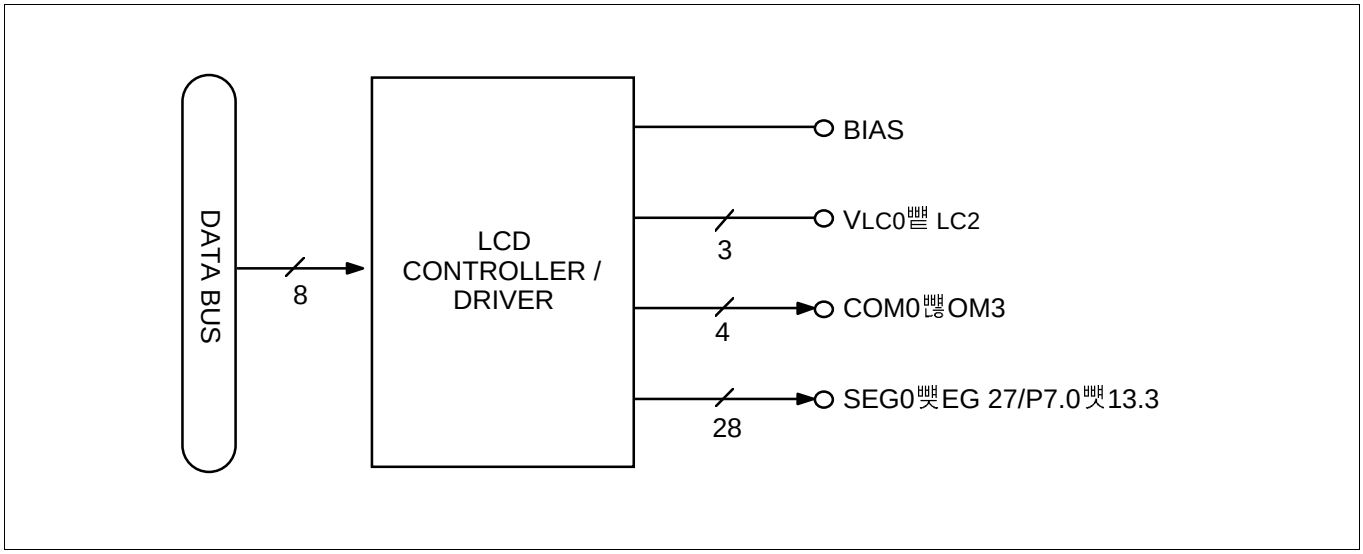


Figure 49. LCD Function Diagram

LCD CIRCUIT DIAGRAM

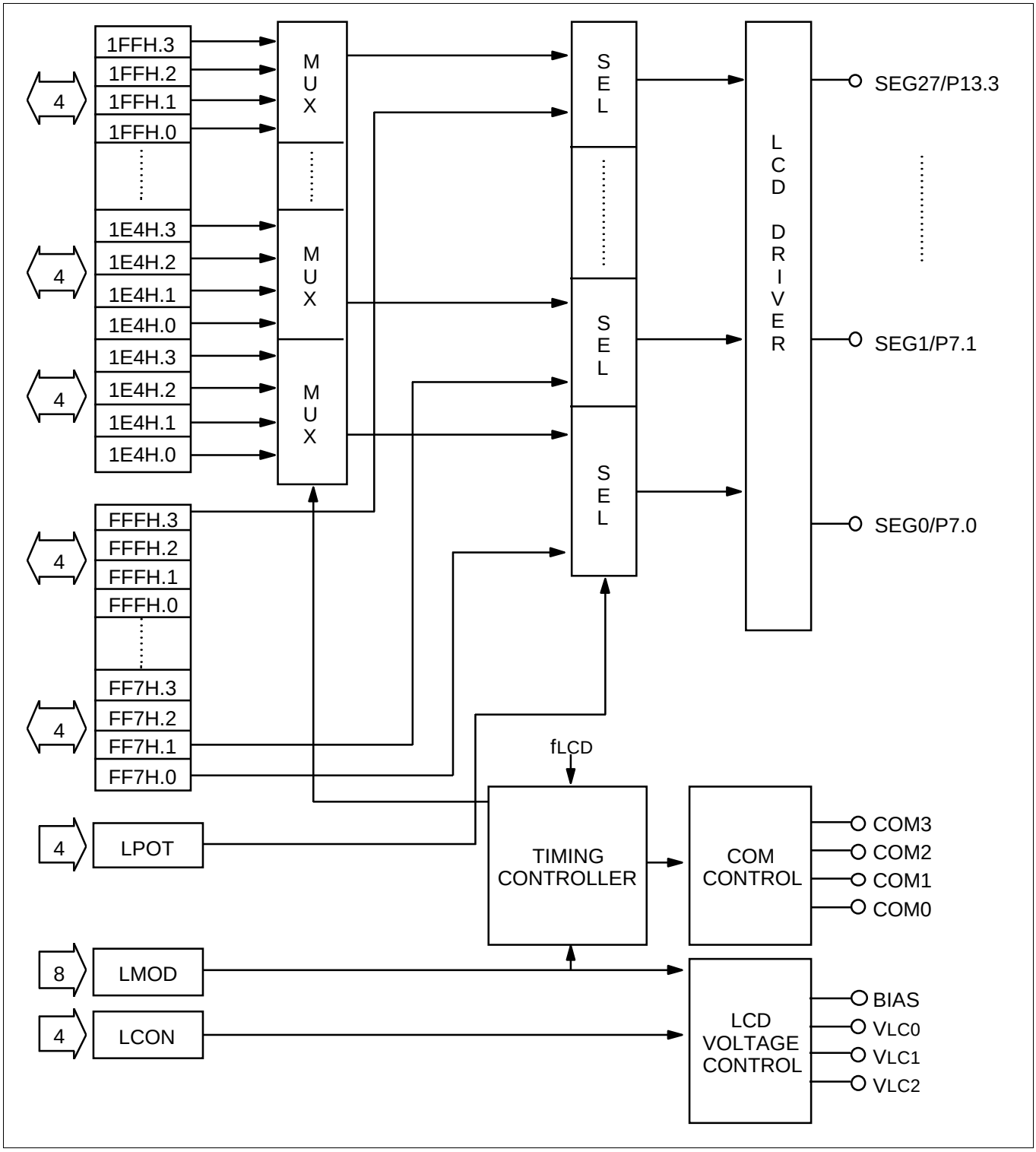


Figure 50. LCD Circuit Diagram

LCD RAM ADDRESS AREA

RAM addresses 1E4H–1FFH are used as LCD data memory. These locations can be addressed by 1-bit or 4-bit instructions. When the bit value of a display segment is "1", the LCD display is turned on; when the bit value is "0", the display is turned off.

Display RAM data is output through segment pins SEG0–SEG27 using a direct memory access (DMA) method that is synchronized with the f_{LCD} signal. RAM addresses in this location that are not used for LCD display can be allocated to general-purpose use.

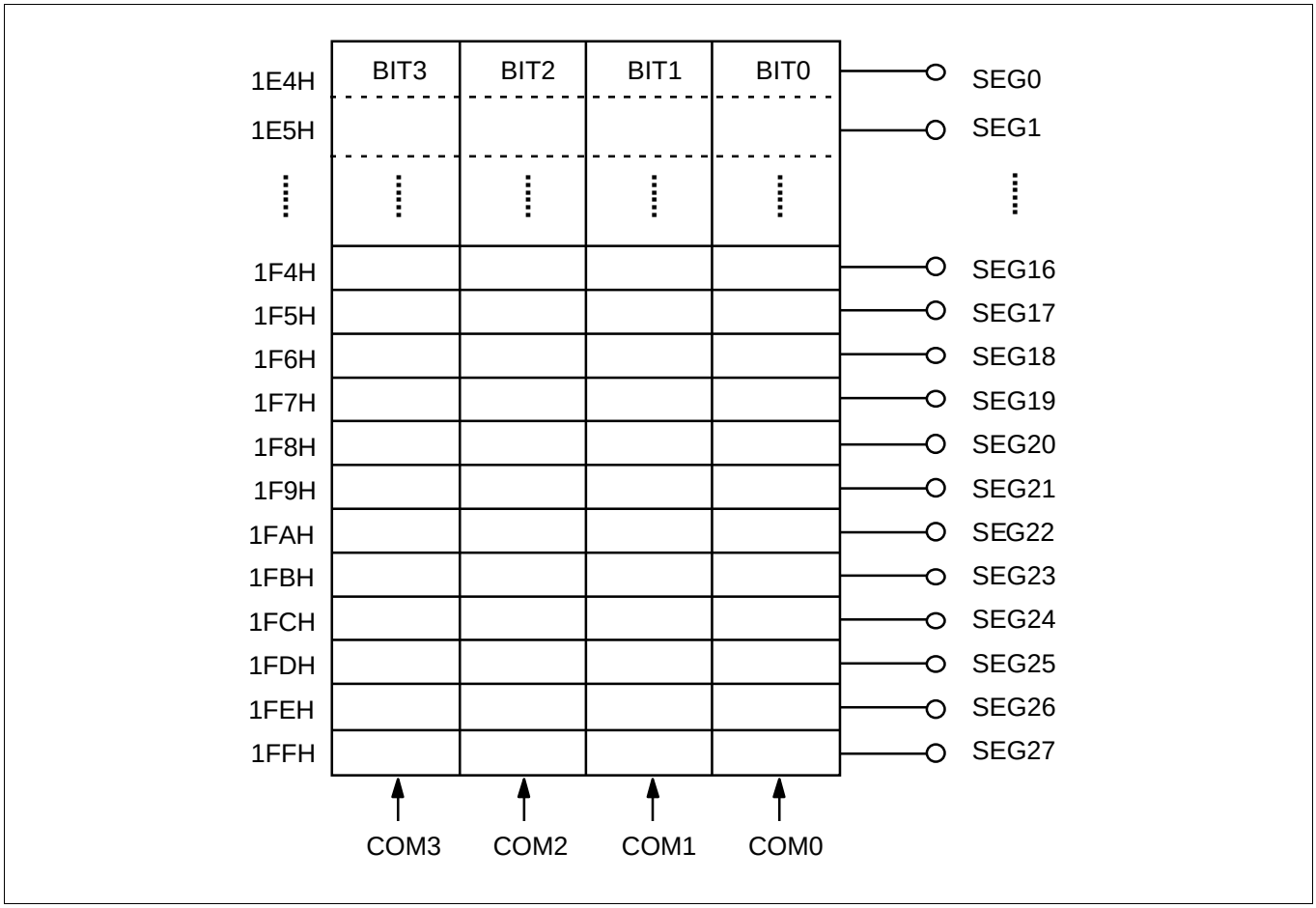


Figure 51. LCD Display Data RAM Organization

Table 34. Display RAM Bits and Sync Clock per Duty Cycle

LCD Duty Cycle	Display RAM Bit(s) Used (1E4H–1FFH)	Display Synchronization Clock (f_{LCD})
Static	Bit 0	COM0
1/2	Bit 0–Bit 1	COM0–COM1
1/3	Bit 0–Bit 2	COM0–COM2
1/4	Bit 0–Bit 3	COM0–COM3

LCD CONTROL REGISTER (LCON)

LCON is a 4-bit write-only register. You use the LCON register to turn the LCD display on and off, and to control the flow of current to dividing resistors in the LCD circuit. A reset operation clears all LCON values to "0". This turns the LCD display off and stops the flow of current to the dividing resistors and LCON.1 is used for P6.

— When LCON.1 is "1", P6 can input/output data.

— When LCON.1 is "0", data line is cut off so P6 can net output/input data.

The reason this mode exist is that if you use segment/output pont as key SCAN output, you can activate P6 when key check time only and that will enchance your LCD quality. The effect of the LCON.0 setting depends on the current value of LMOD.3.

— When LCON.0 is "1" and LMOD.3 is "0", the LCD display is turned off.

— When LCON.0 is "1" and LMOD.3 is "1", the LCD display is turned on and the COM and SEG signal outputs operate in normal display mode.

When LCON.0 is "0", the LCD display is turned off and the current to the dividing resistors is cut off, regardless of the current LMOD.3 value.

Table 35. LCD Control Register (LCON) Organization

LCON Bit	Setting	Description
LCON.3	0	This bit is used for internal testing only and should remain "0"
LCON.2	0	Always "0"
LCON.1	0	Port 6 input/ output enable
	1	Port 6 input/ output disable
LCON.0	0	LCD output Low; cut off current to dividing resistors
	1	When LMOD.3 = "0": LCD output Low; turn display off; When LMOD.3 = "1": COM and SEG output in display mode

Table 36. Relationship of LCON.0 and LMOD.3 Bit Settings

LCON.0	LMOD.3	COM0–COM3	SEG0–SEG27	P7.0–P13.3
0	x	Output Low; turn LCD display off	Output Low; turn LCD display off	Cut off current to dividing resistors
1	0	Output Low; turn LCD display off	Output Low; turn LCD display off	Output latch contents; LCD display is off
	1	COM output corresponds to display mode	SEG output corresponds to display mode	Output latch contents; LCD display is on

NOTE: 'X' means 'don't care.'

LCD MODE REGISTER (LMOD)

The LCD mode register LMOD can be manipulated using 8-bit write instructions. Bit 3 (LMOD.3) can be also written by 1-bit instructions.

LMOD controls the following LCD functions:

— LCD enable/disable (LMOD.3)
— Duty and bias selection (LMOD.3–LMOD.0)
— LCDCK clock frequency selection (LMOD.5–LMOD.4)
— Internal LCD voltage dividing register enable/disable (LMOD.7)

The LCD clock signal, LCDCK, determines the frequency of COM signal scanning of each segment output. This is also referred to as the 'frame frequency'. Because LCDCK is generated by dividing the watch timer clock (fw), the watch timer must be enabled whenever you turn on the LCD display. A reset operation clears the LMOD register values to "0". This produces the following LCD control settings:

— Display is turned off
— LCDCK frequency is the watch timer clock (fw)/2^9 = 64 Hz

If you use a subsystem clock as the watch timer source, the LCD display can continue to operate during Idle mode and Stop1 mode .

The LCD port control register,LPOT, controls the output mode of the 28 pins used for normal outputs (P7.0–P13.3). The value of LMOD.6 is always "0". The LCD output voltage level is determined by the power supply voltage.

Table 37. LCD Clock Signal (LCDCK) Frame Frequency

LCDCK Frequency	Static	1/2 Duty	1/3 Duty	1/4 Duty
fw/2^9 (64 Hz)	64	32	21	16
fw/2^8 (128 Hz)	128	64	43	32
fw/2^7 (256 Hz)	256	128	85	64
fw/2^6 (512 Hz)	512	256	171	128

NOTES:
1. 'fw' is the watch timer clock frequency of 32.768 kHz.
2. The watch timer clock frequency for LCDCK is shown in parentheses in column one.

Table 38. LCD Mode Control Register (LMOD) Organization

LMOD.7	LCD Voltage Dividing Register Control Bit
0	Internal voltage dividing register
1	External voltage dividing register
LMOD.6	**LCD Output Segments and 1-Bit Output Pins**
0	Always "0"

LMOD.5	LMOD.4	LCD Clock (LCDCK) Frequency
0	0	32.768 kHz watch timer clock (fw)/2^9 = 64 Hz
0	1	fw/2^8 = 128 Hz
1	0	fw/2^7 = 256 Hz
1	1	fw/2^6 = 512 Hz

LMOD.3	LMOD.2	LMOD.1	LMOD.0	Duty and Bias Selection for LCD Display
0	x	x	x	LCD display off
1	0	0	0	1/4 duty, 1/3 bias
1	0	0	1	1/3 duty, 1/3 bias
1	0	1	0	1/2 duty, 1/2 bias
1	0	1	1	1/3 duty, 1/2 bias
1	1	0	0	Static

NOTE: 'X' means 'don't care'.

Table 39. Maximum Number of Display Digits Per Duty Cycle

LCD Duty	LCD Bias	COM Output Pins	Maximum Digit Display (× 8 Segment Pins)
Static	Static	COM0	3
1/2	1/2	COM0–COM1	7
1/3	1/2	COM0–COM2	9
1/3	1/3	COM0–COM2	9
1/4	1/3	COM0–COM3	14

LCD DRIVE VOLTAGE

The LCD display is turned on only whenever the voltage difference between the common and segment signals is greater than V_{LCD}. The LCD display is turned off whenever the difference between the common and segment signal voltages is less than V_{LCD}. The turn-on voltage, $+V_{LCD}$ or $-V_{LCD}$, is generated only when both signals are the selected signals of the bias.

Table 43 shows LCD drive voltages for static mode, 1/2 bias, and 1/3 bias.

Table 40. LCD Drive Voltage Values

LCD Power Supply	Static Mode	1/2 Bias	1/3 Bias
V_{LC0}	V_{LCD}	V_{LCD}	V_{LCD}
V_{LC1}	2/3 V_{LCD}	1/2 V_{LCD}	2/3 V_{LCD}
V_{LC2}	1/3 V_{LCD}	1/2 V_{LCD}	1/3 V_{LCD}
V_{LC3}	0 V	0 V	0 V

NOTE

The LCD panel display may deteriorate if a D.C. voltage is applied that lies between the common and segment signal voltage. Therefore, always drive the LCD panel with A.C. voltage.

LCD VOLTAGE DIVIDING RESISTORS

On-chip voltage dividing resistors for the LCD circuit can be configured by software option (LMOD.7). Figure 58 shows the standard voltage dividing resistor circuits.

Using these optional internal voltage dividing resistors, you can drive either a 3-volt or a 5-volt LCD display using external biasing. Bias pins are connected externally to the V_{LCD} pin so that it can handle the different LCD drive voltages. To cut off the current supply to the voltage dividing resistors, clear LCON.0 when you turn the LCD display off.

