



# KS57C3016

## 4-BIT CMOS Microcontroller

### Product Specification

## OVERVIEW

The KS57C3016 single-chip CMOS microcontroller is designed for very high performance using Samsung's newest 4-bit development approach, SAM4 (Samsung Arrangeable Microcontrollers). With an up-to-16-digit LCD direct drive capability, 4-channel A/D converter, 8-bit timer/counter, PLL frequency synthesizer, and 6-channel PWM outputs, the KS57C3016 offers you an excellent design solution for a wide variety of applications, especially those requiring DTS support.

Up to 55 pins of the 100-pin QFP package can be dedicated to I/O. Eight vectored interrupts provide fast response to internal and external events. In addition, the KS57C3016's advanced CMOS technology ensures low power consumption and a wide operating voltage range.

## FEATURES

### Memory

- 1,024 × 4-bit RAM
- 16,384 × 8-bit ROM

### 55 I/O Pins

- Input only: 4 pins
- Output only: 12 pins for bit output or LCD seg output
- I/O: 23 pins (16 I/O pins are n-channel, open-drain)

### LCD Controller/Driver

- Maximum 16-digit LCD direct drive capability
- 32 segment × 4 common
- Display modes: static, 1/2duty (1/2 bias)
- 1/3 duty (1/2 or 1/3 bias), 1/4duty (1/3 bias)

### 8-Bit Basic Timer

- Four interval timer functions

### 8-Bit Timer/Counter

- Programmable 8-bit timer

- External event counter
- Arbitrary clock output
- External clock signal divider
- Serial I/O clock generator

### Watch Timer

- Time interval generation: 0.5s, 3.9 ms at 32768 Hz
- Four frequency outputs to BUZ pin
- Clock 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/external clock source

### A/D Converter

- 4 channels (8-bit resolution)
- 17.8  $\mu$ s conversion speed at 4.5 MHz

### PLL Frequency Synthesizer

- AM input: maximum 30 MHz
- FM input: maximum 200 MHz

### Pulse Width Modulator

- 6-channel (8-bit resolution)

### Intermediate Frequency (IF) Counter

- 16-bit binary counter
- Divide-by-2 and gate control

### Interrupts

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

### Bit Sequential Carrier

- Supports 16-bit serial data transfer in arbitrary format

**Memory-Mapped I/O Structure**

- Data memory bank 15

**Four Power-Down Modes**

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

**Oscillation Sources**

- Crystal or ceramic for main clock
- Crystal for subsystem clock
- Main system clock frequency: 4.5 MHz (typical)
- Subsystem clock frequency: 32.768 kHz
- CPU clock divider (4, 6, 64)