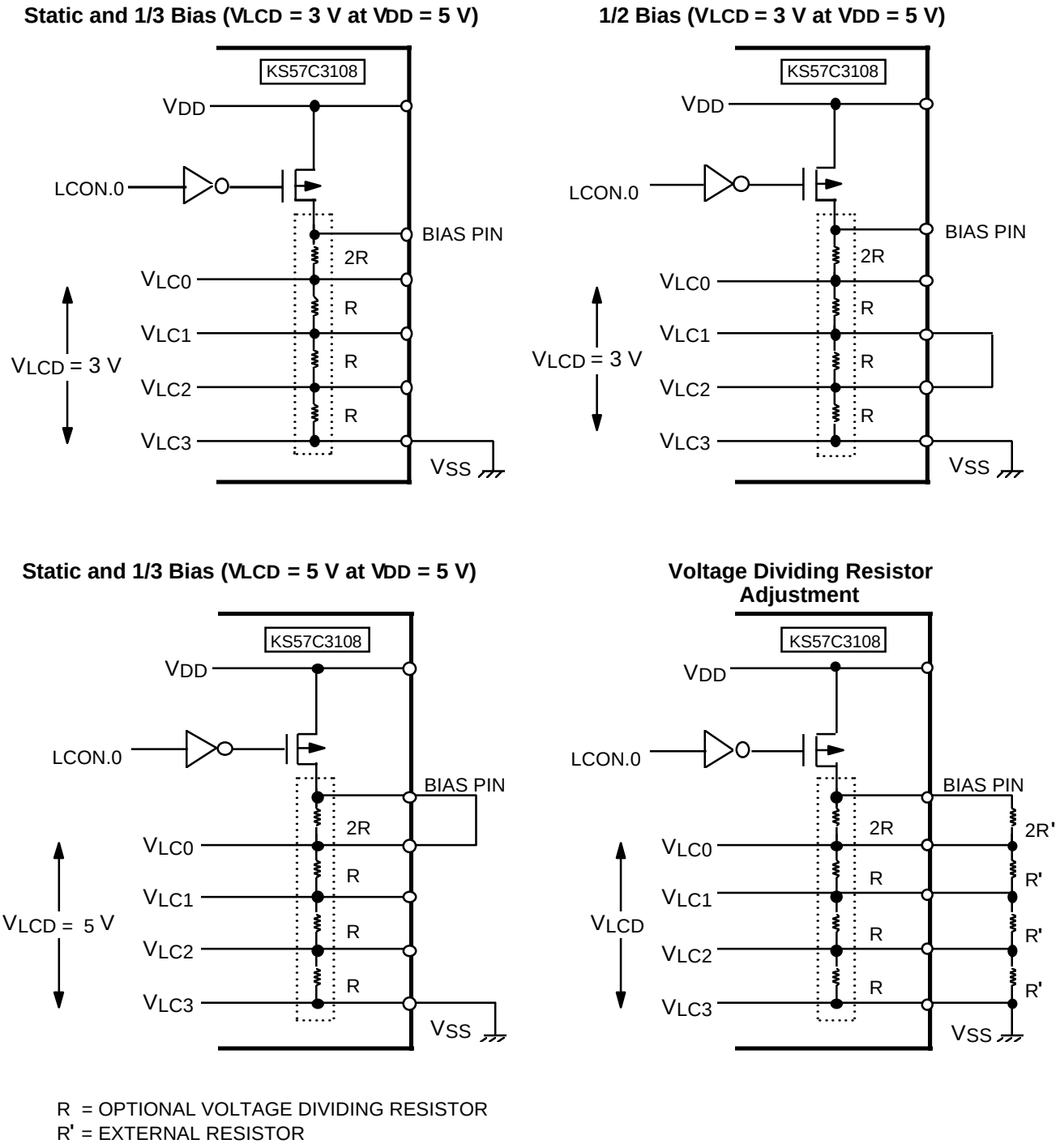


Figure 52. Voltage Dividing Resistor Circuit Diagrams

COMMON (COM) SIGNALS

The common signal output pin selection (COM pin selection) varies according to the selected duty cycle. You should therefore disable any unused COM pins according to this guideline:

— In static mode, disable the COM1, COM2, and COM3 pins

— In 1/2 duty mode, disable the COM2 and COM3 pin

— In 1/3 duty mode, disable the COM3 pin

Any unused COM pins should be opened.

Table 41. Common Signal Pins Used Per Duty Cycle

Display Mode	COM0 Pin	COM1 Pin	COM2 Pin	COM3 Pin
Static	Selected	N/C	N/C	N/C
1/2 duty	Selected	Selected	N/C	N/C
1/3 duty	Selected	Selected	Selected	N/C
1/4 duty	Selected	Selected	Selected	Selected

NOTE: 'NC' means that no connection is required.

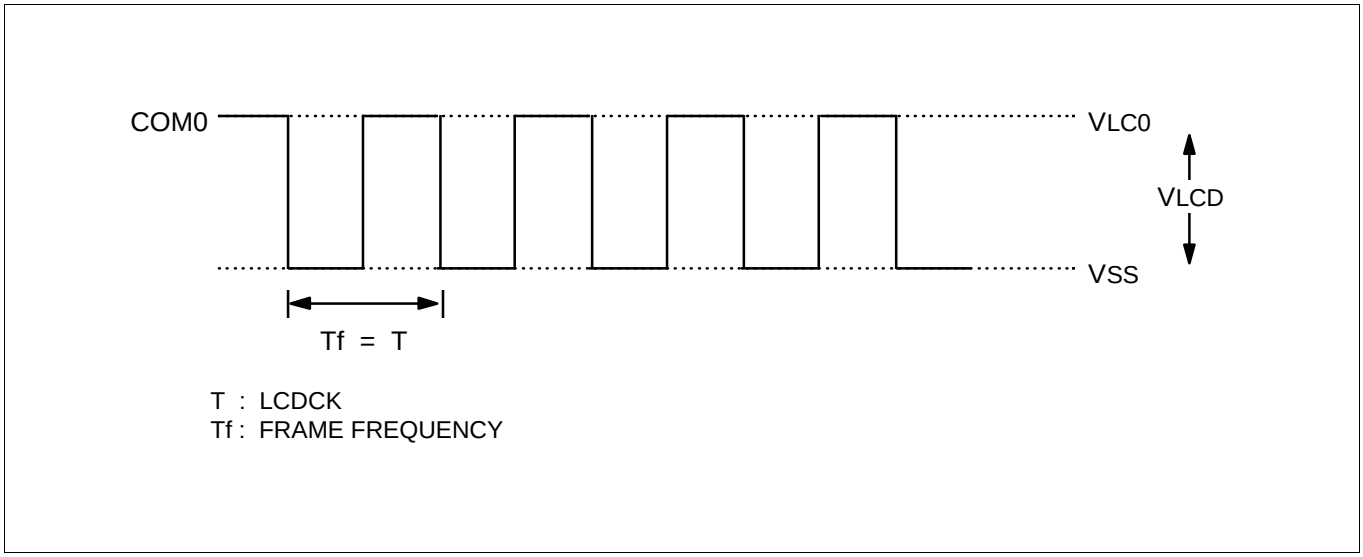


Figure 53. LCD Common Signal Waveform (Static)

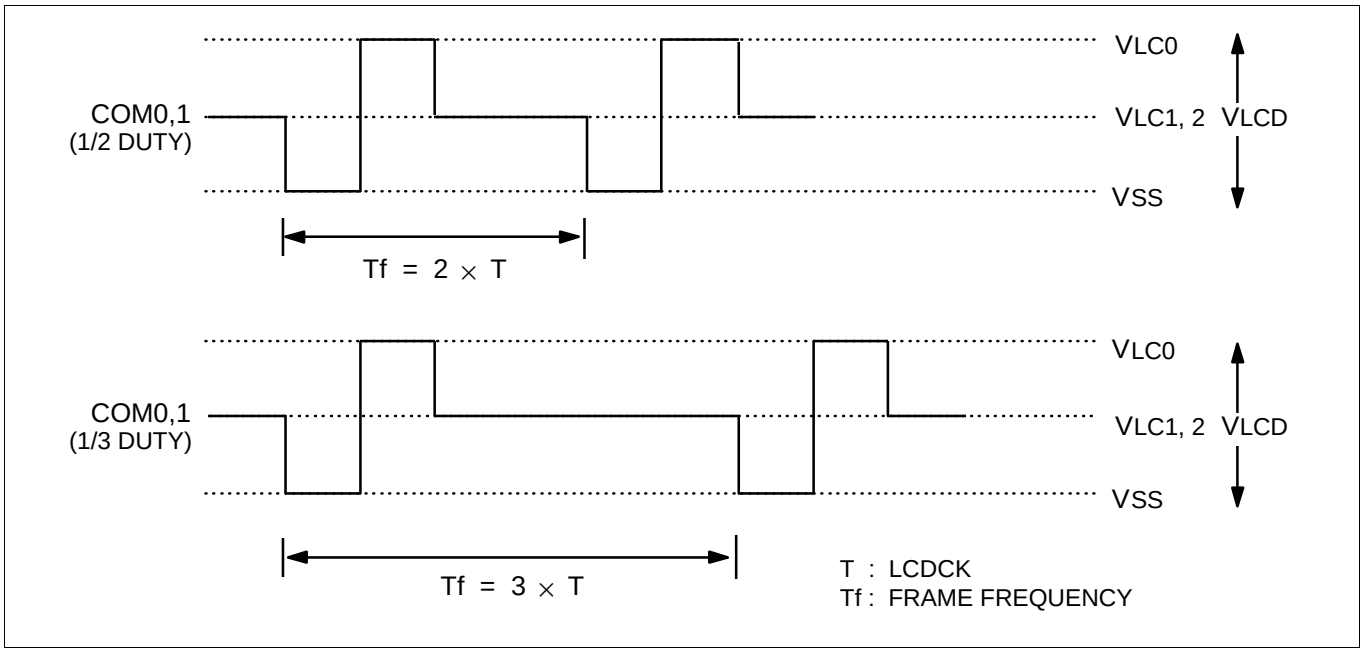


Figure 54. LCD Common Signal Waveforms at 1/2 Bias (1/2, 1/3 Duty)

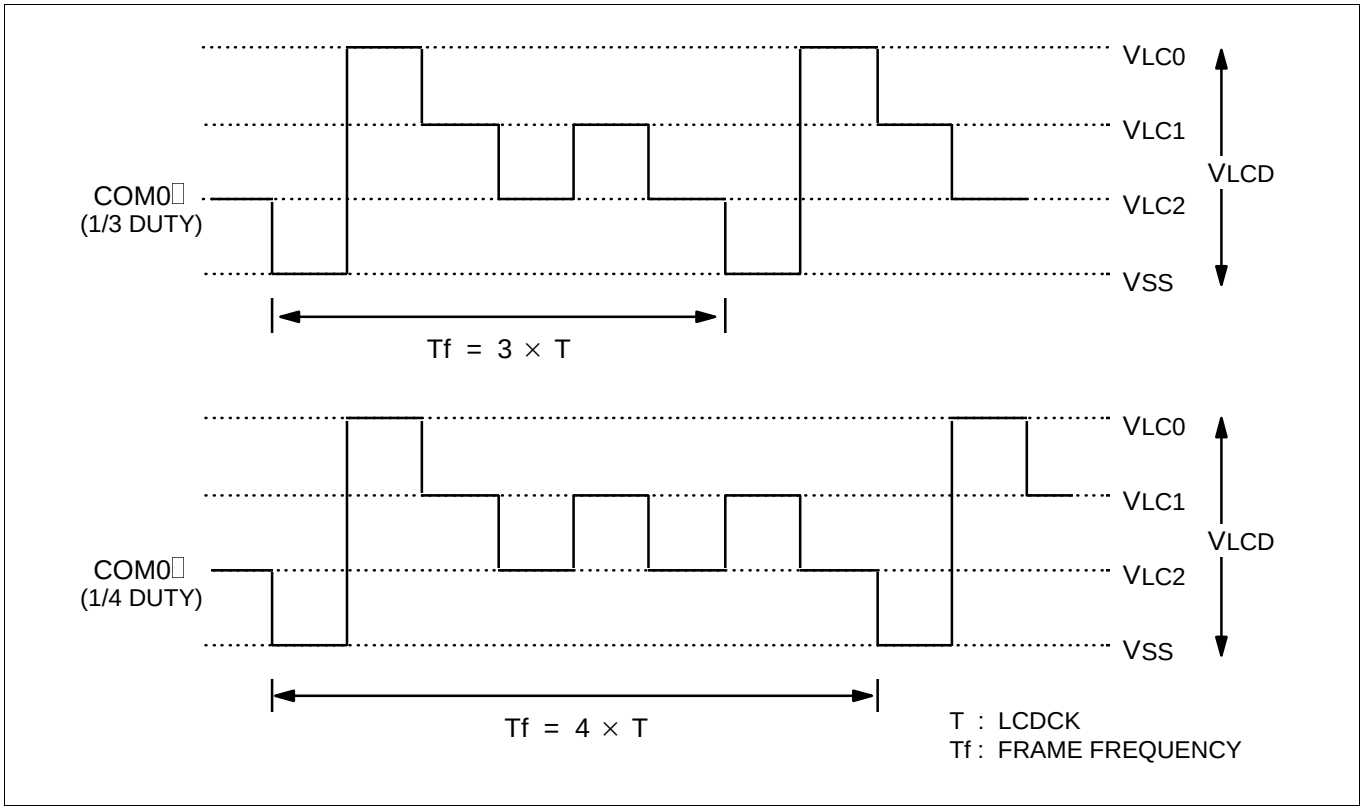


Figure 55. LCD Common Signal Waveforms at 1/3 Bias (1/3, 1/4 Duty)

SEGMENT (SEG) SIGNALS

The 28 LCD segment signal pins are connected to corresponding display RAM locations at 1E4H–1FFH. Bits 0–3 of the display RAM are synchronized with the common signal output pins COM0, COM1, COM2, and COM3.

When the bit value of a display RAM location is "1", a select signal is sent to the corresponding segment pin. When the display bit is "0", a 'no-select' signal is sent to the corresponding segment pin. Each bias has select and no-select signals.

Table 42. Select/No-Select Signals for LCD Static Display Mode

COM	SEG	Select	Non-select
		V_{LC0} / V_{SS}	V_{SS} / V_{LC0}
V_{SS} / V_{LC0}		– V_{LC0} / + V_{LC0}	0 V / 0 V

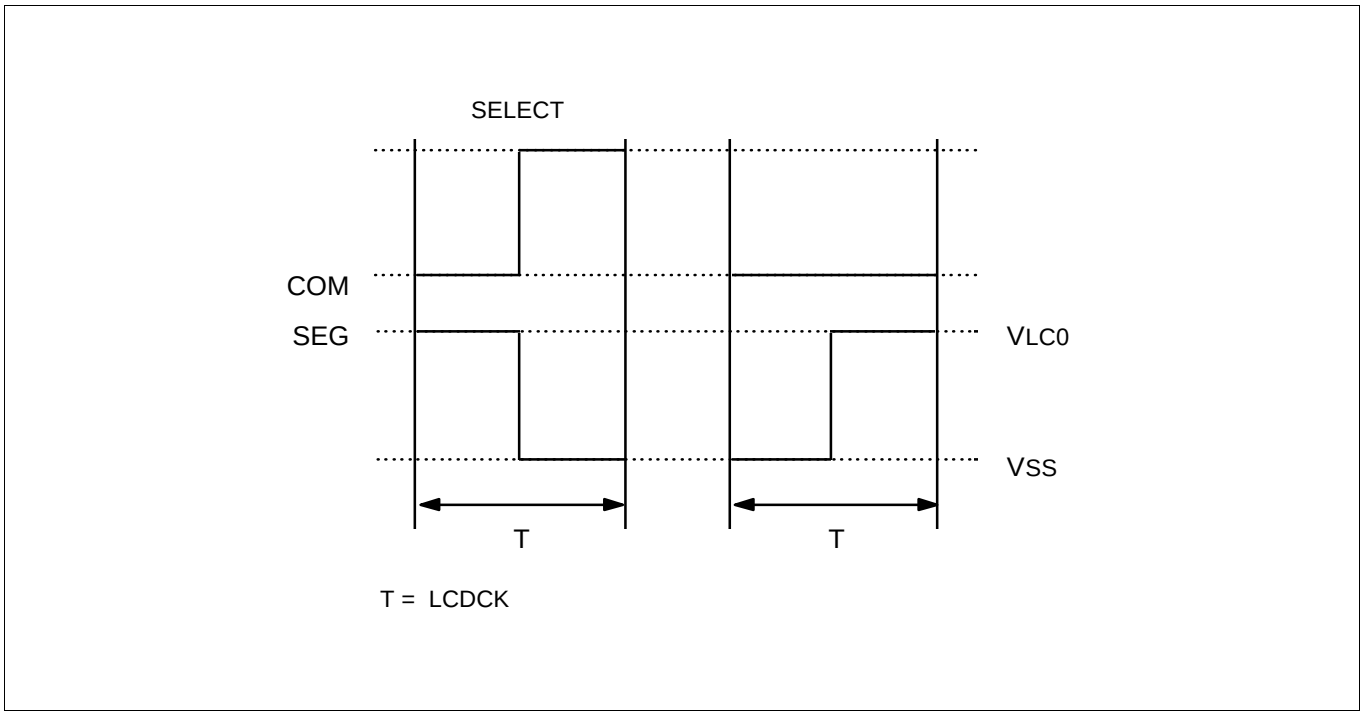


Figure 56. Select/No-Select Bias Signals in Static Display Mode

Table 43. Select/No-Select Signals for LCD 1/2 Bias Display Mode

COM	SEG	Select	Non-select
		V_{LC0} / V_{SS}	V_{SS} / V_{LC0}
Select	V_{SS} / V_{LC0}	– V_{LCD} / + V_{LCD}	0 V / 0 V
Non-select	V_{LC1} = V_{LC2}	– 1/2 V_{LCD} / + 1/2 V_{LCD}	+ 1/2 V_{LCD} / – 1/2 V_{LCD}

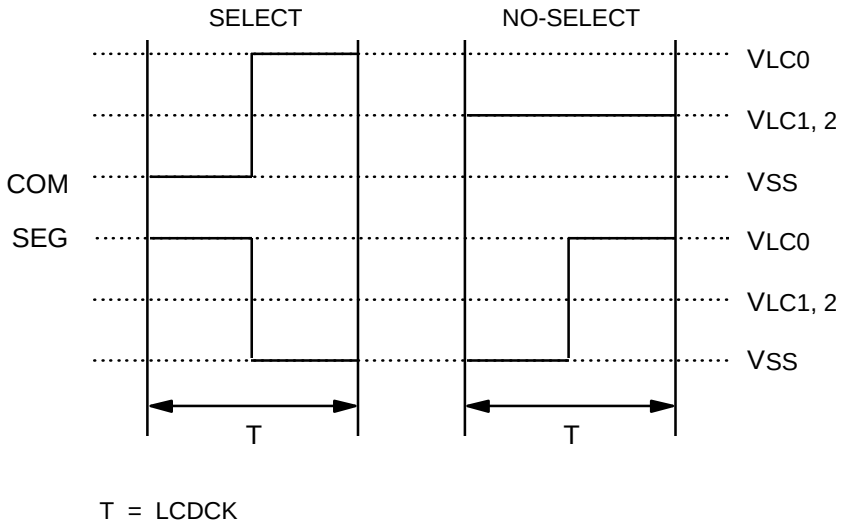


Figure 57. Select/No-Select Bias Signals in 1/2 Bias Display Mode

Table 44. Select/No-Select Signals for LCD 1/3 Bias Display Mode

COM	SEG	Select	Non-select
		V_{LC0} / V_{SS}	V_{SS} / V_{LC1}
Select	V_{SS} / V_{LC0}	– V_{LCD} / + V_{LCD}	– 1/3 V_{LCD} / + 1/3 V_{LCD}
Non-select	V_{LC1} / V_{LC2}	– 1/3 V_{LCD} / + 1/3 V_{LCD}	+ 1/3 V_{LCD} / – 1/3 V_{LCD}