**Instruction Execution Times**

- 0.9, 1.8, 14.2  $\mu$ s at 4.5 MHz
- 122  $\mu$ s at 32.768 kHz

**Operating Temperature Range**

- -40 °C to 85 °C

**Operating Voltage Range**

- 2.7 V to 6.0 V

**Package Type**

- 100-pin QFP

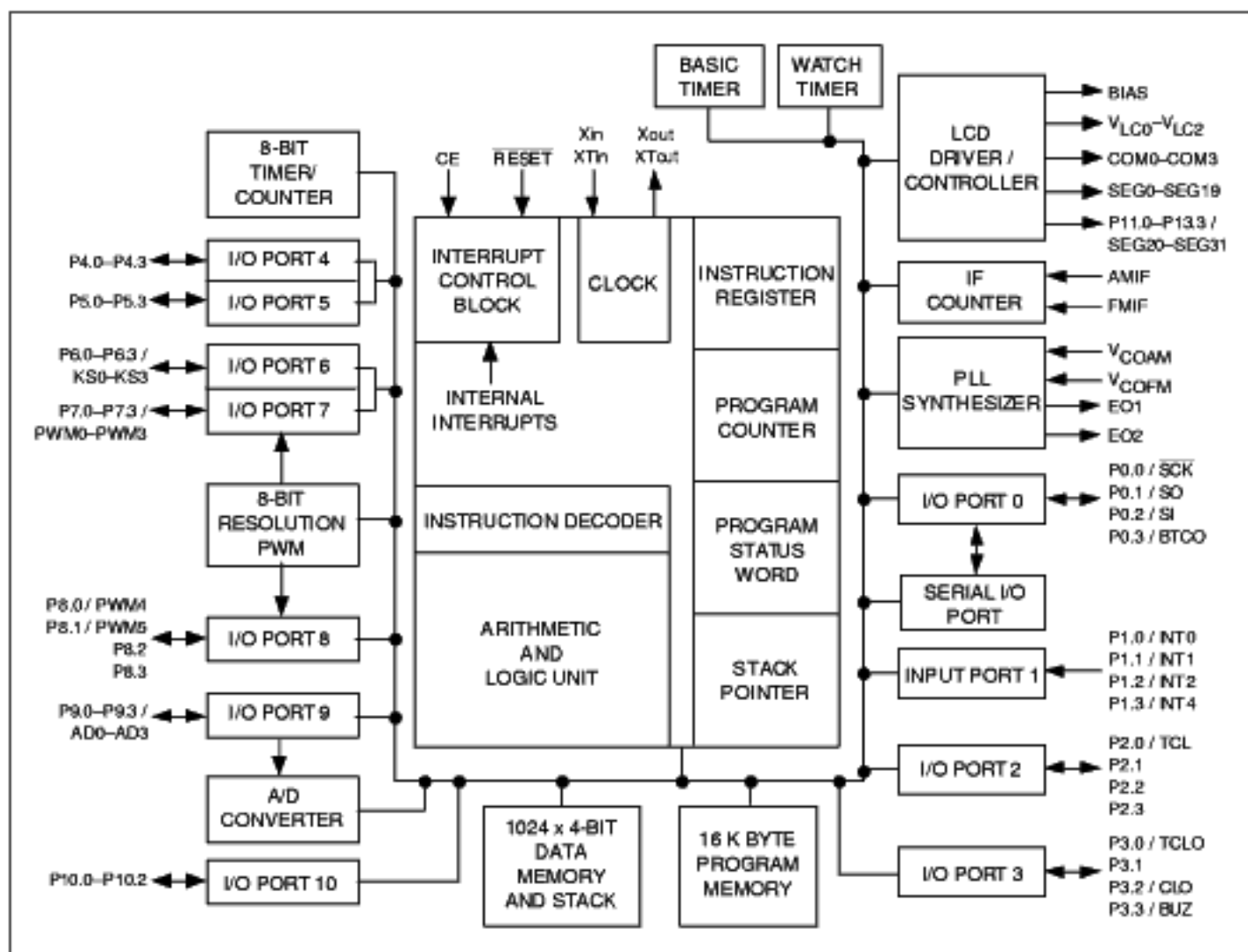


Figure 1. KS57C3016 Block Diagram

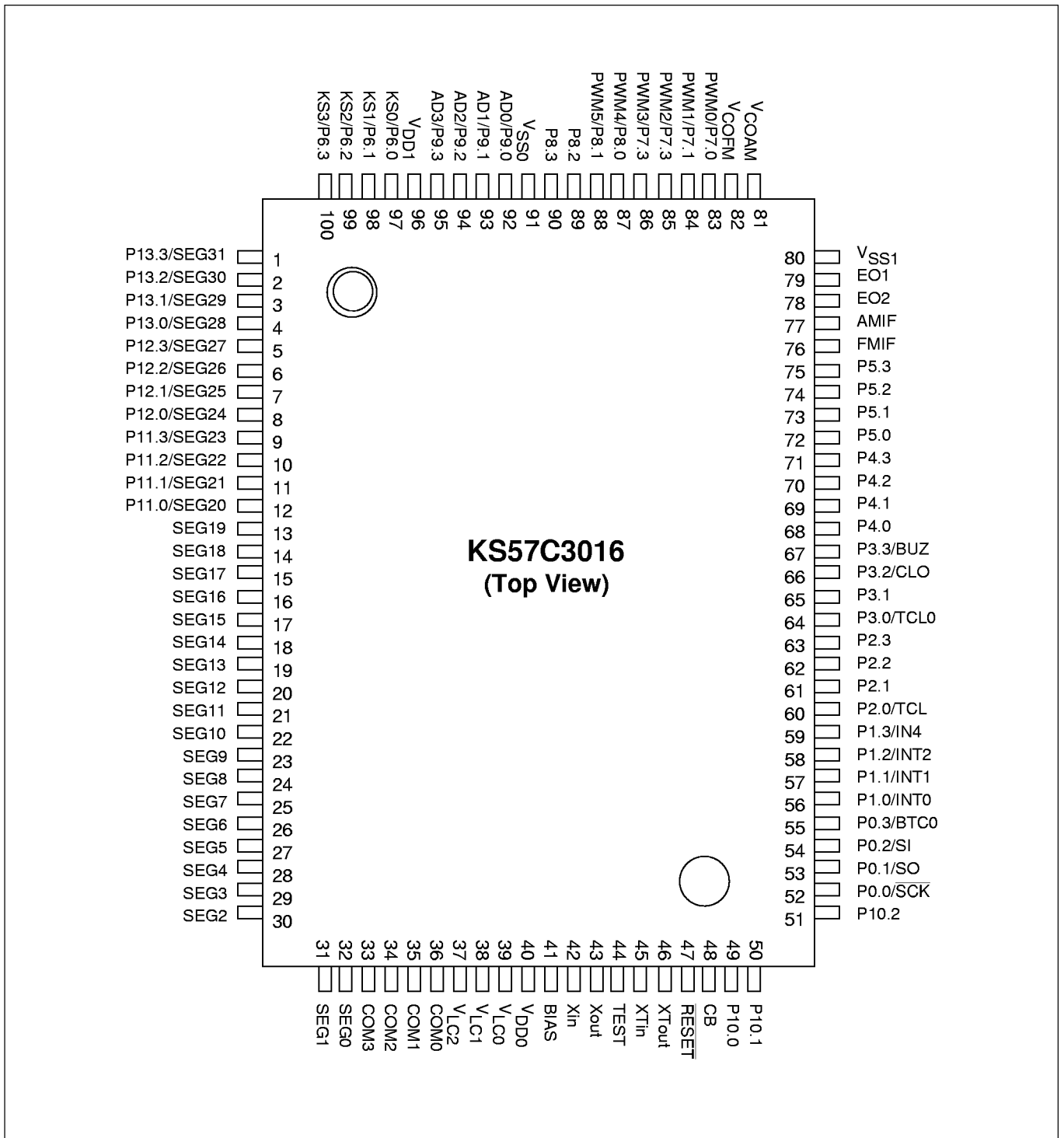


Figure 2. KS57C3016 Pin Assignments (100-QFP)

Table 1. KS57C3016 Pin Descriptions

| Pin Name                               | Pin Type | Description  | Number               | Share Pin                      |
|--|----------|--|----------------------|--------------------------------|
| P0.0<br>P0.1<br>P0.2<br>P0.3           | I/O      | 4-bit I/O port.<br>1-bit and 4-bit read/write and test is possible.<br>Individual pins are software configurable as input or output.<br>4-bit pull-up resistors are software assignable to input pins; pull-up resistors are automatically disabled for output pins. | 52<br>53<br>54<br>55 | SCK<br>SO<br>SI<br>BTCO        |
| P1.0<br>P1.1<br>P1.2<br>P1.3           | I        | 4-bit input port.<br>1-bit and 4-bit read and test is possible.<br>3-bit pull-up resistors are assignable by software to pins P1.0, P1.1, and P1.2.  | 56<br>57<br>58<br>59 | INT0<br>INT1<br>INT2<br>INT4   |
| P2.0<br>P2.1<br>P2.2<br>P2.3           | I/O      | Same as port 0   | 60<br>61<br>62<br>63 | TCL0                           |
| P3.0<br>P3.1<br>P3.2<br>P3.3           | I/O      | Same as port 0   | 64<br>65<br>66<br>67 | TCLO0<br>—<br>CLO<br>BUZ       |
| P4.0–P4.3<br>P5.0–P5.3                 | I/O      | 4-bit I/O ports.<br>N-channel open-drain output up to 9volts.<br>1- and 4-bit read/write and test is possible.<br>Ports 4 and 5 can be paired to support 8-bit data transfer.<br>Pull-up resistors are assignable to individual pins by mask option.                 | 68–71<br>72–75       | —                              |
| P6.0–P6.3                              | I/O      | Same as port 0.<br>Ports 6 and 7 can be paired to enable 8-bit data transfer.  | 97–100               | KS0–KS3                        |
| P7.0–P7.3<br>P8.0–P8.3                 | I/O      | N-channel open-drain 4-bit I/O ports up to 9 volts.<br>1-bit and 4-bit read/write and test is possible.<br>Pins are individually software configurable as input or output.<br>Pull-up resistors are assignable to individual pins by mask option.                    | 83–86<br>87–90       | PWM0–<br>PWM3<br>PWM4–<br>PWM5 |
| P9.0–P9.3                              | I/O      | Same as port 0   | 92–95                | AD0–AD3                        |
| P10.0–P10.2                            | I/O      | Same as port 0 except that port 10 is 3-bit I/O port.  | 49–51                | —                              |
| P11.0–11.3<br>P12.0–12.3<br>P13.0–13.3 | O        | Output ports for 1-bit data  | 12–9<br>8–5<br>4–1   | SEG20–<br>SEG31                |
| SCK                                    | I/O      | Serial I/O interface clock signal  | 52                   | P0.0                           |
| SO                                     | I/O      | Serial data output   | 53                   | P0.1                           |
| SI                                     | I/O      | Serial data input  | 54                   | P0.2                           |

Table 1. KS57C3016 Pin Descriptions (Continued)

| Pin Name                              | Pin Type | Description  | Number | Share Pin   |
|---------------------------------------|----------|--|--------|-------------|
| BTCO                                  | I/O      | Basic timer clock output   | 55     | P0.3        |
| INT0, INT1                            | I        | External interrupts. The triggering edge for INT0 and INT1 is selectable.                                    | 56–57  | P1.0, P1.1  |
| INT2                                  | I        | Quasi-interrupt with detection of rising edges   | 58     | P1.2        |
| INT4                                  | I        | External interrupt with detection of rising and falling edges  | 59     | P1.3        |
| TCL0                                  | I/O      | External clock input for timer/counter 0   | 60     | P2.0        |
| TCLO0                                 | I/O      | Timer/counter 0 clock output   | 64     | P3.0        |
| CLO                                   | I/O      | Clock output   | 66     | P3.2        |
| BUZ                                   | I/O      | 2.2 kHz, 4.4 kHz, 8.8 kHz, or 17.6 kHz frequency output at 4.5 MHz for buzzer sound                          | 67     | P3.3        |
| KS0–KS3                               | I/O      | Quasi-interrupt inputs with falling edge detection   | 97–100 | P6.0–P6.3   |
| PWM0–PWM5                             | I/O      | PWM outputs  | 83–88  | P7.0–P8.1   |
| AD0–AD3                               | I/O      | A/D converter analog inputs  | 92–95  | P9.0–P9.3   |
| SEG0–SEG19                            | O        | LCD segment data outputs   | 32–13  | —           |
| SEG20–SEG31                           | O        | LCD segment data outputs   | 12–1   | P11.0–P13.3 |
| COM0–COM3                             | O        | LCD common signal outputs  | 36–34  | —           |
| CE                                    | I        | Input pin for checking device power. High level during normal operation; low level when PLL operation stops. | 48     | —           |
| EO1–EO2                               | O        | Output for PLL error data  | 78–79  | —           |
| V <sub>COAM</sub> , V <sub>COFM</sub> | I        | External V <sub>COAM</sub> and V <sub>COFM</sub> inputs  | 81–82  | —           |
| AMIF, FMIF                            | I        | Intermediate AM/FM frequency input   | 77–76  | —           |
| TEST                                  | I        | Test signal input (must be connected to V <sub>SS</sub> )  | 44     | —           |
| V <sub>DD0</sub>                      | —        | Main power supply  | 40     | —           |
| V <sub>SS0</sub>                      | —        | Main ground  | 91     | —           |
| V <sub>DD1</sub>                      | —        | Power supply for PLL prescaler   | 96     | —           |
| V <sub>SS1</sub>                      | —        | PLL prescaler ground   | 80     | —           |
| RESET                                 | I        | Reset signal   | 47     | —           |
| BIAS                                  | —        | LCD power control  | 41     | —           |
| V <sub>LC0</sub> –V <sub>LC2</sub>    | —        | LCD power supply. Voltage dividing resistors are assignable by mask option.                                  | 39–37  | —           |
| X <sub>in</sub> , X <sub>out</sub>    | —        | Crystal, ceramic, or RC oscillator signal for main system clock.   | 42, 43 | —           |
| X <sub>Tin</sub> , X <sub>Tout</sub>  | —        | Crystal oscillator signal for subsystem clock.   | 45, 46 | —           |

Table 2. Supplemental KS57C3016 Pin Data

| Pin Numbers    | Pin Names                             | Share Pins            | I/O Type | Reset Value | Circuit Type |
|----------------|---------------------------------------|-----------------------|----------|-------------|--------------|
| 52–55          | P0.0–P0.3                             | SCK, SO, SI,<br>BTCO  | I/O      | Input       | 6            |
| 56–59          | P1.0–P1.3                             | INT0, INT1, INT2      | I        | Input       | 3            |
| 59             | P1.3                                  | INT4                  | I        | Input       | 2            |
| 60–63          | P2.0–P2.3                             | TCL0                  | I/O      | Input       | 6            |
| 64–67          | P3.0–P3.3                             | TCLO0, —, CLO,<br>BUZ | I/O      | Input       | 5            |
| 68–71          | P4.0–P4.3                             | —                     | I/O      | Note        | 10           |
| 72–75          | P5.0–P5.3                             | —                     | I/O      | Note        | 10           |
| 97–100         | P6.0–P6.3                             | KS0–KS3               | I/O      | Input       | 6            |
| 83–86          | P7.0–P7.3                             | PWM0–PWM3             | I/O      | Note        | 10           |
| 87–88<br>89–90 | P8.0–P8.1<br>P8.2–P8.3                | PWM4–PWM5<br>—        | I/O      | Note        | 10           |
| 92–95          | P9.0–P9.3                             | AD0–AD3               | I/O      | Input       | 11           |
| 49–51          | P10.0–P10.2                           | —                     | I/O      | Input       | 5            |
| 9–12           | P11.3–P11.0                           | SEG23–SEG20           | O        | Low         | 9            |
| 5–8            | P12.3–P12.0                           | SEG27–SEG24           | O        | Low         | 9            |
| 1–4            | P13.3–P13.0                           | SEG31–SEG28           | O        | Low         | 9            |
| 32–16          | SEG0–SEG15                            | —                     | O        | Low         | 9            |
| 15–13          | SEG16–SEG19                           | —                     | O        | Low         | 7            |
| 36–33          | COM0–COM3                             | —                     | O        | Low         | 8            |
| 39–37          | V <sub>LC0</sub> –V <sub>LC2</sub>    | —                     | —        | —           | —            |
| 41             | BIAS                                  | —                     | —        | —           | —            |
| 76, 77         | FMIF, AMIF                            | —                     | I        | Input       | 12           |
| 78, 79         | EO2, EO1                              | —                     | O        | Output      | —            |
| 81, 82         | V <sub>COAM</sub> , V <sub>COFM</sub> | —                     | I        | Input       | 12           |
| 48             | CE                                    | —                     | I        | —           | —            |
| 42, 43         | X <sub>in</sub> , X <sub>out</sub>    | —                     | —        | —           | —            |
| 45, 46         | X <sub>Tin</sub> , X <sub>Tout</sub>  | —                     | —        | —           | —            |
| 47             | RESET                                 | —                     | I        | —           | 13           |