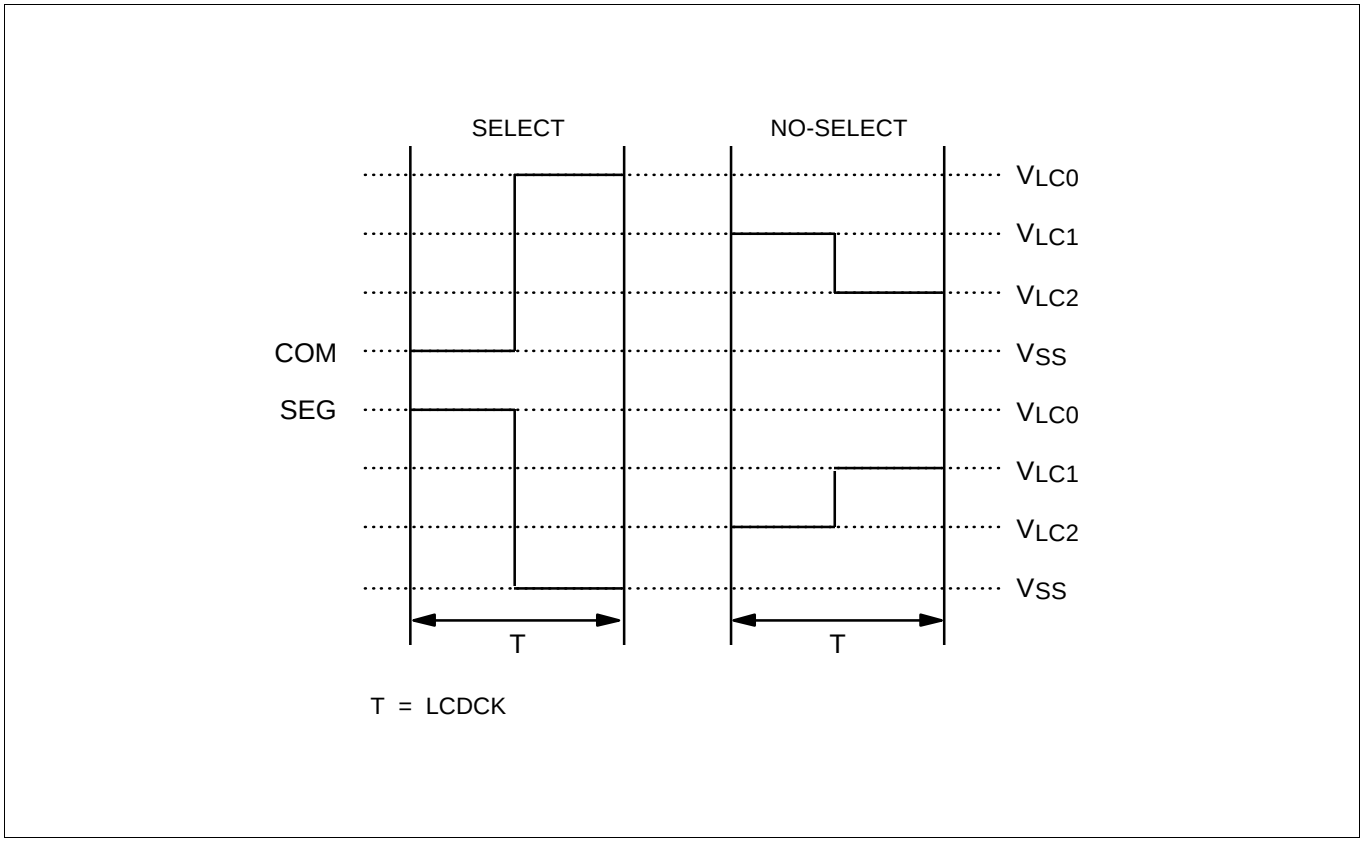


Figure 58. Select/No-Select Signals in 1/3 Bias Display Mode

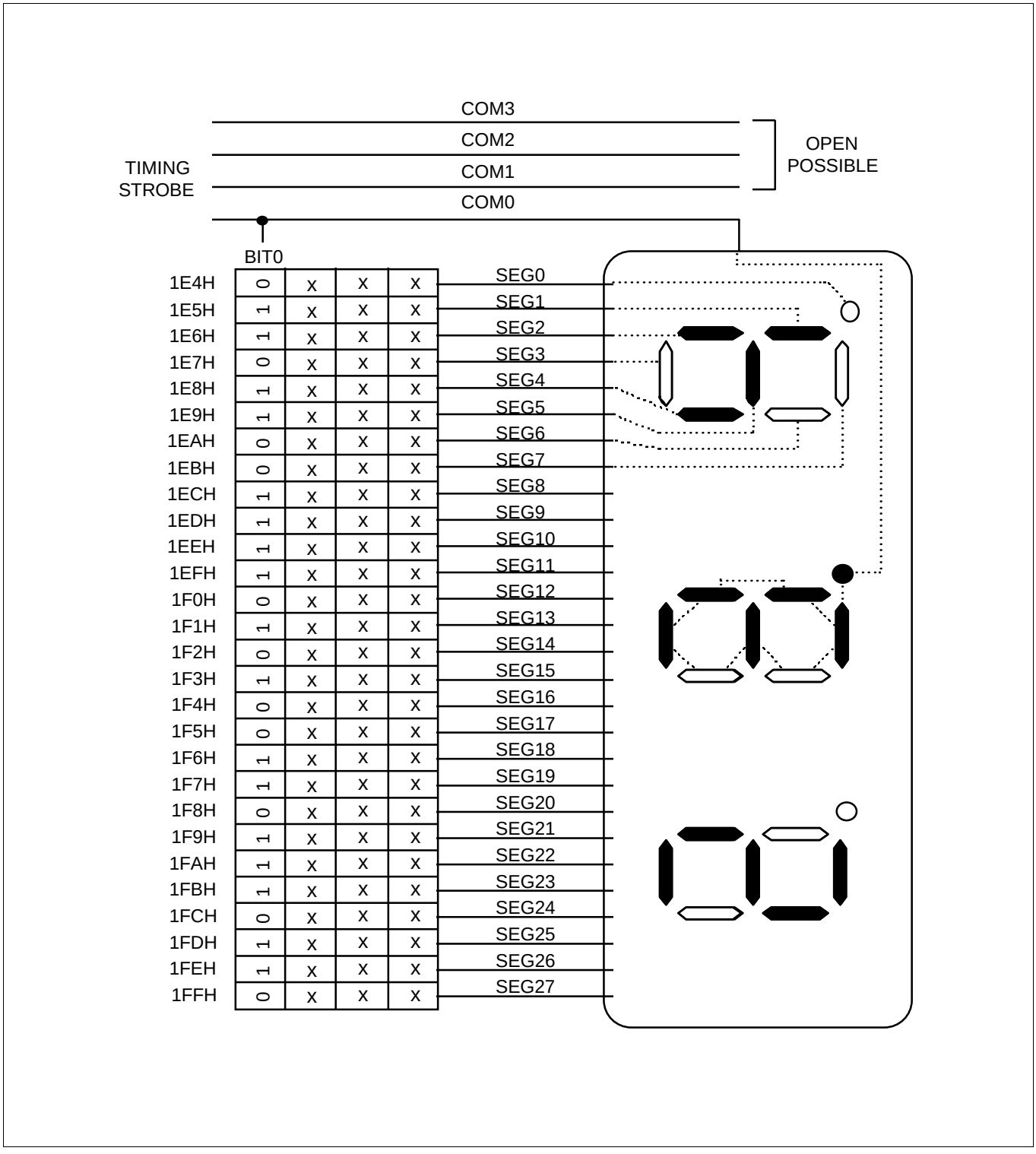


Figure 59. LCD Connection Example (Static)

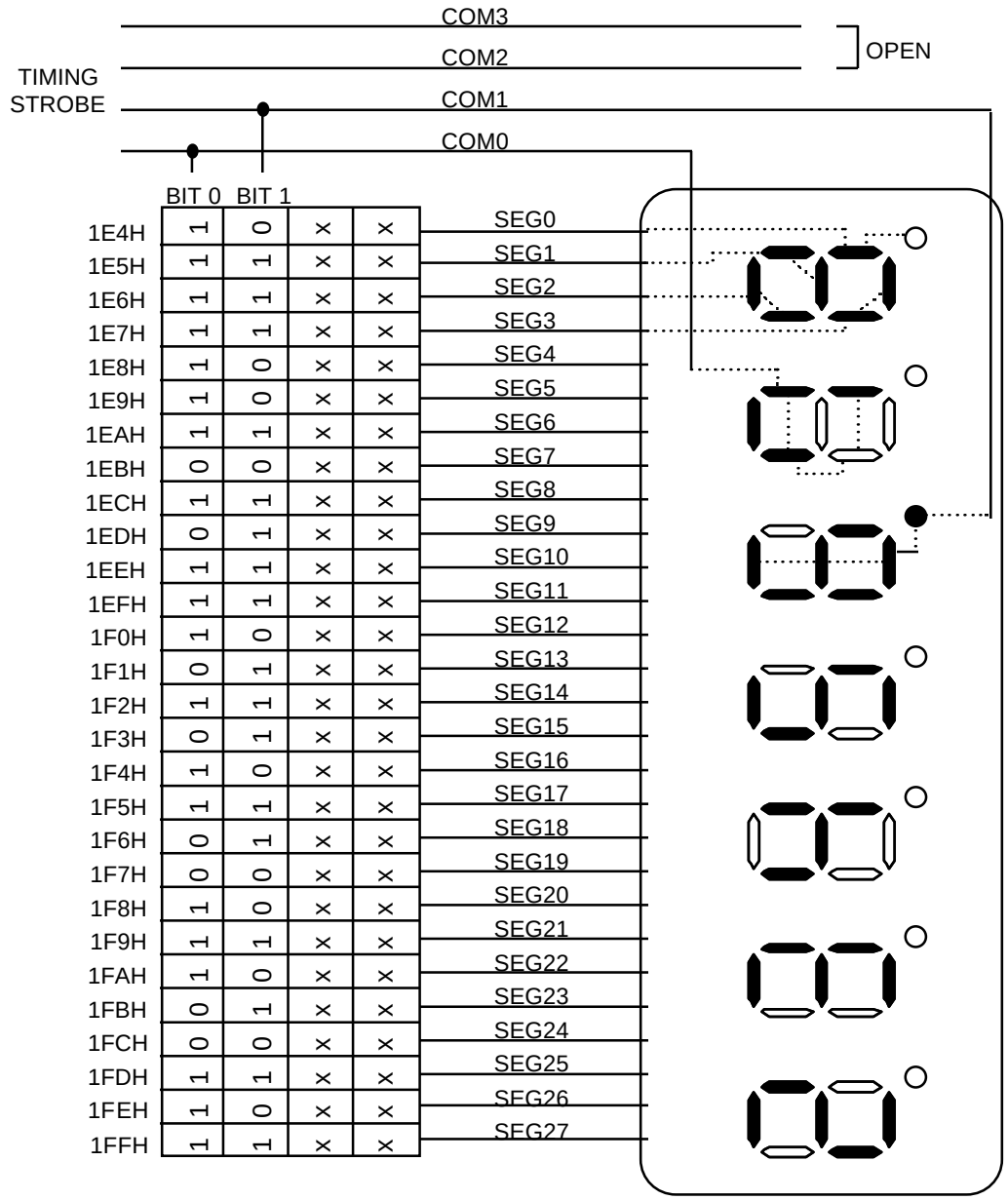


Figure 60. LCD Connection Example (1/2 Duty, 1/2 Bias)

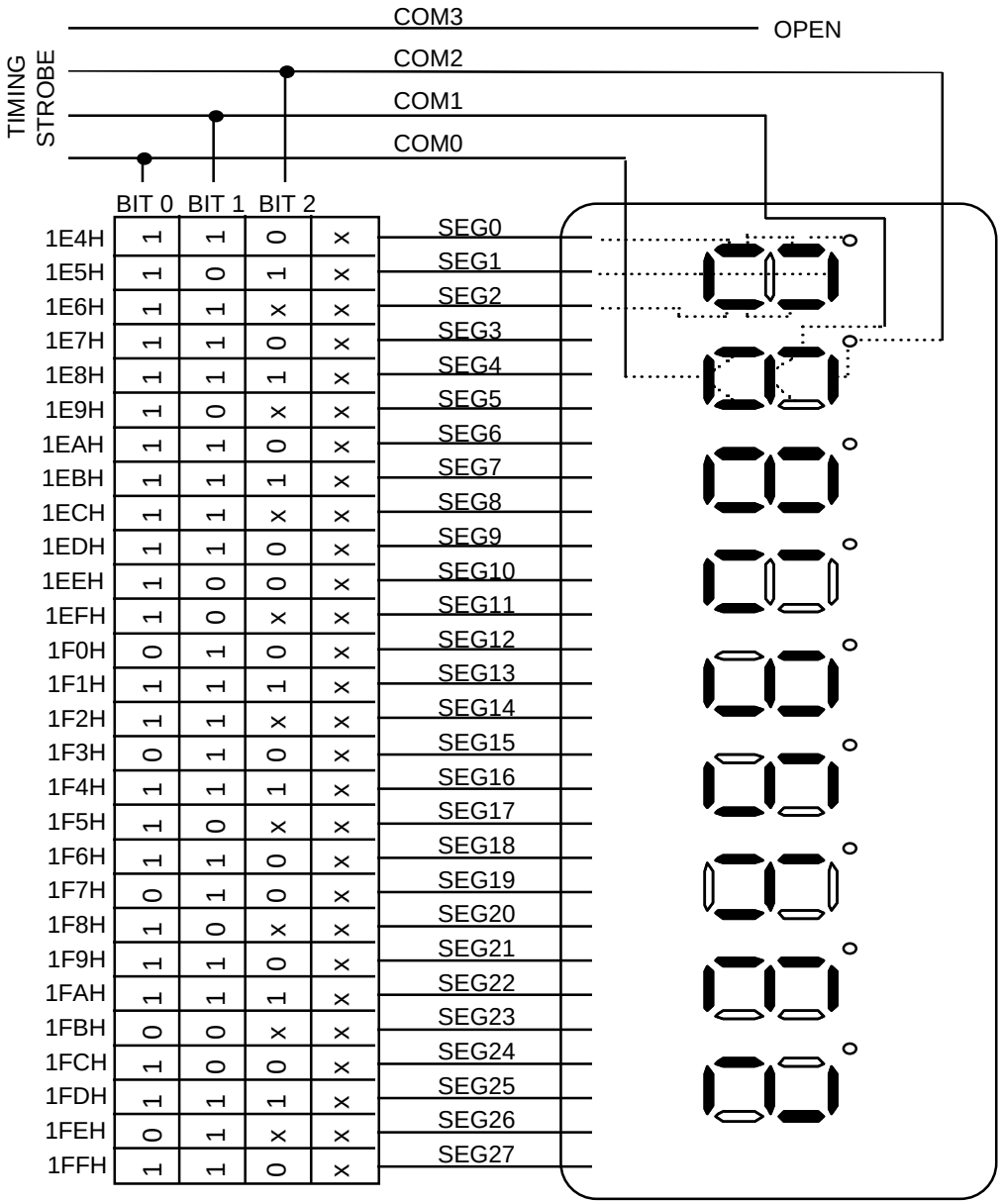


Figure 61. LCD Connection Example (1/3 Duty, 1/3 Bias)

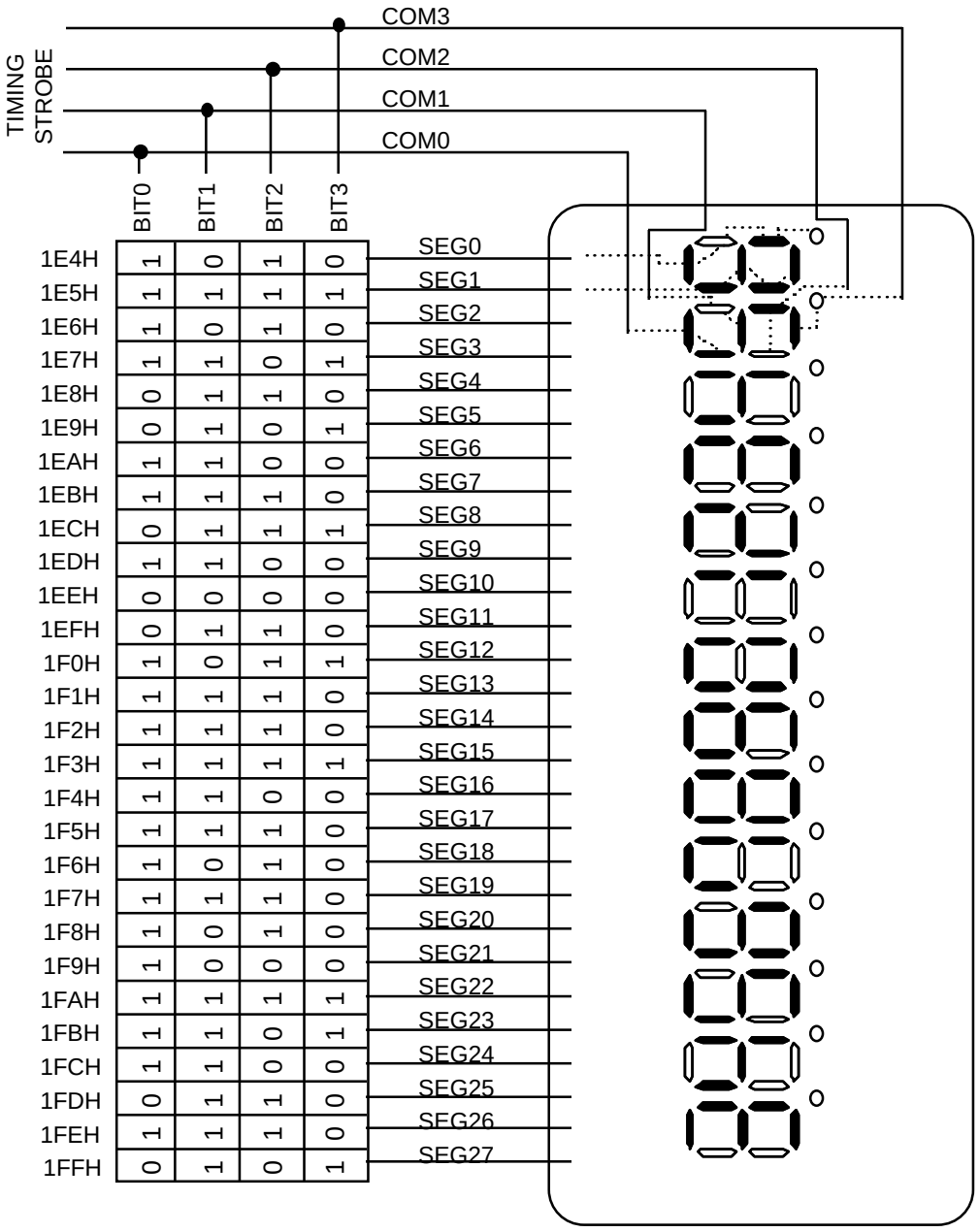


Figure 62. LCD Connection Example (1/4 Duty, 1/3 Bias)