**NOTE:** When pull-up resistors are provided, high level; when pull-up resistors are not provided, high impedance.

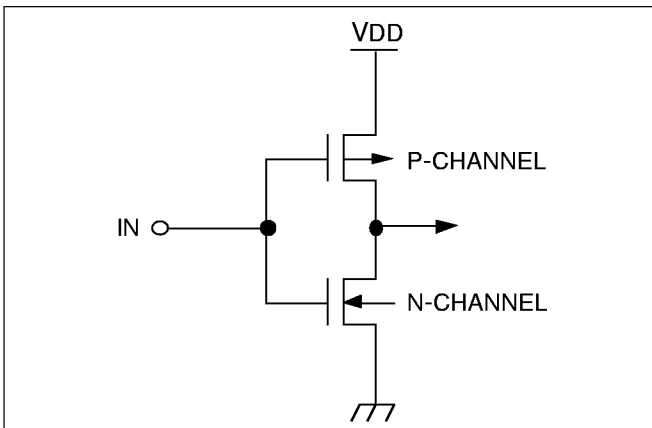


Figure 3. Pin Circuit Type 1

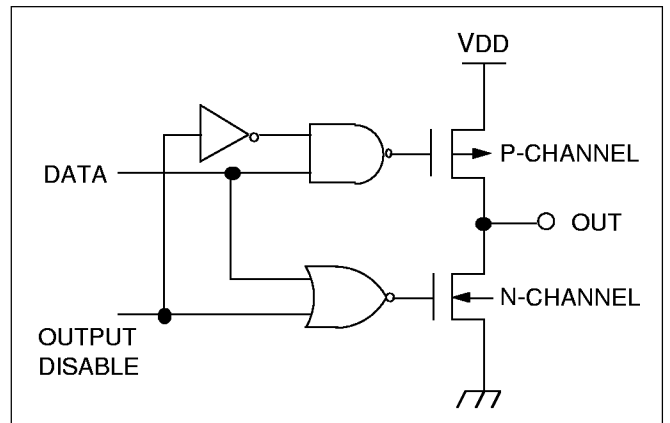


Figure 6. Pin Circuit Type 4

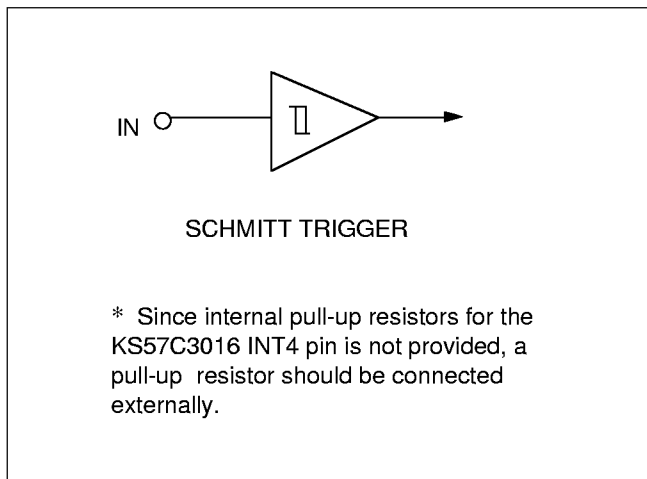


Figure 4. Pin Circuit Type 2

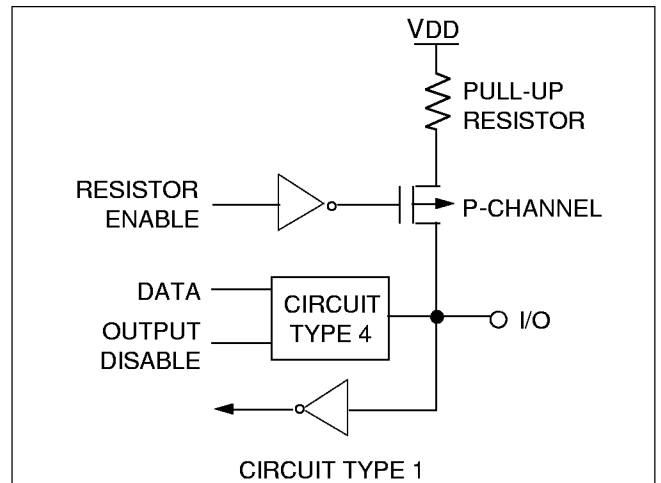


Figure 7. Pin Circuit Type 5

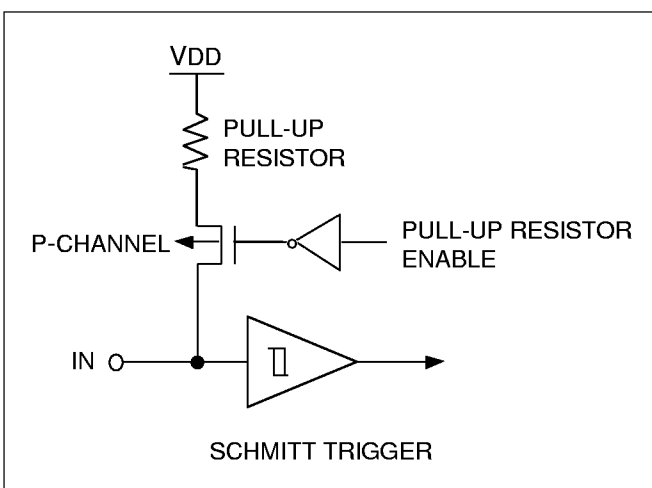


Figure 5. Pin Circuit Type 3

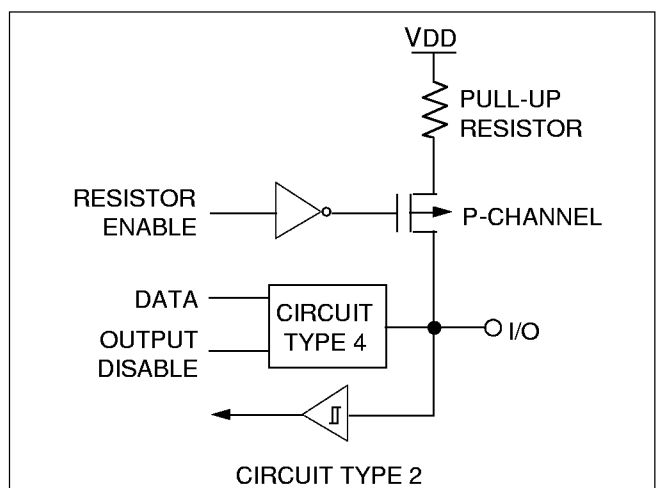


Figure 8. Pin Circuit Type 6

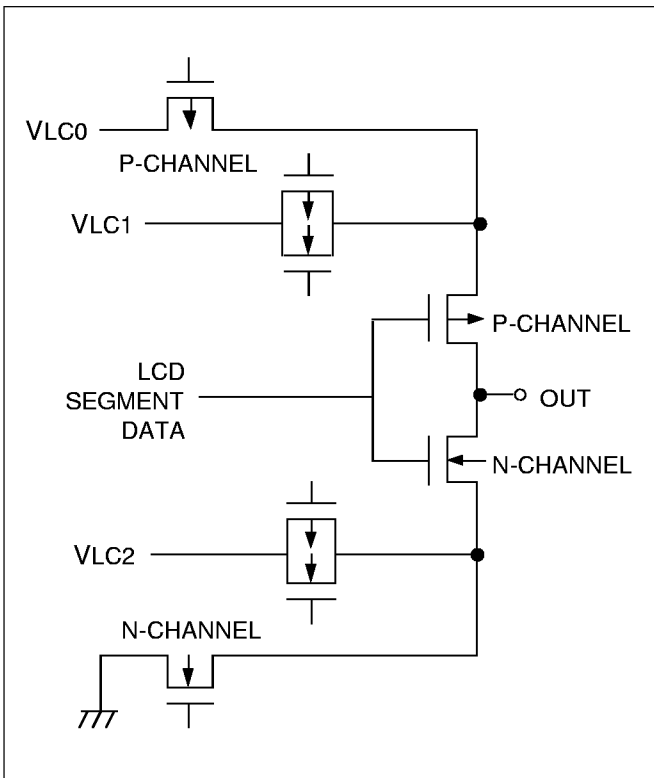


Figure 9. Pin Circuit Type 7

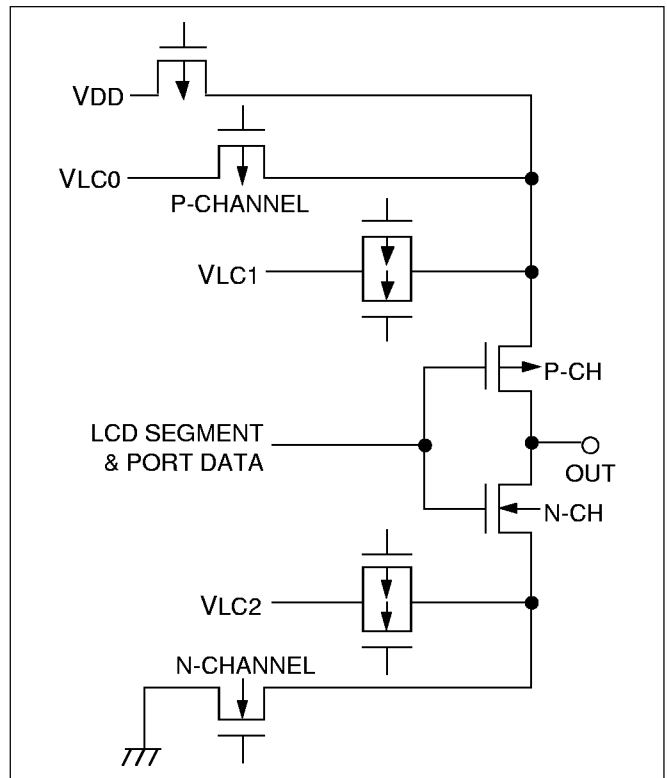


Figure 11. Pin Circuit Type 9

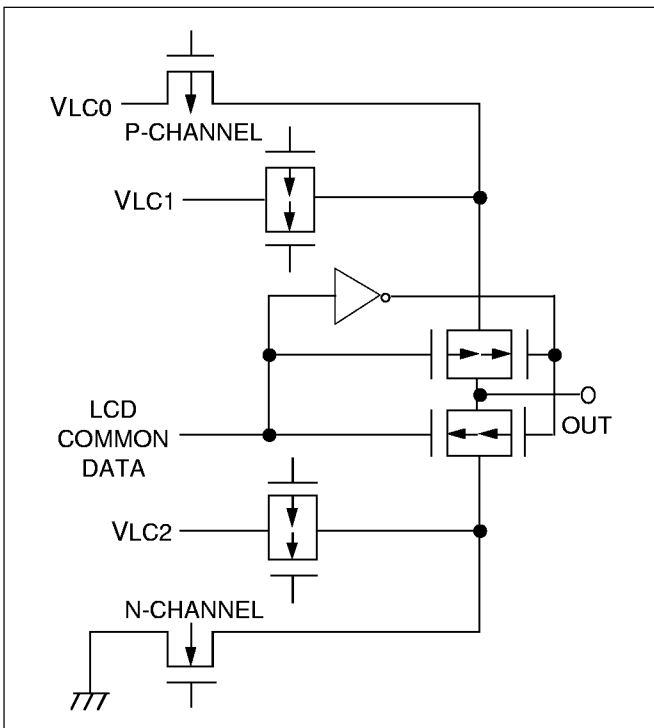