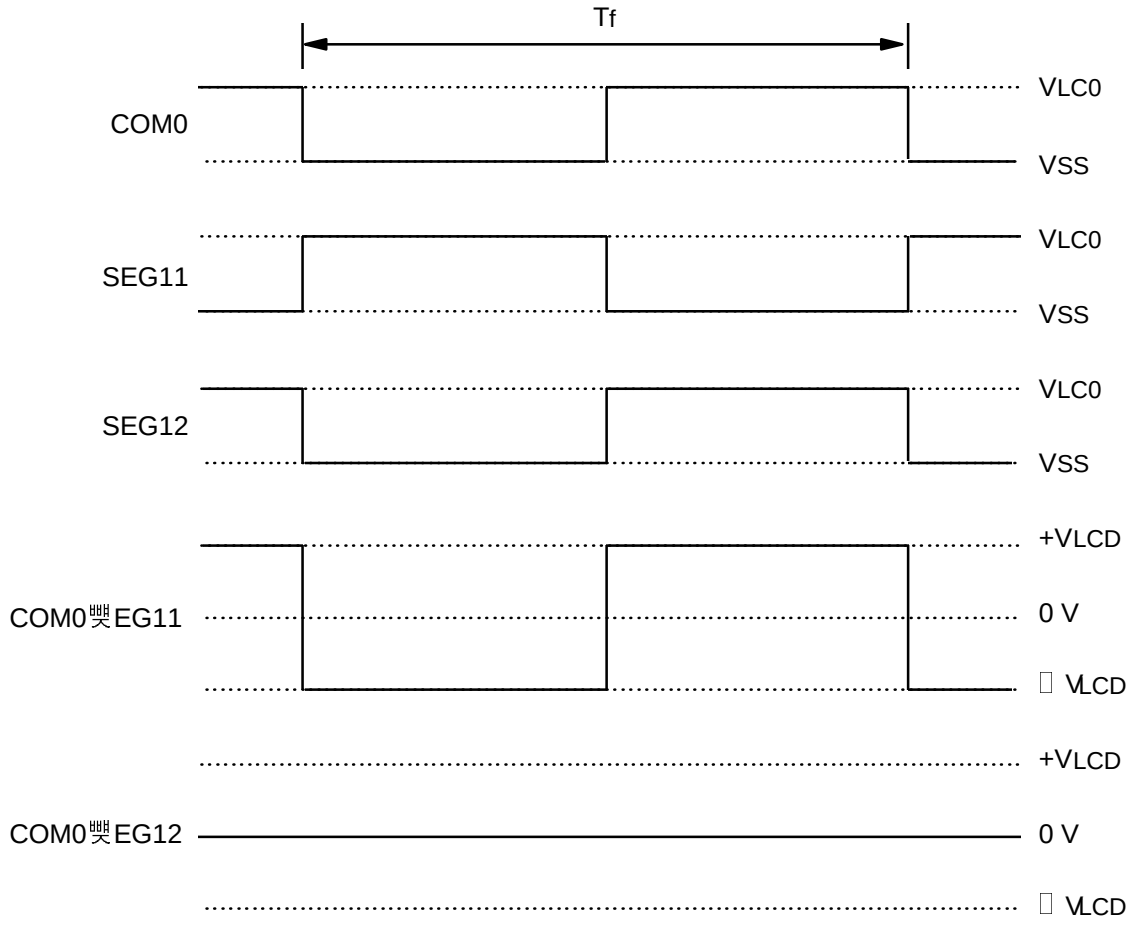


Figure 63. LCD Signal Waveforms in Static Mode

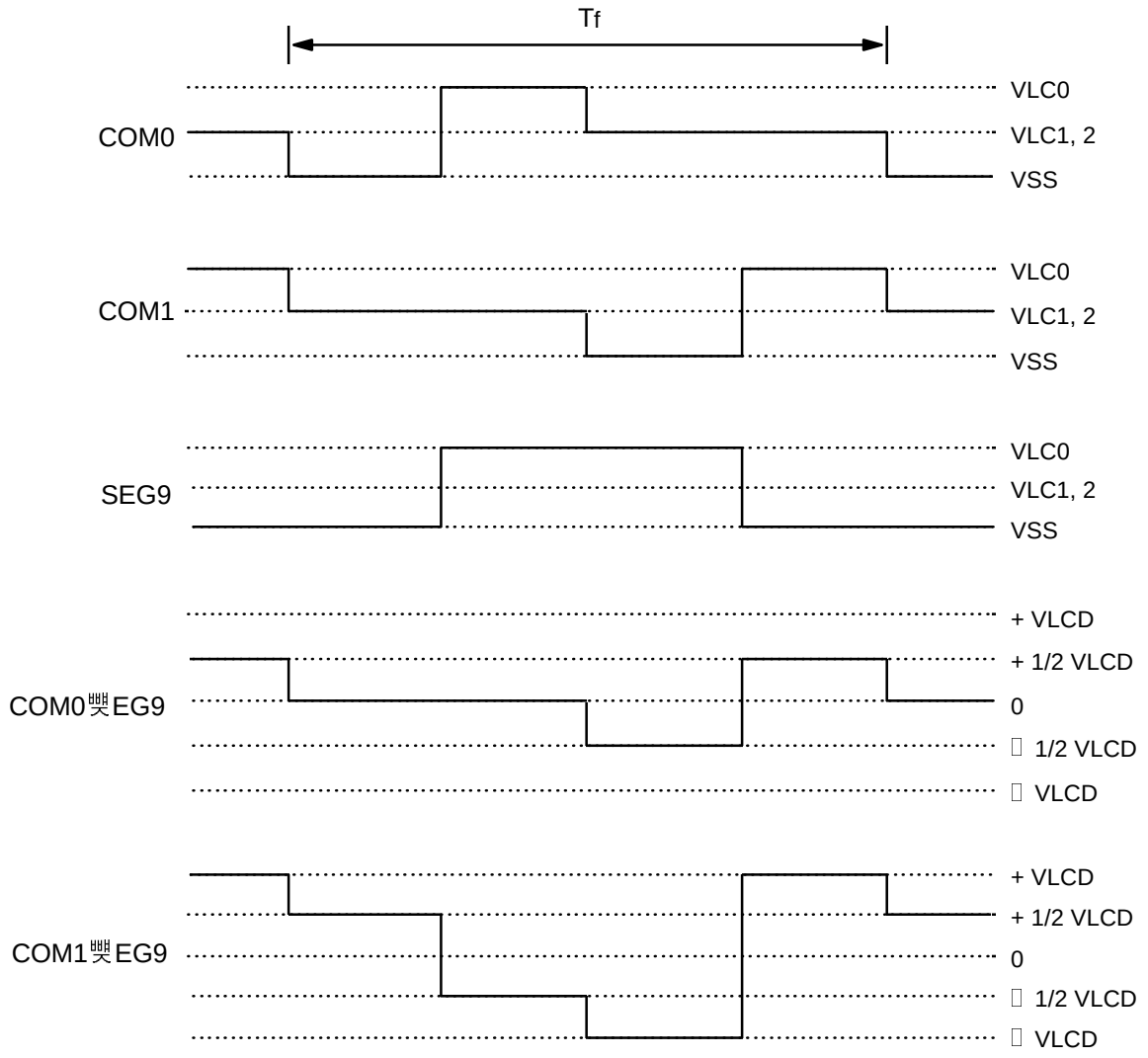


Figure 64. LCD Signal Waveforms at 1/2 Duty, 1/2 Bias

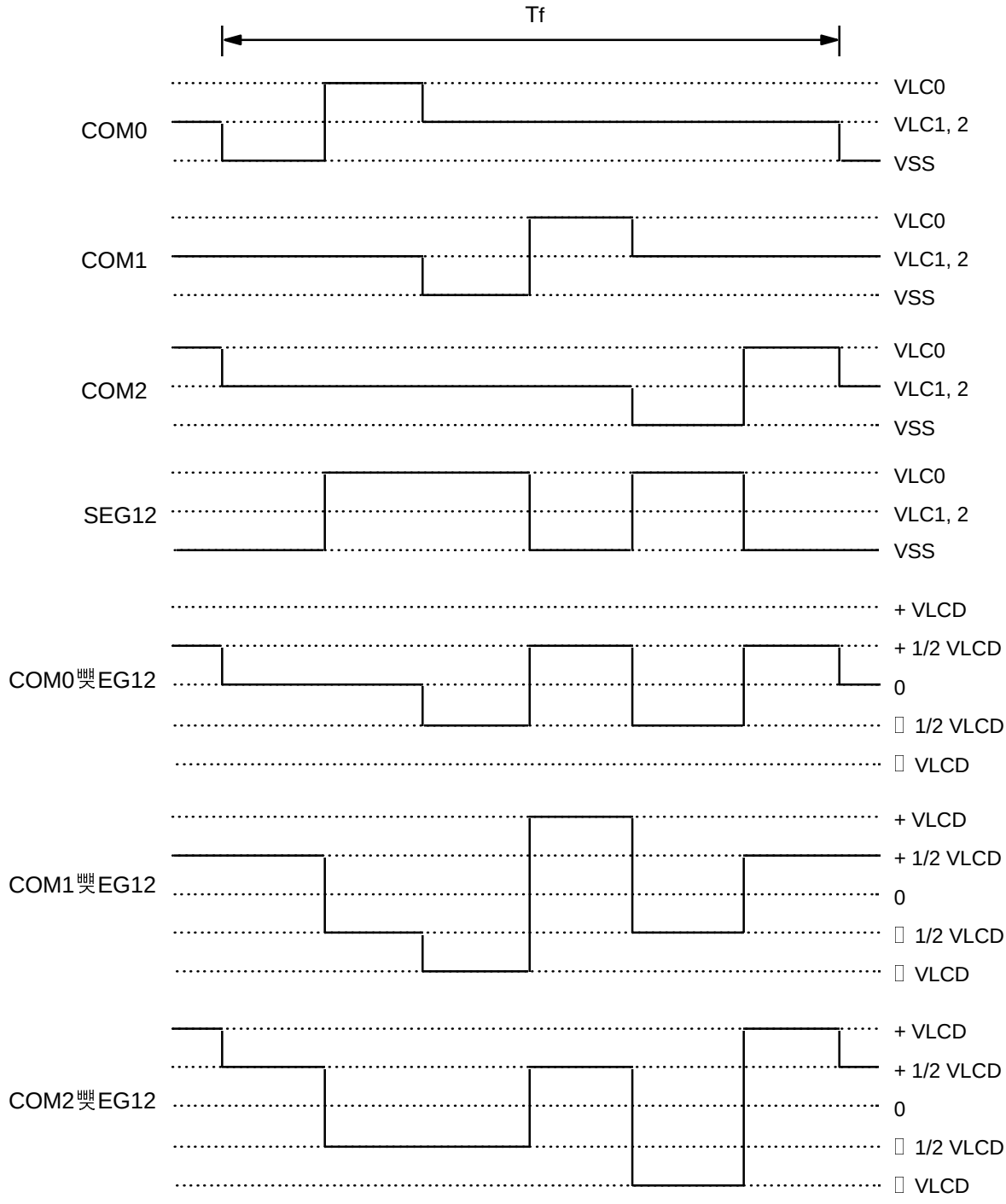


Figure 65. LCD Signal Waveforms at 1/3 Duty, 1/2 Bias

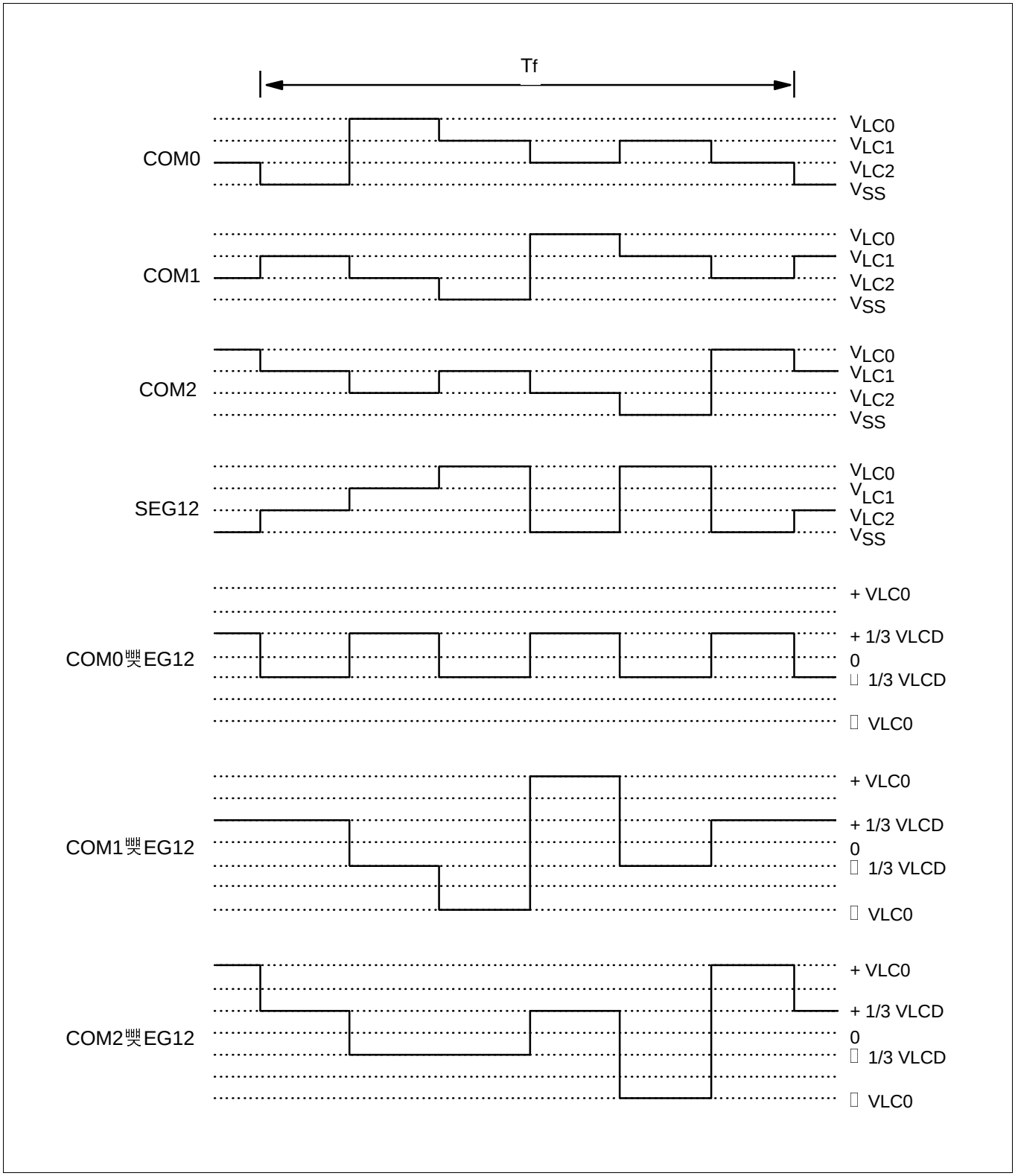


Figure 66. LCD Signal Waveforms at 1/3 Duty, 1/3 Bias

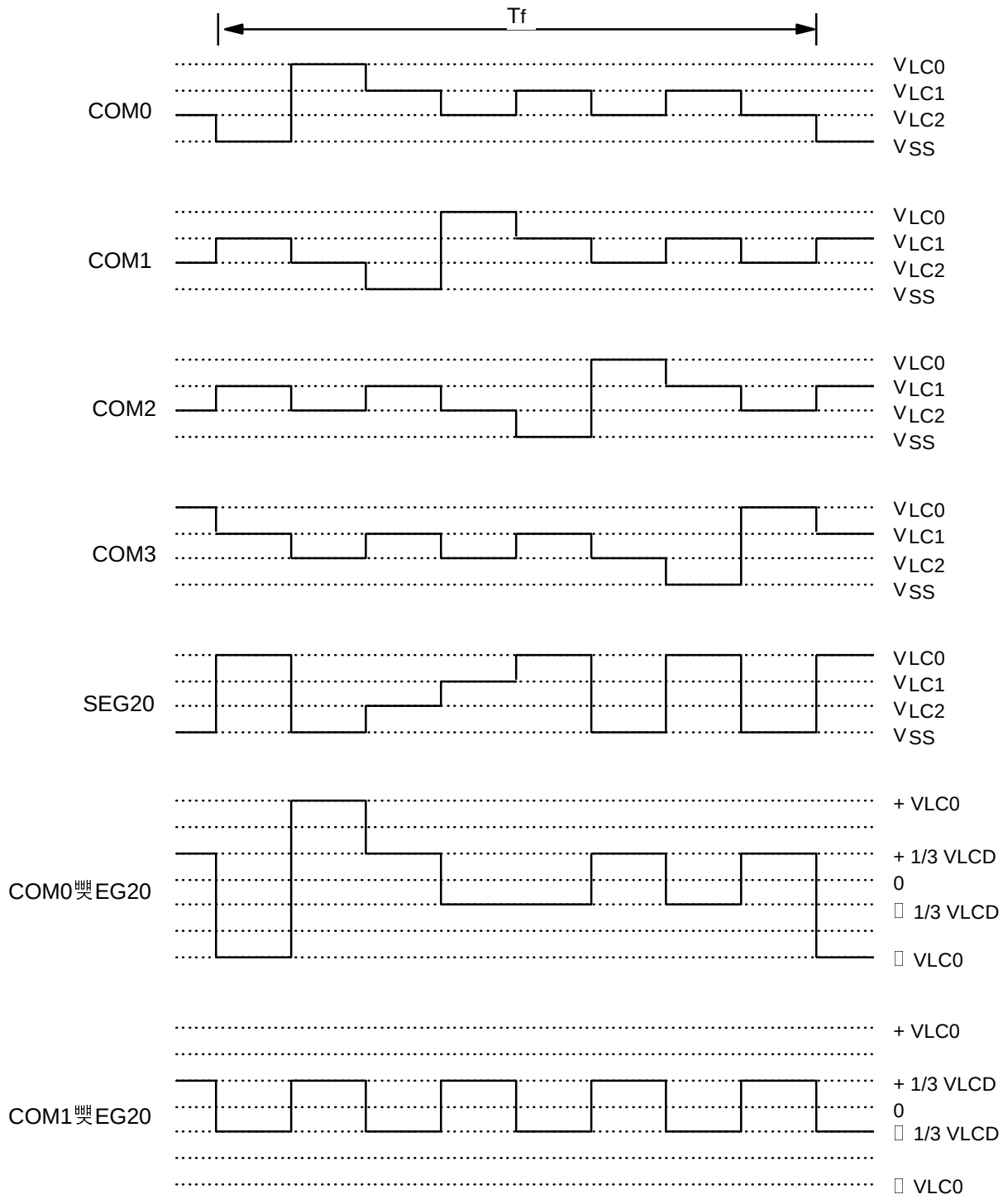


Figure 67. LCD Signal Waveforms at 1/4 Duty, 1/3 Bias

SERIAL I/O INTERFACE

Using the serial I/O interface, 8-bit data can be exchanged with an external device. The transmission frequency is controlled by making the appropriate bit settings to the SMOD register.

The serial interface can run off an internal or an external clock source, or the TOL0 signal that is generated by the 8-bit timer/counter, TC0. If the TOL0 clock signal is used, you can modify its frequency to adjust the serial data transmission rate.