Figure 10. Pin Circuit Type 8

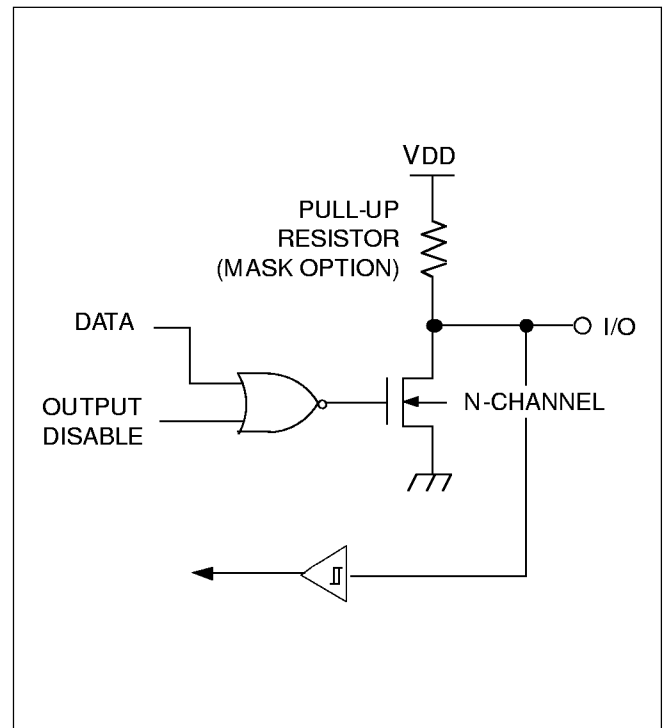


Figure 12. Pin Circuit Type 10



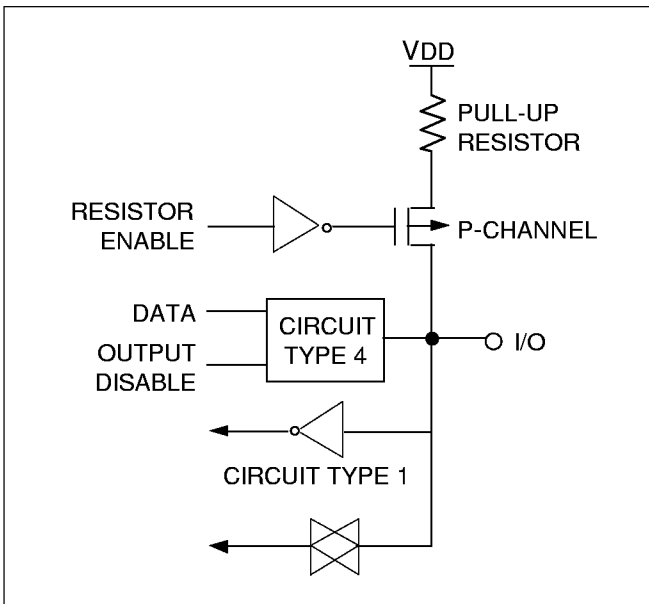


Figure 13. Pin Circuit Type 11

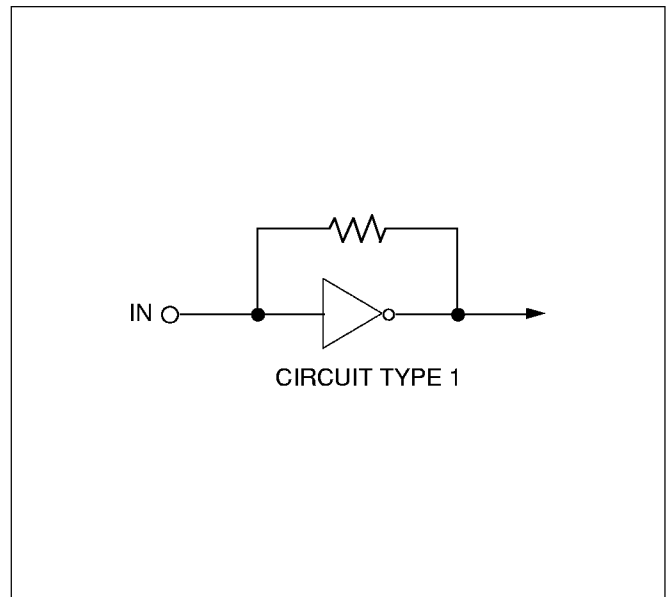


Figure 14. Pin Circuit Type 12

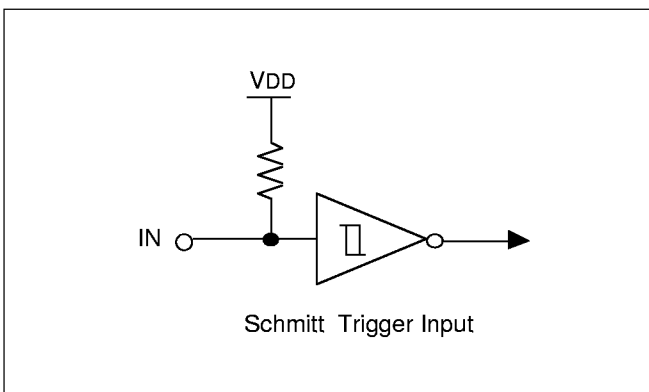


Figure 15. Pin Circuit Type 13

**PROGRAM MEMORY (ROM)**

ROM maps for KS57C3016 devices are mask programmable at the factory. In its standard configuration, the 16,384 × 8-bit program memory has four areas that are directly addressable by the program counter (PC):

- 16-byte area for vector addresses
- 96-byte instruction reference area
- 16-byte general-purpose area
- 16,256-byte general-purpose area

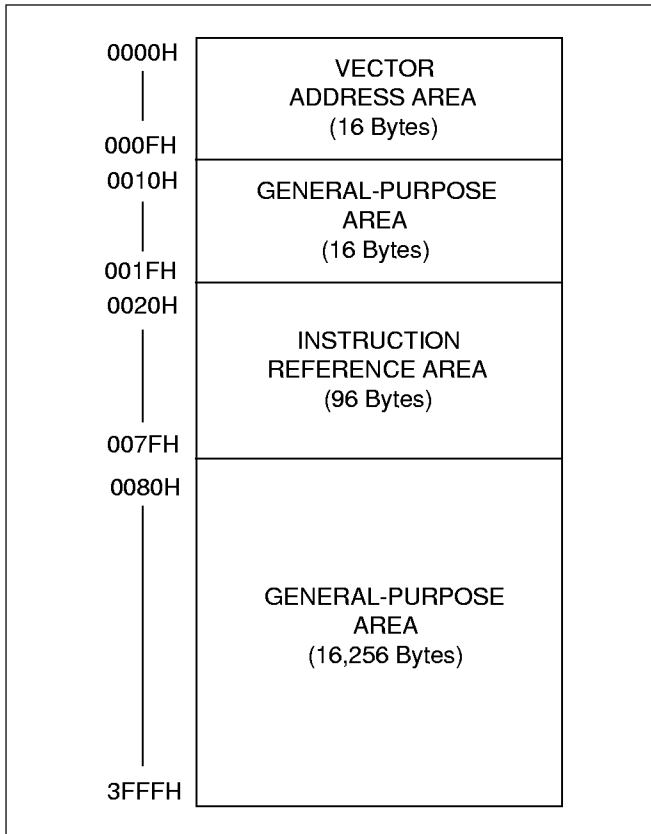


Figure 16. ROM Map

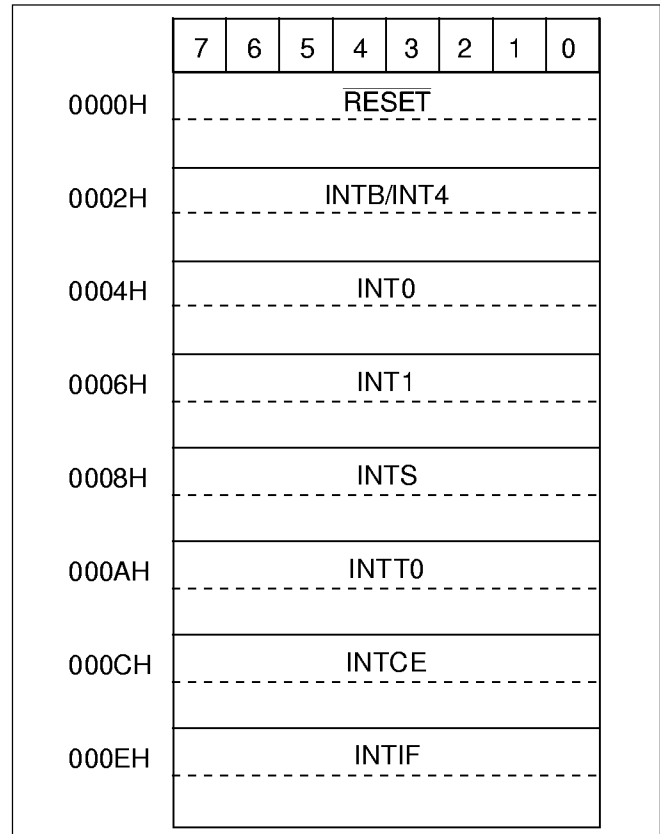


Figure 17. Vector Address Map

**DATA MEMORY (RAM)**

In its standard configuration, the 1024 × 4-bit data memory has seven areas:

- 32 × 4-bit working register area
- 224 × 4-bit general-purpose area in bank 0 which is also used as the stack area
- 224 × 4-bit general-purpose area in bank 1
- 32 × 4-bit area for LCD data in bank 1
- 256 × 4-bit general-purpose area in bank 2

- 256 × 4-bit general-purpose area in bank 3
- 128 × 4-bit area in bank 15 for memory-mapped I/O addresses

**I/O MAP FOR HARDWARE REGISTERS**

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

| ADDRESSING MODE |  | DA<br>DA.b |          | @HL<br>@H + DA.b |          | @WX<br>@WL | mema.b | memb.@L |  |
|-----------------|--|------------|----------|------------------|----------|------------|--------|---------|--|
|                 |  | EMB = 0    | EMB = 1  | EMB = 0          | EMB = 1  | X          | X      | X       |  |
| RAM AREAS       |  |            |          |                  |          |            |        |         |  |
| 000H            | WORKING REGISTERS                          |            |          |                  |          |            |        |         |  |
| 01FH<br>020H    |  |            |          |                  |          |            |        |         |  |
| 07FH<br>080H    | BANK 0<br>(GENERAL REGISTERS AND STACK)    |            | SMB = 0  |                  | SMB = 0  |            |        |         |  |
| 0FFH            |  |            |          |                  |          |            |        |         |  |
| 100H            | BANK 1<br>(GENERAL REGISTERS)              |            | SMB = 1  |                  | SMB = 1  |            |        |         |  |
| 1D7H            |  |            |          |                  |          |            |        |         |  |
| 1D8H            | BANK 1<br>(DISPLAY REGISTERS)              |            | SMB = 1  |                  | SMB = 1  |            |        |         |  |
| 1FFH            |  |            |          |                  |          |            |        |         |  |
| 200H            | BANK 2<br>(GENERAL REGISTERS)              |            | SMB = 1  |                  | SMB = 1  |            |        |         |  |
| 2FFH            |  |            |          |                  |          |            |        |         |  |
| 300H            | BANK 3<br>(GENERAL REGISTERS)              |            | SMB = 1  |                  | SMB = 1  |            |        |         |  |
| 3FFH            |  |            |          |                  |          |            |        |         |  |
| F80H            | BANK 15<br>(PERIPHERAL HARDWARE REGISTERS) |            | SMB = 15 |                  | SMB = 15 | FB0H       |        |         |  |
|                 |  |            |          |                  |          |            | FBFH   |         |  |
|                 |  |            |          |                  |          |            | FC0H   |         |  |
| FFFH            |  |            |          |                  |          | FF0H       |        |         |  |

**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 18. Data Memory (RAM) Address Structure

Table 3. I/O Map for Memory Bank 15

| Memory Bank 15 |          |                                    |       |      |       | Addressing Mode |        |       |       |
|----------------|----------|------------------------------------|-------|------|-------|-----------------|--------|-------|-------|
| Address        | Register | Name                               |       |      |       | R/W             | 1-Bit  | 4-Bit | 8-Bit |
| F81H–F80H      | SP       | Stack Pointer                      |       |      |       | R/W             | No     | No    | Yes   |
| F85H           | BMOD     | Basic Timer Mode Register          |       |      |       | W               | .3     | Yes   | No    |
| F87H–F86H      | BCNT     | Basic Timer Counter Register       |       |      |       | R               | No     | No    | Yes   |
| F89H–F88H      | WMOD     | Watch Timer Mode Register          |       |      |       | W               | .3 (1) | No    | Yes   |
| F8DH–F8CH      | LMOD     | LCD Mode Register                  |       |      |       | W               | .3     | No    | Yes   |
| F8EH           | LCON     | LCD Control Register               |       |      |       | W               | No     | Yes   | No    |
| F8FH           | SUBSTP   | Subclock Stop Control Register     |       |      |       | W               | .3     | Yes   | No    |
| F91H–F90H      | TMOD0    | Timer/Counter 0 Mode Register      |       |      |       | W               | .3     | No    | Yes   |
| F92H           |          | "0"                                | TOE0  | BOE  | "0"   | R/W             | Yes    | Yes   | No    |
| F95H–F94H      | TCNT0    | Timer/Counter 0 Counter Register   |       |      |       | R               | No     | No    | Yes   |
| F97H–F96H      | TREF0    | Timer/Counter 0 Reference Reg      |       |      |       | W               | No     | No    | Yes   |
| F99H–F98H      | IFCNT0   | IF Counter 0                       |       |      |       | R               | No     | No    | Yes   |
| F9BH–F9AH      | IFCNT1   | IF Counter1                        |       |      |       | R               | No     | No    | Yes   |
| F9CH           | IFMOD    | IF Counter Mode Register           |       |      |       | R/W             | Yes    | Yes   | No    |
| F9DH           | PLLREG   | PLL Control Register               |       |      |       | R               | Yes    | Yes   | No    |
| FB0H           | PSW      | IS1                                | IS0   | EMB  | ERB   | R/W             | Yes    | Yes   | Yes   |
| FB1H           |          | C (2)                              | SC2   | SC1  | SC0   | R               | No     | No    |       |
| FB2H           | IPR      | Interrupt Priority Register        |       |      |       | W               | IME    | Yes   | No    |
| FB3H           | PCON     | Power Control Register             |       |      |       | W               | No     | Yes   | No    |
| FB4H           | IMOD0    | External Interrupt 0 Mode Register |       |      |       | W               | No     | Yes   | No    |
| FB5H           | IMOD1    | External Interrupt 1 Mode Register |       |      |       | W               | No     | Yes   | No    |
| FB6H           | IMOD2    | External Interrupt 2 Mode Register |       |      |       | W               | No     | Yes   | No    |
| FB7H           | SCMOD    | System Clock Mode Register         |       |      |       | W               | Yes    | No    | No    |
| FB8H           |          | IE4                                | IRQ4  | IEB  | IRQB  | R/W             | Yes    | Yes   | No    |
| FBAH           |          | "0"                                | "0"   | IEW  | IRQW  | R/W             | Yes    | Yes   | No    |
| FBBH           |          | IEIF                               | IRQIF | IECE | IRQCE |                 |        |       |       |
| FBCH           |          | "0"                                | "0"   | IET0 | IRQT0 |                 |        |       |       |
| FBDH           |          | "0"                                | "0"   | IES  | IRQS  |                 |        |       |       |
| FBEH           |          | IE1                                | IRQ1  | IE0  | IRQ0  |                 |        |       |       |
| FBFH           |          | "0"                                | "0"   | IE2  | IRQ2  |                 |        |       |       |
| FC0H           | BSC0     | Bit Sequential Carrier 0           |       |      |       |                 |        |       |       |