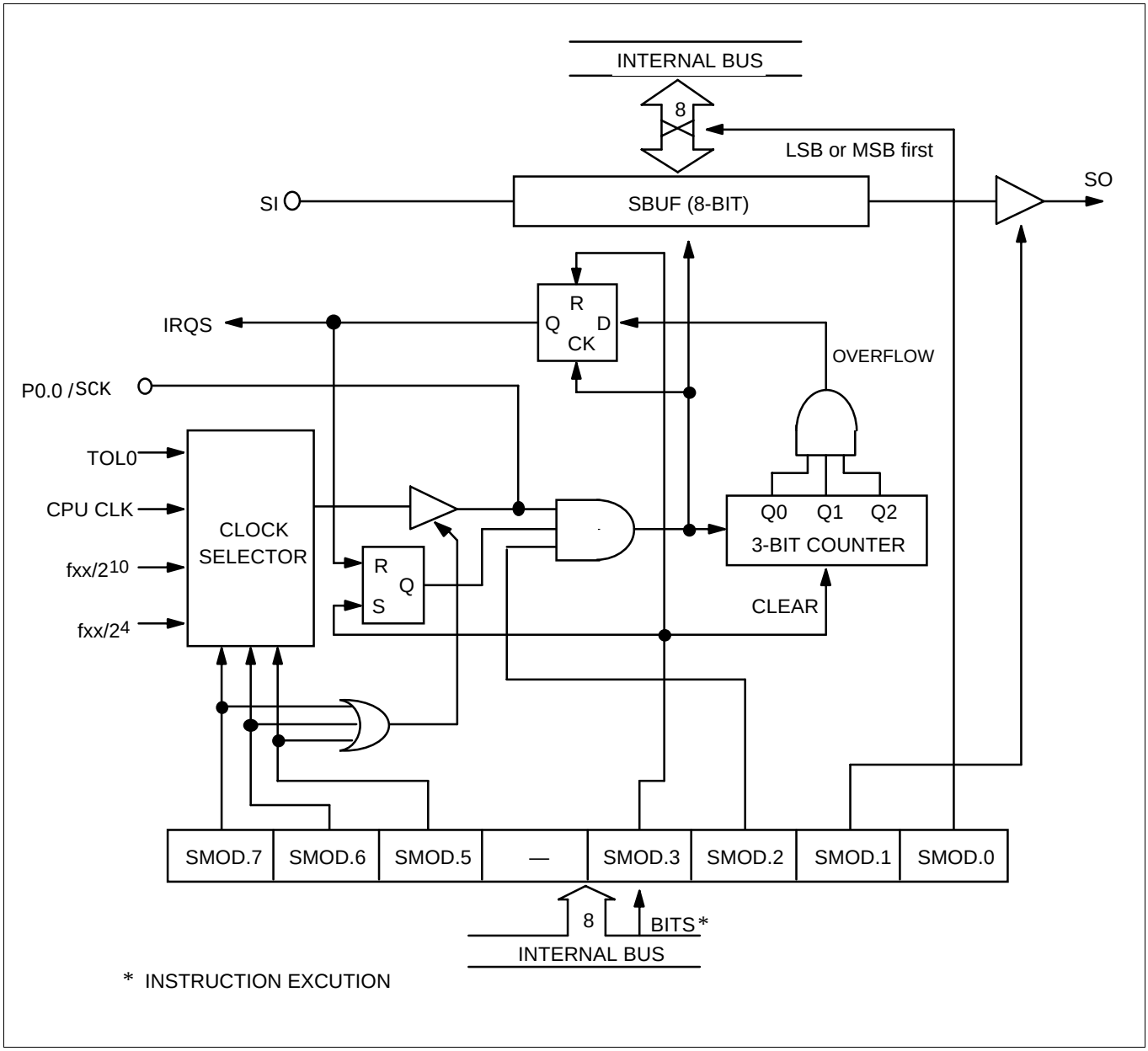


Figure 68. Serial I/O Interface Circuit Diagram

SERIAL I/O MODE REGISTER (SMOD)

The serial I/O mode register, SMOD, is an 8-bit register that specifies the operation mode of the serial interface. Its reset value is logical zero. SMOD is organized in two 4-bit registers, as follows:

SMOD register settings let you select either MSB-first or LSB-first serial transmission, and to operate in transmit-and-receive mode or receive-only mode. SMOD is a write-only register and can be addressed only by 8-bit RAM control instructions. An exception is SMOD.3, which can be written by a 1-bit RAM control instruction. When SMOD.3 is set to "1", the erial interface interrupt request flag, IRQS, and the 3-bit serial clock counter value are cleared to zero, and an SIO operation is initiated. When the SIO transmission starts, SMOD.3 is cleared to "0".

Table 45. SIO Mode Register (SMOD) Organization

SMOD.0	0	Most significant bit (MSB) is transmitted first
	1	Least significant bit (LSB) is transmitted first
SMOD.1	0	Receive-only mode
	1	Transmit-and-receive mode
SMOD.2	0	Disable the data shifter and clock counter; retain contents of IRQS flag when serial transmission is completed.
	1	Enable the data shifter and clock counter; set IRQS flag to "1" when serial transmission is completed.
SMOD.3	1	Clear IRQS flag and 3-bit clock counter to "0"; initiate transmission and then reset this bit to "0"
SMOD.4	0	Bit not used; value is always "0"

SMOD.7	SMOD.6	SMOD.5	Clock Selection	R/W Status of SBUF
0	0	0	External clock at SCK pin	SBUF is enabled when SIO operation is halted or when SCK goes High.
0	0	1	Use TOL0 clock from TC0	
0	1	x	CPU clock: fxx/4, fxx/8, fxx/64	Enable SBUF read/write
1	0	0	4.39 kHz clock: $fxx/2^{10}$	SBUF is enabled when SIO operation is halted or when SCK goes High.
1	1	1	281 kHz clock: $fxx/2^4$	

NOTES:
1. 'fxx' = system clock; 'x' means 'don't care.'
2. kHz frequency ratings assume a system clock (fxx) running at 4.5 MHz.
3. The SIO clock selector circuit cannot select a $fxx/2^4$ clock if the CPU clock is fxx/64.

SERIAL I/O TIMING DIAGRAMS

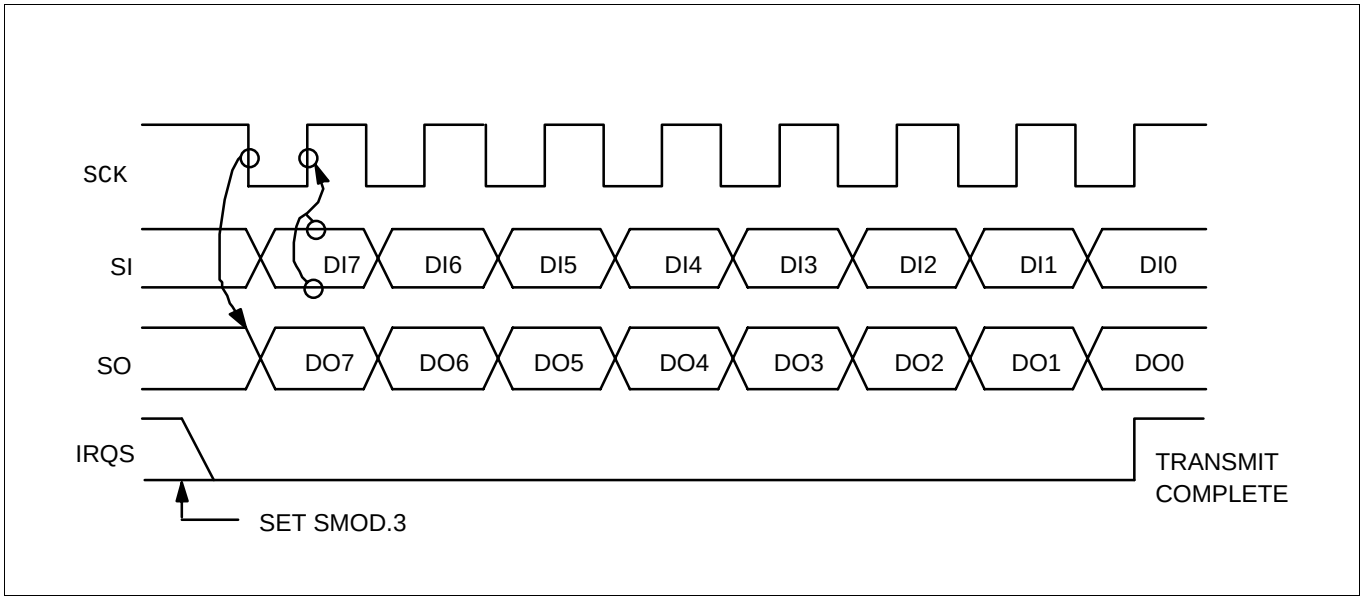


Figure 69. SIO Timing in Transmit/Receive Mode

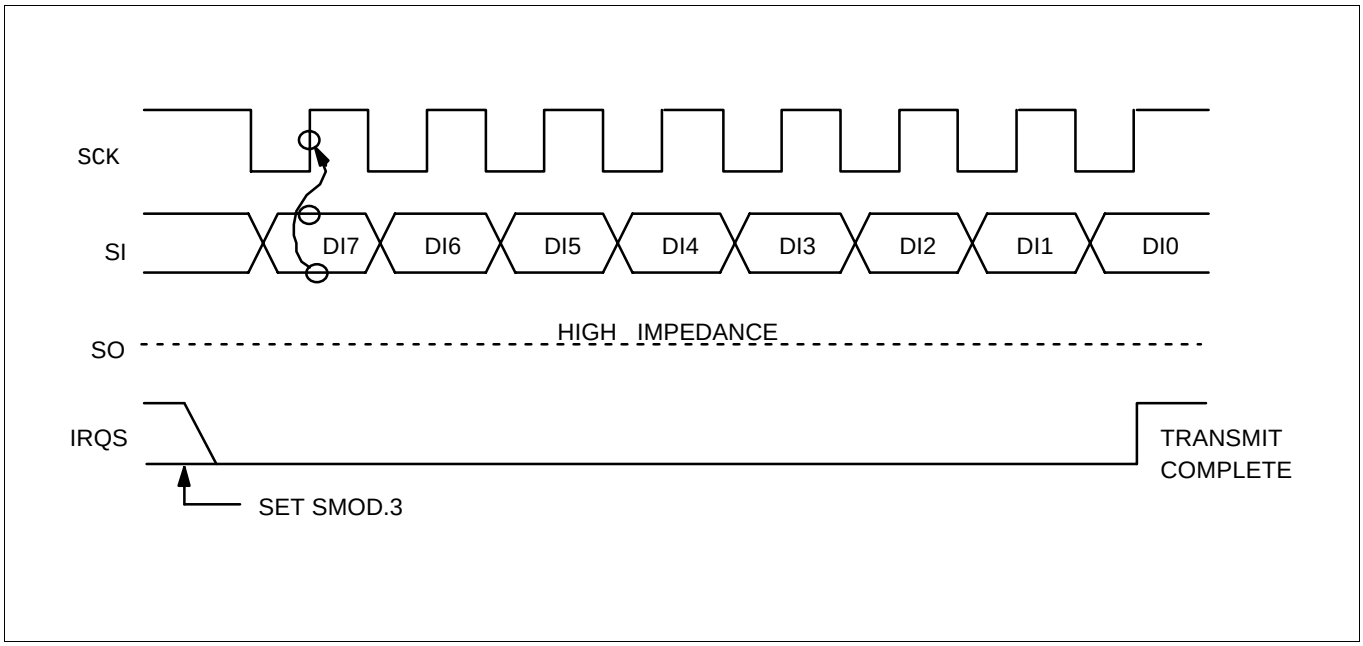


Figure 70. SIO Timing in Receive-Only Mode

SERIAL I/O BUFFER REGISTER (SBUF)

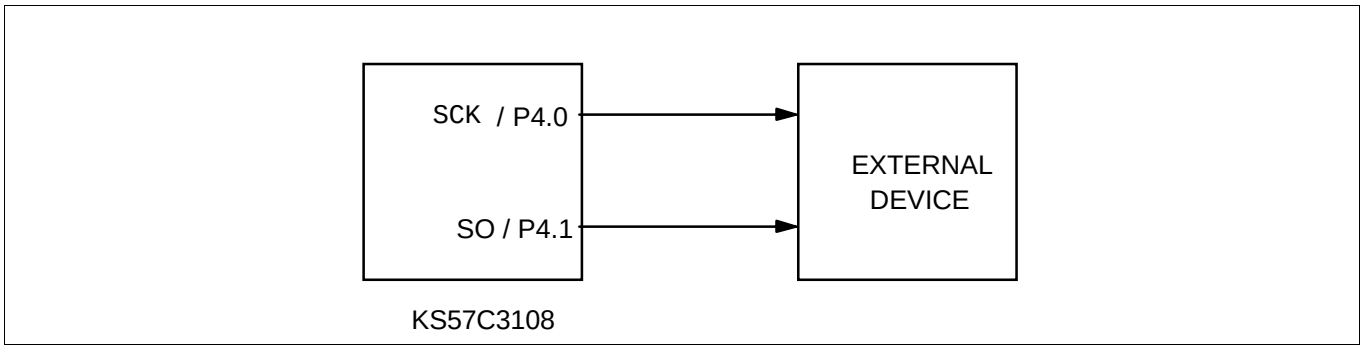
The serial I/O buffer register ,SBUF, can be read or written using 8-bit RAM control instructions. After a reset operation, the value of SBUF is undetermined.

When the serial interface operates in transmit-and-receive mode (SMOD.1 = "1"), transmit data in the SIO buffer register are output to the SO pin (P4.1) at the rate of one bit for each falling edge of the SIO clock. Receive data are simultaneously input from the SI pin (P4.2) to SBUF at the rate of one bit for each rising edge of the SIO clock. When receive-only mode is used, incoming data are input to the SIO buffer at the rate of one bit for each rising edge of the SIO clock.

☞ PROGRAMMING TIP — Setting Transmit/Receive Modes for Serial I/O

1. Transmit the data value 48H through the serial I/O interface using an internal clock frequency of fxx/2^4 and in MSB-first mode:

```
        BITS    EMB
        SMB     15
        LD      EA,#03H
        LD      PMG2,EA         ;  P4.0 / SCK and P4.1 / SO ← output
        LD      EA,#48H         ;
        LD      SBUF,EA         ;
        LD      EA,#0EEH
        LD      SMOD,EA         ;  SIO data transfer
```



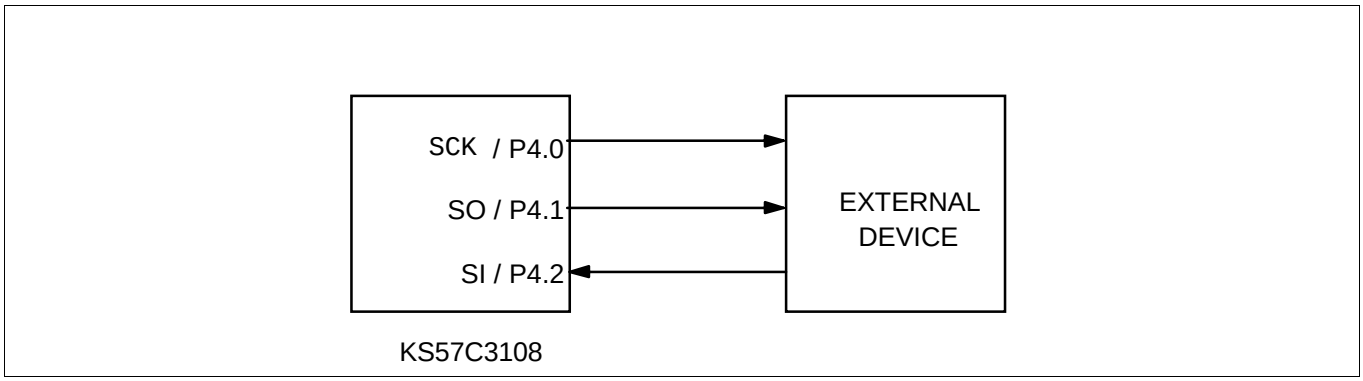
2. Use CPU clock to transfer and receive serial data at high speed:

```
        BITR    EMB
        LD      EA,#03H
        LD      PMG2,EA         ;  P4.0 / SCK and P4.1 / SO ← Output, P4.2 / SI ← Input
        LD      EA,TDATA        ;  TDATA address = Bank0 (20H–7FH)
        LD      SBUF,EA
        LD      EA,#4FH
        LD      SMOD,EA         ;  SIO start
        BITR    IES
STEST   BTSTZ   IRQS
        JR      STEST
        LD      EA,SBUF
        LD      RDATA,EA        ;  RDATA address = Bank0 (20H–7FH)
```

☞ PROGRAMMING TIP — Setting Transmit/Receive Modes for Serial I/O (Continued)

3. Transmit and receive an internal clock frequency of 4.09 kHz (at 4.19 MHz) in LSB-first mode:

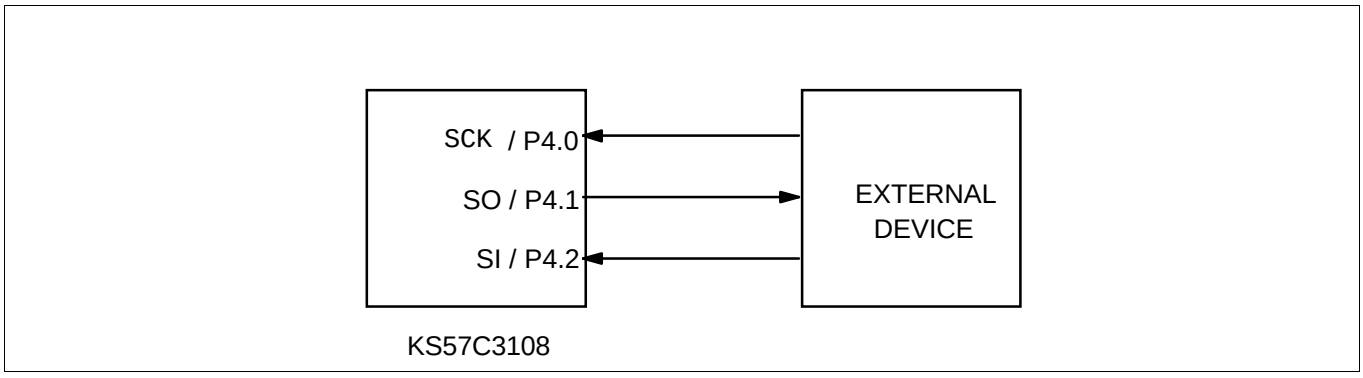
```
        BITR    EMB
        LD      EA,#03H
        LD      PMG2,EA         ;  P4.0 / SCK and P4.1 / SO ← Output, P4.2/SI ← Input
        LD      EA,TDATA        ;  TDATA address = Bank0 (20H–7FH)
        LD      SBUF,EA
        LD      EA,#8FH
        LD      SMOD,EA         ;  SIO start
        EI
        BITS    IES
        .
        .
        .
NTS     PUSH    SB              ;  Store SMB, SRB
        PUSH    EA              ;  Store EA
        BITR    EMB
        LD      EA,TDATA        ;  EA ← transmit data
                                ;  TDATA  address = bank 0 (20H–7FH)
        XCH     EA,SBUF         ;  EA ← receive data
        LD      RDATA,EA        ;  RDATA ← receive data
                                ;  RDATA address = bank 0 (20H–7FH)
        BITS    SMOD.3          ;  SIO start
        POP     EA
        POP     SB
        IRET
```



☞ **PROGRAMMING TIP — Setting Transmit/Receive Modes for Serial I/O (Continued)**

4. Transmit and receive an external clock in LSB-first mode:

```
        BITR    EMB
        LD      EA,#02H
        LD      PMG2,EA         ;  P4.1 / SO ← Output, P4.0 / SCK and P4.2 / SI ← Input
        LD      EA,TDATA
        LD      SBUF,EA
        LD      EA,#0FH
        LD      SMOD,EA         ;  SIO start
        EI
        BITS    IES
        .
        .
        .
NTS     PUSH    SB              ;  Store SMB, SRB
        PUSH    EA              ;  Store EA
        BITR    EMB
        LD      EA,TDATA        ;  EA ← transmit data
                                ;  TDATA  address = bank 0 (20H–7FH)
        XCH     EA,SBUF         ;  EA ← receive data
        LD      RDATA,EA        ;  RDATA ← receive data
                                ;  RDATA address = bank 0 (20H–7FH)
        BITS    SMOD.3          ;  SIO start
        POP     EA
        POP     SB
        IRET
```



High Speed SIO Transmission

ELECTRICAL DATA

Table 46. Absolute Maximum Ratings

(T_A = 25 °C)

Parameter	Symbol	Conditions	Rating	Units
Supply Voltage	V_{DD}	—	– 0.3 to + 7.0	V
Input Voltage	V_I	All I/O ports	– 0.3 to V_{DD} + 0.3 (With pull-up resistor)	V
Output Voltage	V_O	—	– 0.3 to V_{DD} + 0.3	V
Output Current High	I_{OH}	One I/O port active	– 15	mA
		All I/O ports active	– 30	
Output Current Low	I_{OL}	One I/O port active	+ 30 (Peak value)	mA
			+ 15 (1)	
		All I/O Ports active except P1	+ 100 (Peak value)	
			+ 60 (1)	
Operating Temperature	T_A	—	– 40 to + 85	°C
Storage Temperature	T_{stg}	—	– 65 to + 150	°C

NOTE: The values for Output Current Low (I_{OL}) are calculated as Peak Value $\times \sqrt{\text{Duty}}$.

Table 47. D.C. Electrical Characteristics

(T_A = – 40 °C to + 85 °C, V_{DD} = 2.7 V to 6.0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Input High Voltage	V_{IH1}	All input pins except those specified below for V_{IH2} and V_{IH3}	0.7 V_{DD}	—	V_{DD}	V
	V_{IH2}	P0.2, P1, P4.0, P4.1, P4.2, P6, RESET, and CE	0.8 V_{DD}		V_{DD}	
	V_{IH3}	X_{IN}, X_{OUT}, and XT_{IN}	V_{DD} – 0.5		V_{DD}	
Input Low Voltage	V_{IL1}	All pins except those specified below for V_{IL2} and V_{IL3}	0	—	0.3 V_{DD}	V
	V_{IL2}	P0.2, P1, P4.0, P4.1, P4.2, P6, RESET, and CE			0.2 V_{DD}	
	V_{IL3}	X_{IN}, X_{OUT}, and XT_{IN}			0.4	

Table 47. D.C. Electrical Characteristics (Continued)

(T_A = – 40 °C to + 85 °C, V_{DD} = 2.7 V to 6.0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Output High Voltage	V_{OH}	V_{DD} = 4.5 V to 6.0 V I_{OH} = – 1.6 mA Ports 0, 2, 3, 6, EO, and P7–P13	V_{DD} – 0.5	—	—	V
Output Low Voltage	V_{OL}	V_{DD} = 4.5 V to 6.0 V I_{OL} = 1.6 mA Ports 0, 2–6, EO, and P7–P13	—	—	0.4	V
		I_{OL} = 400 µA Ports 0, 2–6, EO, and P7–P13		—	0.2	
Input High Leakage Current	I_{LIH1}	$V_I = V_{DD}$ All input pins except those specified below for I_{LIH2}	—	—	3	µA
	I_{LIH2}	$V_I = V_{DD}$ X_{IN} and XT_{IN} only			20	
Input Low Leakage Current	I_{LIL1}	V_I = 0 V All input pins except X_{IN}, XT_{IN}, and RESET	—	—	– 3	µA
	I_{LIL2}	V_I = 0 V X_{IN}, and XT_{IN} only			– 20	
Output High Leakage Current	I_{LOH}	$V_O = V_{DD}$ All output pins.	—	—	3	µA
Output Low Leakage Current	I_{LOL}	V_O = 0 V	—	—	– 3	µA
Pull-Up Resistor	R_{L1}	V_I = 0 V; V_{DD} = 5 V ± 10% Ports 0, 1 (except P1.3), 2, 3, 4, 5	15	46	80	KΩ
		V_{DD} = 3 V ± 10%	30	90	200	
	R_{L2}	V_I = 0 V; V_{DD} = 5 V ± 10% RESET	100	230	400	
		V_I = 0 V; V = 3 V ± 10% RESET	200	490	800	

Table 47. D.C. Electrical Characteristics (Continued)

(T_A = – 40 °C to + 85 °C, V_{DD} = 2.7 V to 6.0 V)

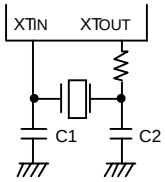
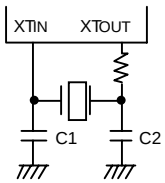
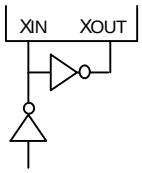
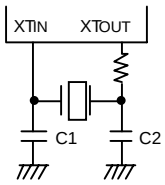
Parameter	Symbol	Conditions	Min	Typ	Max	Units
Supply Current (1)	I_{DD1} (2)	V_{DD} = 5 V ± 10% (3) 4.5 MHz crystal oscillator C1 = C2 = 22 pF f_{in} = 150 MHz V_{in} = 0.5 V_{DD} CE high; PLL operates	—	12	20	mA
	I_{DD2} (2)	Idle mode; V_{DD} = 5 V ± 10% 4.5 MHz crystal oscillator C1 = C2 = 22 pF CPU clock = fxx/4 CE low; PLL stops	—	1.4	1.8	
		V_{DD} = 3 V ± 10% CPU clock = fxx/64		0.23	1.0	
	I_{DD3} (4)	V_{DD} = 3 V ± 10% 32 kHz crystal oscillator CE low; PLL stops	—	25	120	µA
	I_{DD4} (4)	Idle mode; V_{DD} = 3 V ± 10% 32 kHz crystal oscillator		20	30	
	I_{DD5}	Stop 1 mode; XT_{IN} = 0 V V_{DD} = 5 V ± 10% CE low; PLL stops	—	0.6	5	
		V_{DD} = 3 V ± 10%		0.2	3	
	I_{DD6}	V_{DD} = 5 V ± 10 % 4.5 MHz crystal oscillator CPU clock = fxx/4 CE low; PLL stops	—	4.2	8	mA
		V_{DD} = 3 V ± 10 % CPU clock = fxx/64		0.7	1.2	
	I_{DD7}	Stop 2 mode; V_{DD} = 5 V ± 10% CE low; PLL stops	—	0.12	2.0	µA
		V_{DD} = 3 V ± 10% CPU clock = fxx/64		0.05	1.0	

NOTES :

1. The currents in the following circuits are not included: on-chip pull-up resistors, output port drive currents, internal LCD voltage dividing resistors and A/D converter. (I_{DD1} and I_{DD7} are guaranteed in TA = -20 °C to + 85 °C)
2. Data includes power consumption for subsystem clock oscillation.
3. For high-speed controller operation, the power control register (PCON) must be set to 0011B.
4. When the system clock control register, SCMOD, is set to 1001B, main system clock oscillation stops and the subsystem clock is used.

Table 48. Main System Clock Oscillator Characteristics

(T_A = – 40 °C + 85 °C, V_{DD} = 2.7 V to 6.0 V)

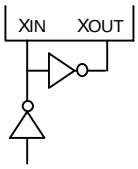
Oscillator	Clock Configuration	Parameter	Test Condition	Min	Typ	Max	Units
Ceramic Oscillator		Oscillation frequency (1)	–	0.4	–	5.0	MHz
		Stabilization time (2)	Stabilization occurs when V_{DD} is equal to the minimum oscillator voltage range.	–	–	4	ms
Crystal Oscillator		Oscillation frequency (1)	–	0.4	4.5	5.0	MHz
		Stabilization time (2)	V_{DD} = 4.5 V to 6.0 V	–	–	10	ms
			V_{DD} = 2.7 V to 4.5 V	–	–	30	
External Clock		X_{IN} input frequency (1)	-–	0.4	–	4.5	MHz
		X_{IN} input high and low level width (t_{XH}, t_{XL})	–	111	–	1250	ns
Crystal Oscillator		Oscillation frequency (1)	—	32	32.768	35	kHz
		Stabilization time (2)	V_{DD} = 4.5 V to 6.0 V	—	1.0	2	s
			V_{DD} = 2.7 V to 4.5 V	—	—	10	

NOTES:

1. Oscillation frequency and X_{in} input frequency data are for oscillator characteristics only.
2. Stabilization time is the interval required for oscillating stabilization after a power-on occurs, or when stop mode is terminated.

Table 49. Subsystem Clock Oscillator Characteristics

(T_A = – 40 °C + 85 °C, V_{DD} = 2.7 V to 6.0 V)

Oscillator	Clock Configuration	Parameter	Test Condition	Min	Typ	Max	Units
External Clock		XT_{IN} input frequency (1)	—	32	—	100	kHz
		XT_{IN} input high and low level width (t_{XH}, t_{XL})	—	5	—	15	µs

NOTES:
1. Oscillation frequency and XT_{IN} input frequency data are for oscillator characteristics only.
2. Stabilization time is the interval required for oscillating stabilization after a power-on occurs.

RECOMMENDED OSCILLATOR CONSTANTS

Main System Clock: Ceramic Resonator (Ta = –40 °C – 85 °C)

Manufacturer	Product Name	Load Cap (pF)		Oscillator Voltage Range (V)		Remarks
		C1	C2	MIN	MAX	
TDK	FCR4.19MC5	–	–	2.7	6.0	On-chip Capacitor: 30 pF±20%, Leaded Type
	FCR4.19M5	33	33	2.7	6.0	Leaded Type
	CCR4.19MC3	–	–	2.7	6.0	On-chip Capacitor: 36 pF±20%, SMD Type

Table 50. Input/Output Capacitance

(T_A = 25 °C, V_{DD} = 0 V)

Parameter	Symbol	Condition	Min	Typ	Max	Units
Input Capacitance	C_{IN}	f = 1 MHz; Unmeasured pins are returned to V_{SS}	—	—	15	pF
Output Capacitance	C_{OUT}		—	—	15	pF
I/O Capacitance	C_{IO}		—	—	15	pF

Table 51. A.C. Electrical Characteristics

(T_A = – 40 °C to + 85 °C, V_{DD} = 2.7 V to 6.0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Instruction Cycle Time (1)	t_{CY}	V_{DD} = 4.5 V to 6.0 V	0.89	—	64	µs
		V_{DD} = 2.7 V to 4.5 V	3.8		64	
		With subsystem clock (fxt)	114	122	125	
TCL0, TCL1 Input Frequency	f_{TI0}, f_{TI1}	V_{DD} = 4.5 V to 6.0 V	0	—	1	MHz
		V_{DD} = 2.7 V to 4.5V			275	kHz
TCL0, TCL1 Input High, Low Width	t_{TIH0}, t_{TIL0} t_{TIH1}, t_{TIL1}	V_{DD} = 4.5 V to 6.0 V	0.48	—	—	µs
		V_{DD} = 2.7 V to 4.5 V	1.8			
SCK Cycle Time	t_{KCY}	V_{DD} = 4.5 V to 6.0 V External SCK source	800	—	—	ns
		Internal SCK source	950			
		V_{DD} = 2.7 V to 4.5 V External SCK source	3200			
		Internal SCK source	3800			
SCK High, Low Width	t_{KH}, t_{KL}	V_{DD} = 4.5 V to 6.0 V External SCK source	400	—	—	ns
		Internal SCK source	$t_{KCY}/2 - 50$			
		V_{DD} = 2.7 V to 4.5 V External SCK source	1600			
		Internal SCK source	$t_{KCY}/2 - 150$			
SI Setup Time to SCK High	t_{SIK}	External SCK source	100	—	—	ns
		Internal SCK source	150			
SI Hold Time to SCK High	t_{KSI}	External SCK source	400	—	—	ns
		Internal SCK source	400			
Output Delay for SCK to SO	t_{KSO}	V_{DD} = 4.5 V to 6.0 V External SCK source	—	—	300	ns
		Internal SCK source			250	
		V_{DD} = 2.7 V to 4.5 V External SCK source			1000	
		Internal SCK source			1000	

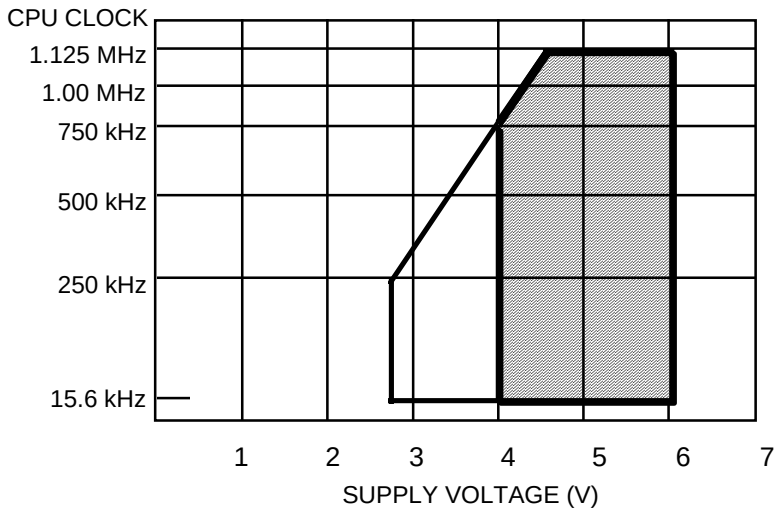
NOTE: Unless otherwise specified, Instruction Cycle Time condition values assume a main system clock (fx) source.

Table 51. A.C. Electrical Characteristics (Continued)

(T_A = – 40 °C to + 85 °C, V_{DD} = 2.7 V to 6.0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Interrupt Input High, Low Width	t_{INTH}, t_{INTL}	INT0	(See Note)	–	–	µs
		INT1, INT2, INT4, KS0–KS3	10			
RESET Input Low Width	t_{RSL}	Input	0.5	–	–	
Operating Frequency	f_{VCOAM}	V_{COAM} mode, sine wave input; V_{IN} = 0.3 V_{P-P}	0.5	–	30	MHz
	f_{VCOFM}	V_{COFM} mode, sine wave input; V_{IN} = 0.3 V_{P-P}	30	–	150	
	f_{AMIF}	AMIF mode, sine wave input V_{IN} = 0.3 V_{P-P}	0.1	–	1.0	
	f_{FMIF}	FMIF mode, sine wave input V_{IN} = 0.3 V_{P-P}	5	–	15	
AD Converting Resolution	–	–	–	–	8	bits
Total Conversion Error	–	T_a = –10 to 50 ° C	–	± 1	± 1.5	LSB

NOTE: The minimum value for INT0 is based on a clock of $2t_{CY}$ or 128 / fx as assigned by the IMOD register setting.



CPU CLOCK = 1/n × oscillator frequency (n = 4, 8, 64)

When PLL operates, operating voltage range is 4.0 V to 6.0 V and only 4.5 MHz of oscillating frequency should be used.

Figure 71. Operating Voltage Range

Table 52. RAM Data Retention Supply Voltage in Stop Mode

(T_A = – 40 °C to + 85 °C)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Data retention supply voltage	V_{DDDR}	–	2.0	–	6.0	V
Data retention supply current	I_{DDDR}	V_{DDDR} = 2.0 V	–	0.1	10	µA
Release signal set time	t_{SREL}	–	0	–	–	µs
Oscillator stabilization wait time (1)	t_{WAIT}	Released by RESET	–	2^{17} / fx	–	ms
		Released by interrupt	–	(2)	–	

NOTES:
1. During oscillator stabilization wait time, all CPU operations must be stopped to avoid instability during oscillator start-up.
2. Use the basic timer mode register (BMOD) interval timer to delay execution of CPU instructions during the wait time.

TIMING WAVEFORMS

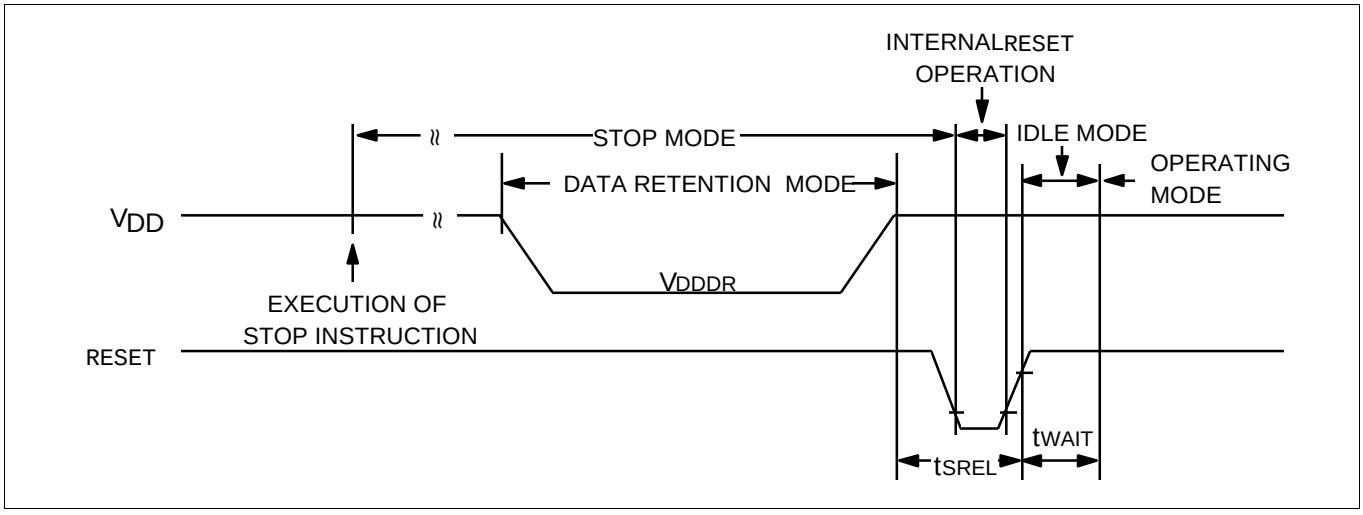


Figure 72. Stop Mode Release Timing When Initiated By RESET

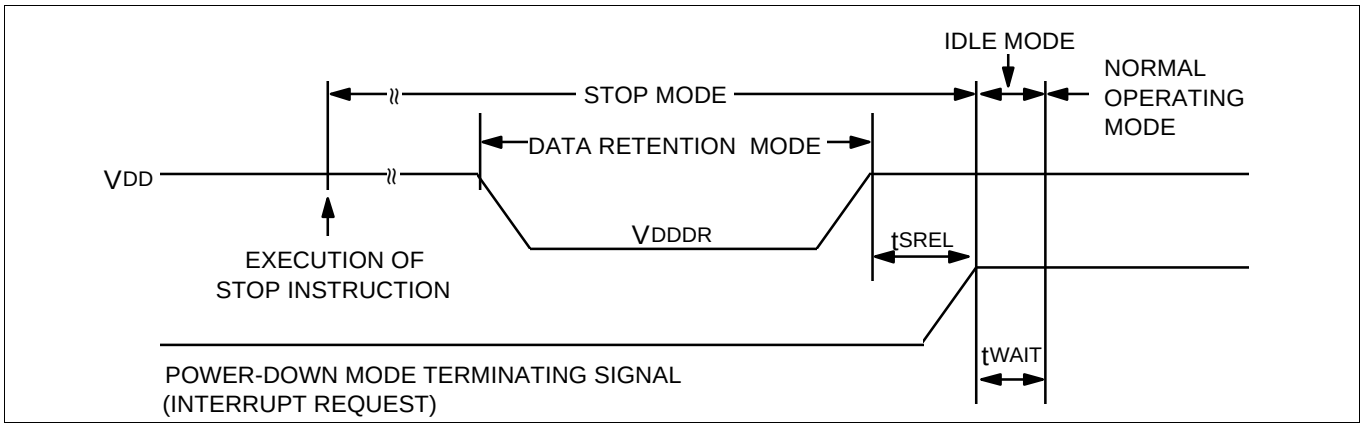


Figure 73. Stop Mode Release Timing When Initiated By Interrupt Request

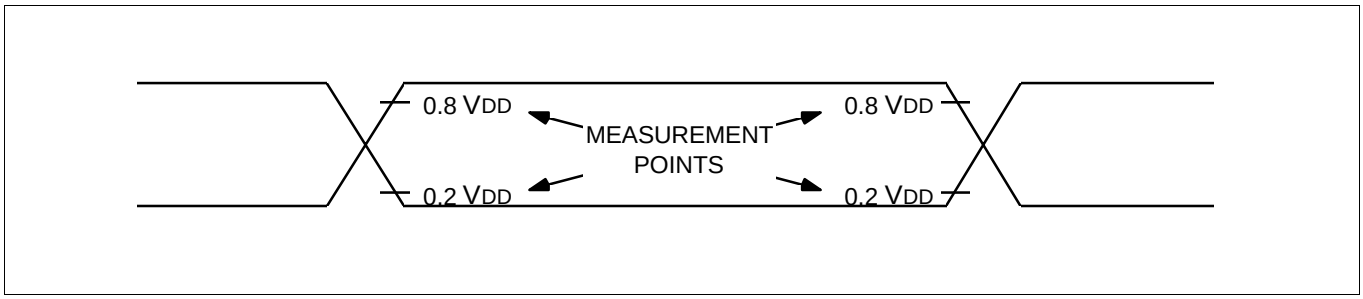


Figure 74. A.C. Timing Measurement Points (Except for X_{IN} and XT_{IN})