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

|           |         |                          |     |     |     |     |
|-----------|---------|--------------------------|-----|-----|-----|-----|
| FC1H      | BSC1    | Bit Sequential Carrier 1 | R/W | Yes | Yes | Yes |
| FC2H      | BSC2    | Bit Sequential Carrier 2 |     |     |     |     |
| FC3H      | BSC3    | Bit Sequential Carrier 3 |     |     |     |     |
| FC4H      | PLLD0   | PLL Data Register 0      | W   | No  | Yes | Yes |
| FC5H      | PLLD1   | PLL Data Register 1      |     |     |     |     |
| FC6H      | PLLD2   | PLL Data Register 2      |     |     |     |     |
| FC7H      | PLLD3   | PLL Data Register 3      |     |     |     |     |
| FC8H      | PLMOD0  | PLL Mode Register 0      | W   | No  | Yes | No  |
| FC9H      | PLMOD1  | PLL Mode Register 1      |     |     |     |     |
| FCBH–FCAH | PWMREF0 | PWM Reference Register 0 | R/W | No  | No  | Yes |
| FCDH–FCCH | PWMREF1 | PWM Reference Register 1 |     |     |     |     |
| FCFH–FCEH | PWMREF2 | PWM Reference Register 2 |     |     |     |     |
| FD0H      | CLMOD   | Clock Mode Register      | W   | No  | Yes | No  |
| FD3H–FD2H | PWMREF3 | PWM Reference Register 3 | R/W | No  | No  | Yes |
| FD5H–FD4H | PWMREF4 | PWM Reference Register 4 | R/W | No  | No  | Yes |
| FD7H–FD6H | PWMREF5 | PWM Reference Register 5 | R/W | No  | No  | Yes |
| FD9H–FD8H | ADATA   | ADC Data Register        | R   | No  | No  | Yes |
| FDAH      | ADMOD   | ADC Mode Register        | R/W | Yes | Yes | No  |
| FDBH      | AFLAG   | ADC Flag Register        | (3) | Yes | Yes | No  |
| FDDH–FDCH | PUMOD1  | Pull-up Mode Register 1  | W   | No  | No  | Yes |
| PDFH–FDEH | PUMOD2  | Pull-up Mode Register 2  |     |     |     |     |
| FE1H–FE0H | SMOD    | SIO Mode Register        | W   | .3  | No  | Yes |
| FE3H–FE2H | PWMOD   | PWM Mode Register        | R/W | .1  | No  | Yes |
| FE5H–FE4H | SBUF    | SIO Buffer Register      | R/W | No  | No  | Yes |
| FE7H–FE6H | PMG0    | Port Mode Group 0        | W   | No  | No  | Yes |
| FE9H–FE8H | PMG1    | Port Mode Group 1        |     |     |     |     |
| FEBH–FEAH | PMG2    | Port Mode Group 2        |     |     |     |     |
| FEDH–FECH | PMG3    | Port Mode Group 3        |     |     |     |     |
| FEFH–FEEH | PMG4    | Port Mode Group 4        |     |     |     |     |
| FF0H      | P0      | Port 0                   | R/W | Yes | Yes | No  |
| FF1H      | P1      | Port 1                   | R   |     |     |     |
| FF2H      | P2      | Port 2                   | R/W |     |     |     |
| FF3H      | P3      | Port 3                   | R/W |     |     | No  |
| FF4H      | P4      | Port 4                   | R/W |     |     |     |
| FF5H      | P5      | Port 5                   | R/W |     |     |     |

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

|      |     |         |     |     |     |     |
|------|-----|---------|-----|-----|-----|-----|
| FF6H | P6  | Port 6  | R/W | Yes | Yes | Yes |
| FF7H | P7  | Port 7  | R/W |     |     | No  |
| FF8H | P8  | Port 8  | R/W |     |     |     |
| FF9H | P9  | Port 9  | R/W |     |     |     |
| FFAH | P10 | Port 10 | R/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.

**BIT SEQUENTIAL CARRIER (BSC)**

The bit sequential carrier (BSC) is a 16-bit general register that is mapped in data memory bank 15. Using the BSC, you can specify sequential addresses and bit locations using 1-bit indirect addressing (memb.@L).

BSC bit addressing is independent of the current EMB value. In this way, programs can process 16-bit data by moving the bit location sequentially and then

incrementing or decrementing the value of the L register.

For 8-bit manipulations, the 4-bit register names BSC0 and BSC2 must be specified and the upper and lower 8 bits manipulated separately. If the values of the L register are 0H at BSC0.@L, the address and bit location assignment is FC0H.0. If the L register content is FH at BSC0.@L, the address and bit location assignment is FC3H.3.

**Table 4. BSC Register Organization**

| Name | Address | Bit 3  | Bit 2  | Bit 1  | Bit 0  |
|------|---------|--------|--------|--------|--------|
| BSC0 | FC0H    | BSC0.3 | BSC0.2 | BSC0.1 | BSC0.0 |
| BSC1 | FC1H    | BSC1.3 | BSC1.2 | BSC1.1 | BSC1.0 |
| BSC2 | FC2H    | BSC2.3 | BSC2.2 | BSC2.1 | BSC2.0 |
| BSC3 | FC3H    | BSC3.3 | BSC3.2 | BSC3.1 | BSC3.0 |

 **PROGRAMMING TIP — Using the BSC Register to Output 16-Bit Data**

To use the bit sequential carrier (BSC) register to output 16-bit data (5937H) to the P3.0 pin:

```

BITS      EMB
SMB       15
LD        EA,#37H      ;
LD        BSC0,EA     ; BSC0 ← A, BSC1 ← E
LD        EA,#59H      ;
LD        BSC2,EA     ; BSC2 ← A, BSC3 ← E
SMB       0
LD        L,#0H       ;
AGN LDB   C,BSC0.@L   ;
LDB       P3.0,C      ; P3.0 ← C
INCS     L
JR        AGN
RET
    
```

### INTERRUPTS

The KS57C3016 has four external interrupts, four internal interrupts and two quasi-interrupts. Table 5 shows the conditions for interrupt generation. 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.