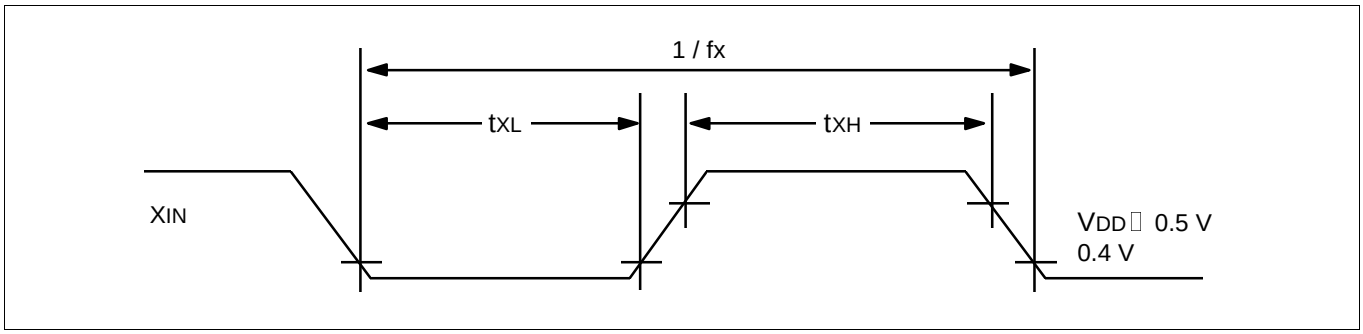


Figure 75. Clock Timing Measurement at X_{IN}

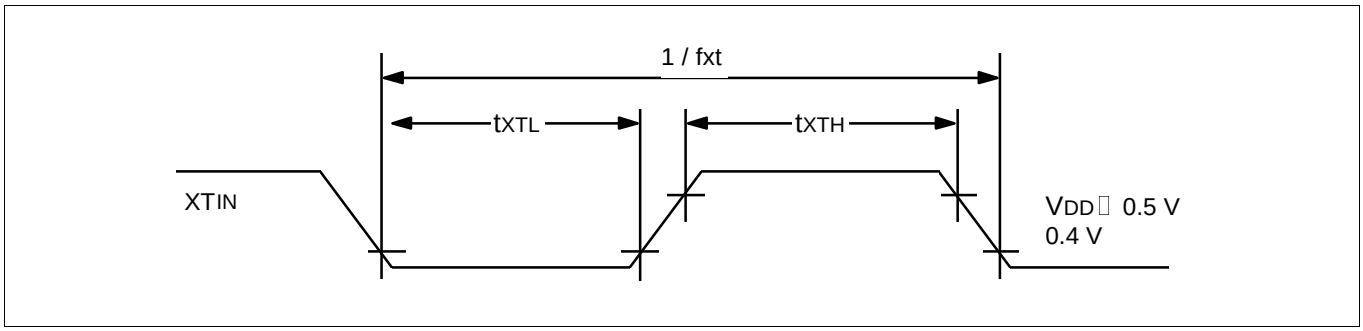


Figure 76. Clock Timing Measurement at XT_{IN}

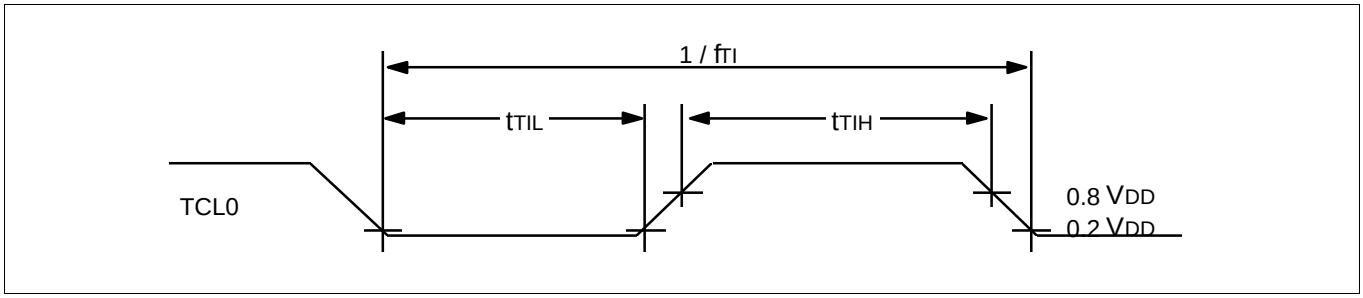


Figure 77. TCL0 Timing

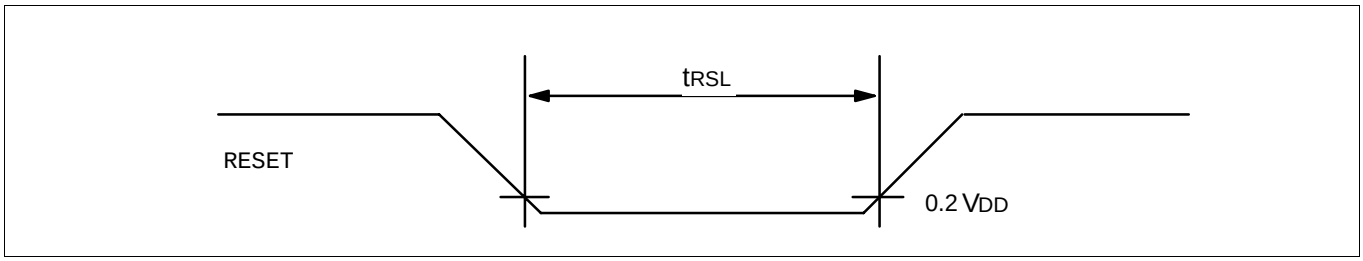


Figure 78. Input Timing for RESET

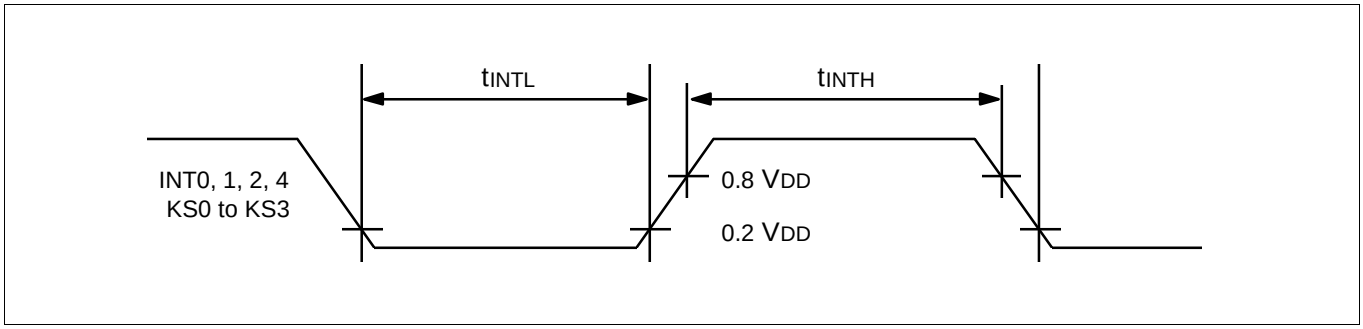


Figure 79. Input Timing for External Interrupts and Quasi-Interrupts

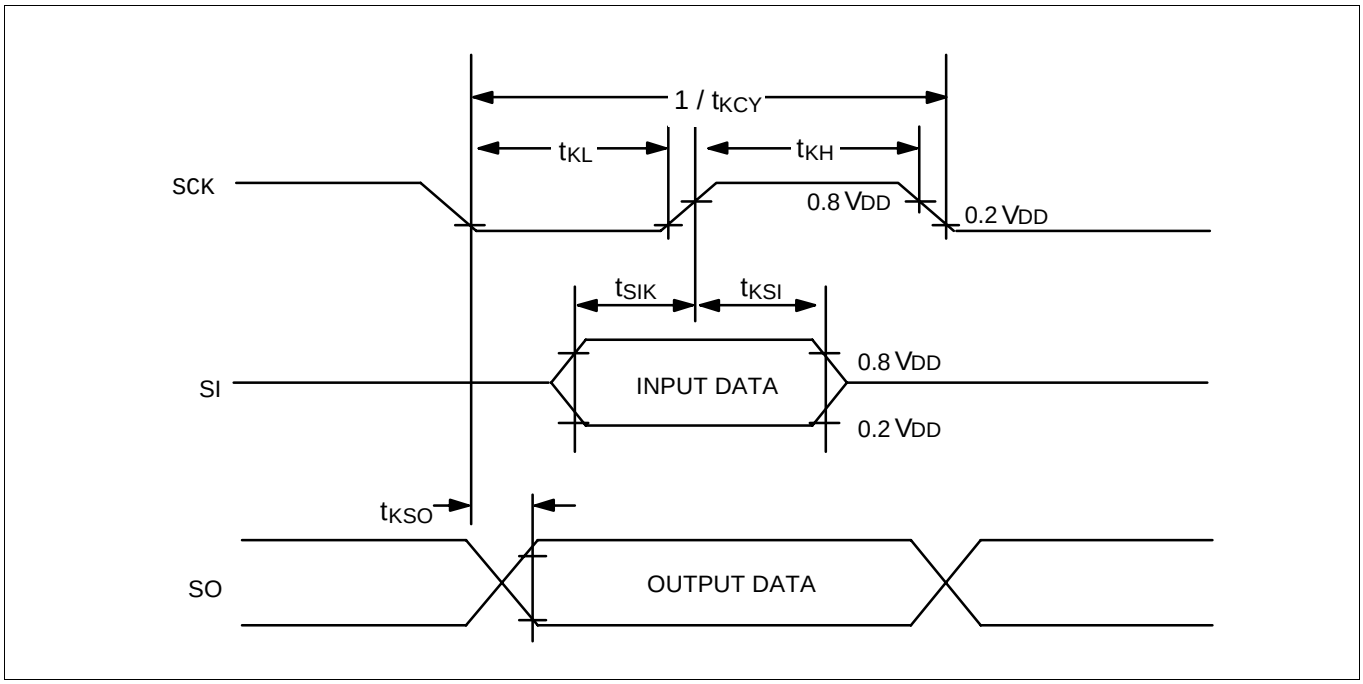


Figure 80. Serial Data Transfer Timing

CHARACTERISTIC CURVES

NOTE

The characteristic values shown in the following graphics are based on actual test measurements. They do not, however, represent guaranteed operating values.

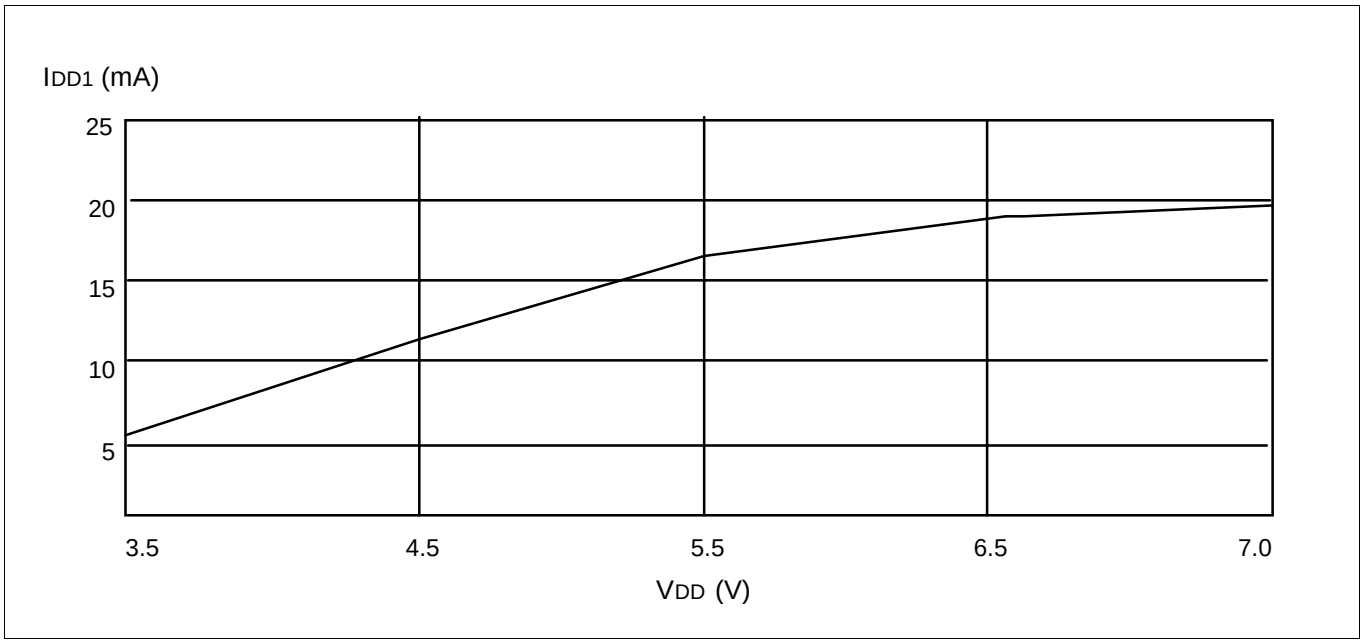


Figure 81. I_{DD1} VS. V_{DD}

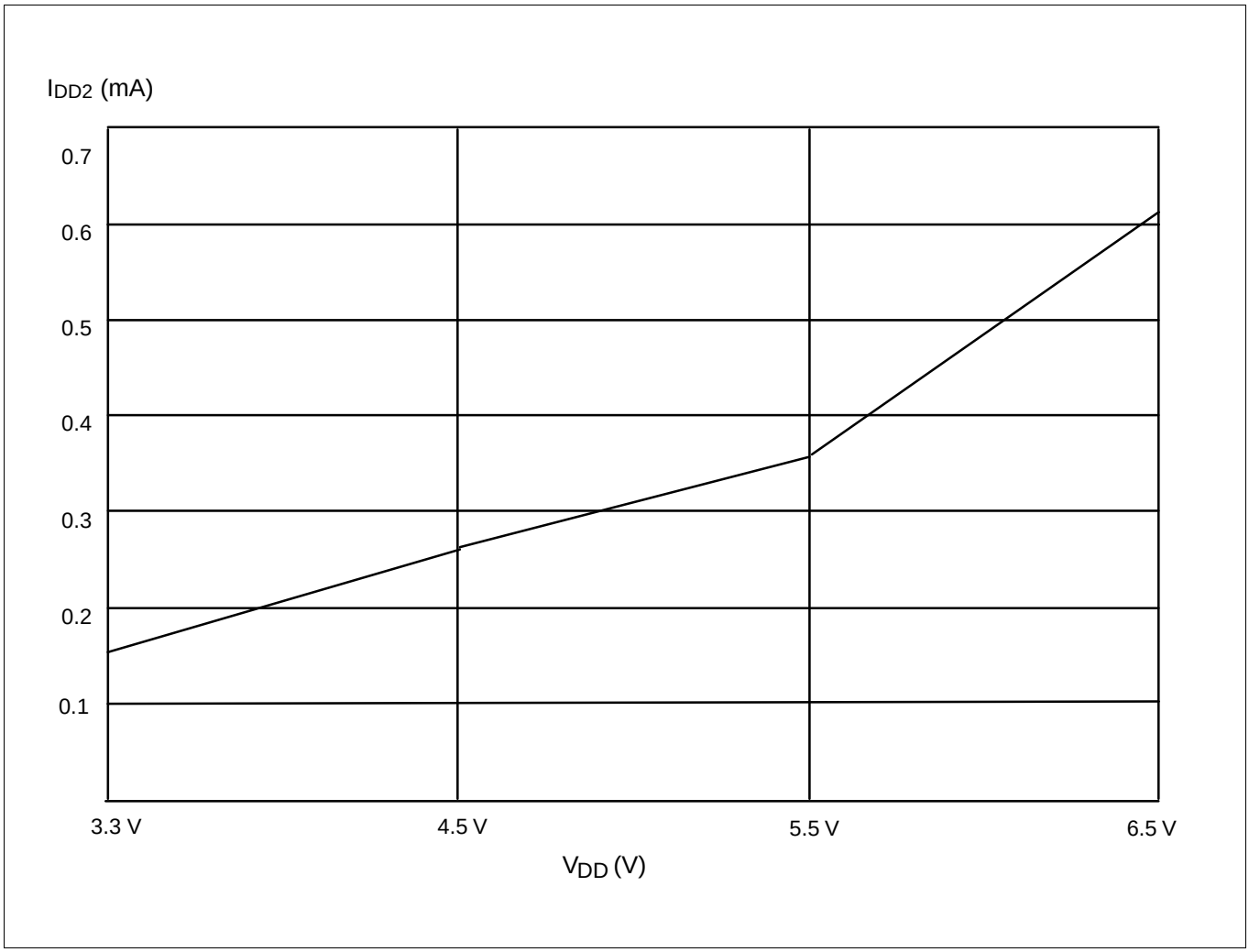


Figure 82. I_{DD2} VS. V_{DD}

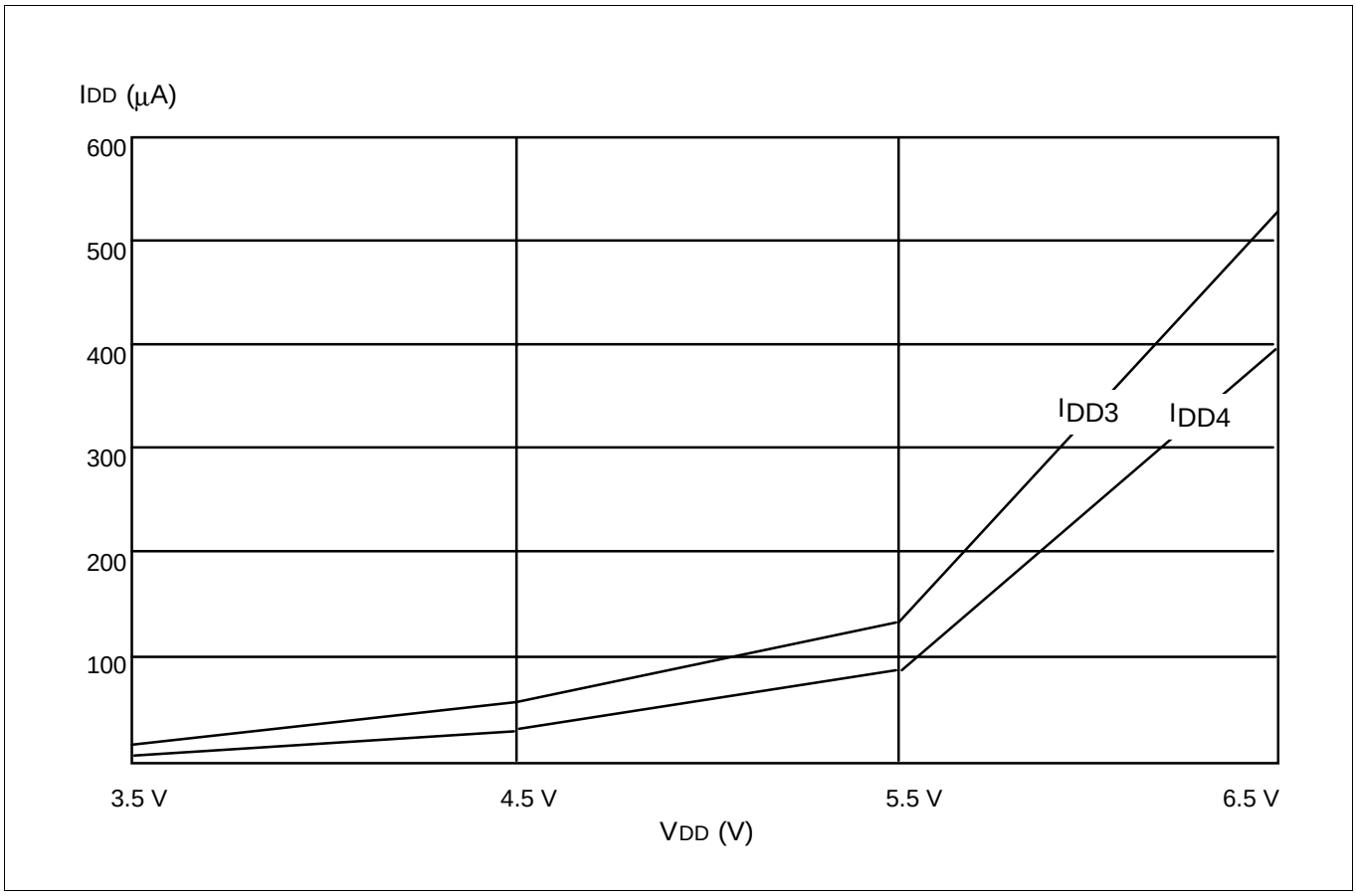


Figure 83. I_{DD3} and I_{DD4} VS V_{DD}

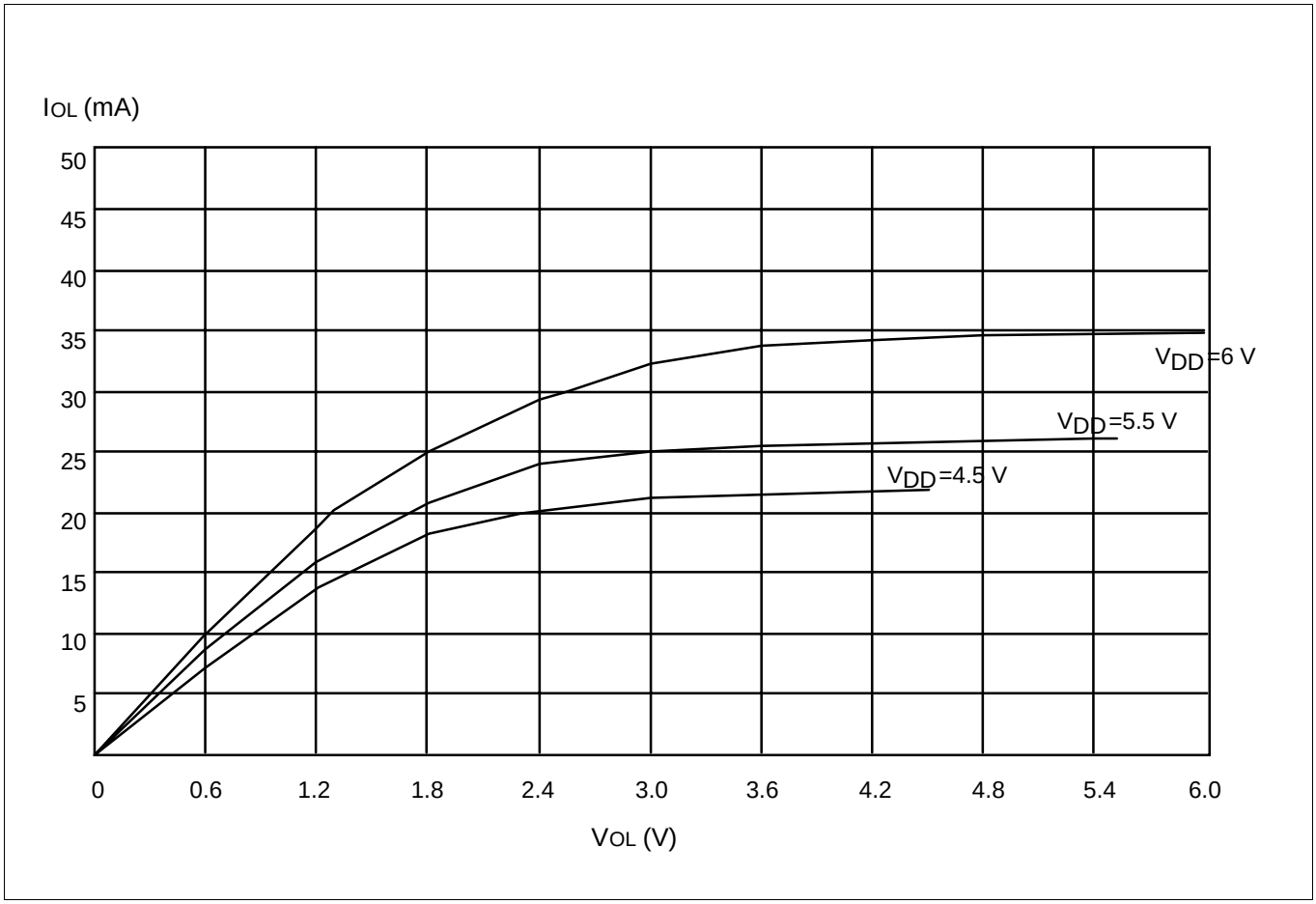


Figure 84. I_{OL} VS. V_{OL} (Ports 0, 2–6)

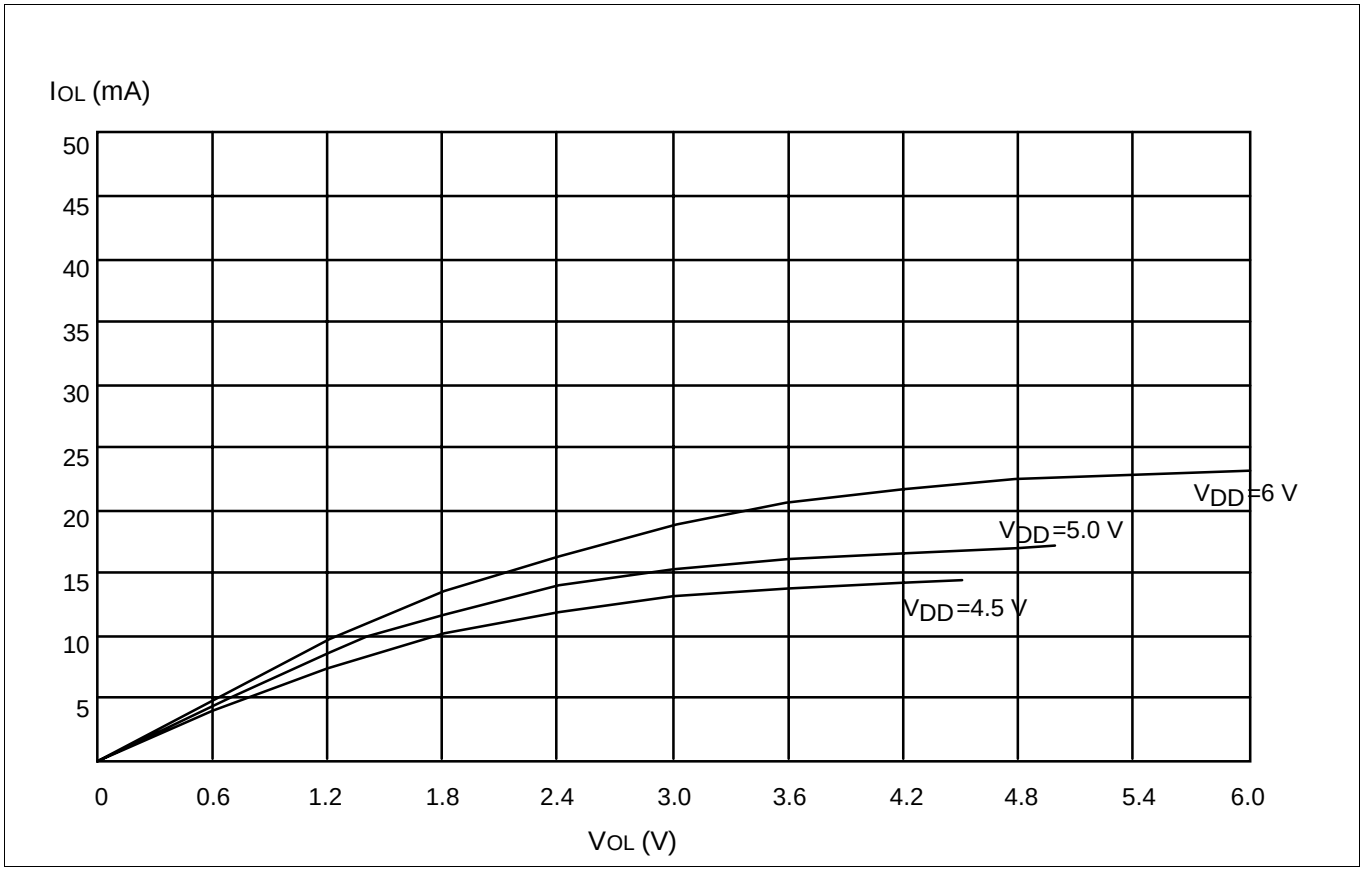


Figure 85. I_{OL} VS. V_{OL} (Ports 7–13)

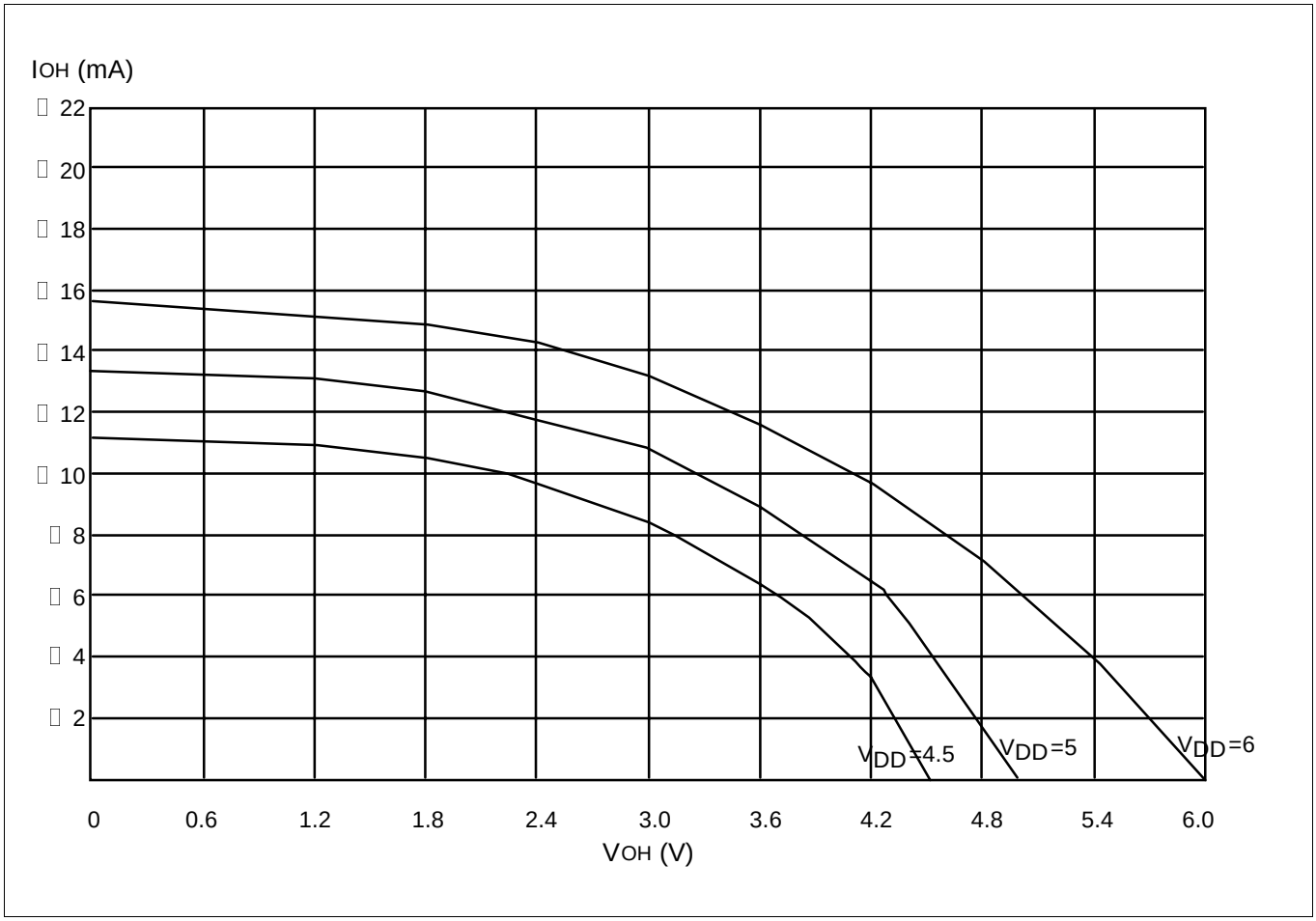


Figure 86. I_{OH} VS. V_{OH} (Ports 0, 2–6)

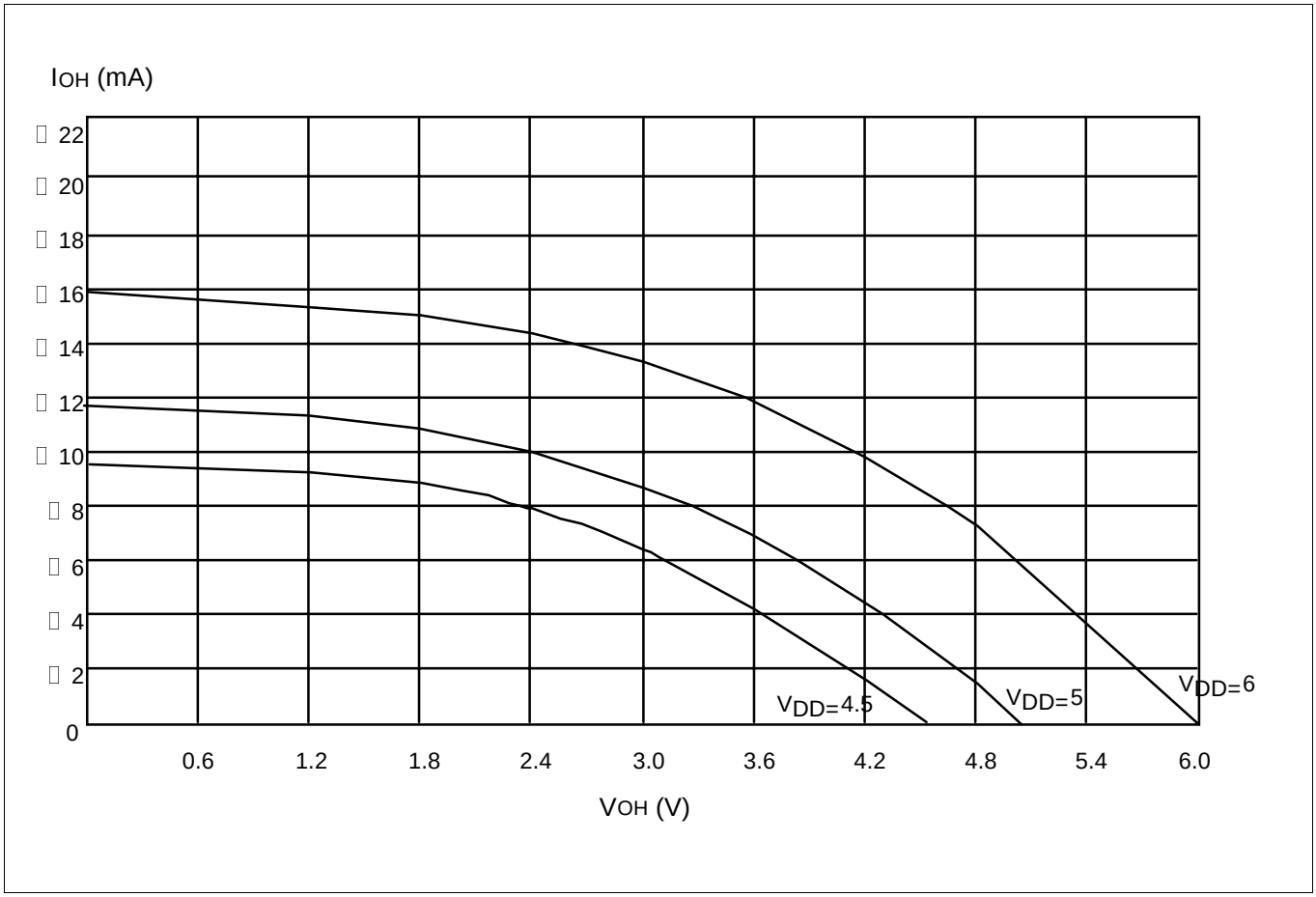


Figure 87. I_{OH} VS. V_{OH} (Ports 7–13)

NOTES