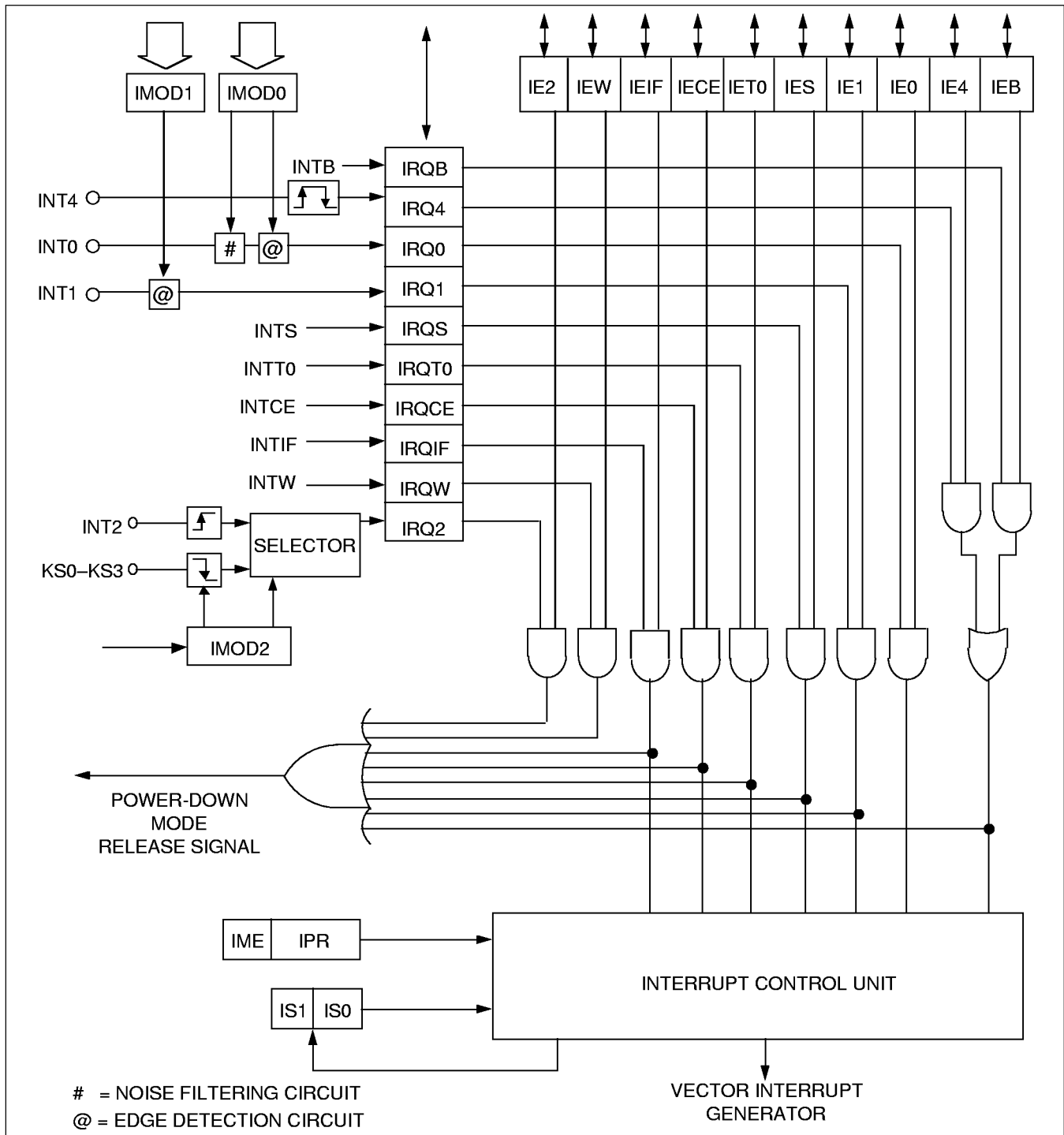


Figure 19. Interrupt Control Circuit Diagram



**Table 5. Interrupt Request Flag Conditions and Priorities**

| Interrupt Source | Internal / External | Condition for IRQx Flag Setting  | Interrupt Priority | Request Flag Name |
|------------------|---------------------|--|--------------------|-------------------|
| INTB             | I                   | Reference time interval signal from basic timer  | 1                  | IRQB              |
| INT4             | E                   | Both rising and falling edges detected at INT4   | 1                  | IRQ4              |
| INT0             | E                   | Rising or falling edge detected at INT0 pin  | 2                  | IRQ0              |
| INT1             | E                   | Rising or falling edge detected at INT1 pin  | 3                  | IRQ1              |
| INTS             | I                   | Completion signal for serial transmit-and-receive or receive-only operation                | 4                  | IRQS              |
| INTT0            | I                   | Signals for TCNT0 and TREF0 registers match  | 5                  | IRQT0             |
| INTCE            | E                   | When falling edge is detected at CE pin  | 6                  | IRQCE             |
| INTIF            | I                   | When gate closes   | 7                  | IRQIF             |
| INT2 *           | E                   | Rising edge detected at INT2 or else a falling edge is detected at any of the KS0–KS3 pins | —                  | IRQ2              |
| INTW             | I                   | Time interval of 0.5 s or 3.19 ms  | —                  | IRQW              |

\* The quasi-interrupt INT2 is only used for testing incoming signals.

**INTERRUPT ENABLE FLAGS (IEx)**

IEx flags, when set to "1", enable specific interrupt requests to be serviced. When the interrupt request flag is set to "1", an interrupt will not be serviced until its corresponding IEx flag is also enabled. The IPR register contains a global disable bit, IME, which disables all interrupt at once.

**Table 6. Interrupt Enable and Request Flag**

| Address | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|-------|-------|-------|-------|
| FB8H    | IE4   | IRQ4  | IEB   | IRQB  |
| FBAH    | 0     | 0     | IEW   | IRQW  |
| FBBH    | IEIF  | IRQIF | IECE  | IRQCE |
| FBCH    | 0     | 0     | IET0  | IRQT0 |
| FBDH    | 0     | 0     | IES   | IRQS  |
| FBEH    | IE1   | IRQ1  | IE0   | IRQ0  |
| FBFH    | 0     | 0     | IE2   | IRQ2  |

**NOTES:**

1. IEx refers to all interrupt enable flags.
2. IRQx refers to all interrupt request flags.
3. IEx = "0" is interrupt disable mode.
4. IEx = "1" is interrupt enable mode.

**INTERRUPT PRIORITY**

Each interrupt source can also be individually programmed to high levels by modifying the IPR register. When IS1 = 0 and IS0 = 1, a low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt.

If you clear the interrupt status flags (IS1 and IS0) to "0" in a interrupt service routine, a high-priority interrupt can be interrupted by low-priority interrupt (multi-level interrupt). Before the IPR can be modified by 4-bit write instructions, all interrupts must first be disabled by a DI instruction.

When all interrupts are low priority (the lower three bits of the IPR register are "0"), the interrupt requested first will have high priority. Therefore, the first-requested interrupt cannot be superseded by any other interrupt.

If two or more interrupt requests are received simultaneously, the priority level is determined according to the standard interrupt priorities, where the default priority is assigned by hardware when the lower three IPR bits = "0".

In this case, the higher-priority interrupt request is serviced and the other interrupt is inhibited. Then, when the high-priority interrupt is returned from its service routine by an IRET instruction, the inhibited service routine is started.

**Table 7. 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 |
| 0     | 0     | 1     | INTB and INT4                                  |
| 0     | 1     | 0     | INT0   |
| 0     | 1     | 1     | INT1   |
| 1     | 0     | 0     | INTS   |
| 1     | 0     | 1     | INTT0  |
| 1     | 1     | 0     | INTCE  |
| 1     | 1     | 1     | INTIF  |

**Table 8. Default Priorities**

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

**EXTERNAL INTERRUPTS**

The external interrupt mode registers IMOD0 and IMOD1 are used to control the triggering edge of the input signal at INT0 and INT1, respectively. The INT4 interrupt is an exception because its input signal generates an interrupt request on both rising and falling edges.

**Table 9. IMOD0 Register Organization (4-Bit W)**

| 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       | 0       | Rising edge detection                  |
|         | 0 | 0       | 1       | Falling edge detection                 |
|         | 0 | 1       | 0       | Both rising and falling edge detection |
|         | 0 | 1       | 1       | IRQ0 flag cannot be set to "1"         |

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. Since 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.

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 its IEx flag.
5. Enable all interrupts with an EI instruction.

**EXTERNAL INTERRUPT MODE REGISTER**

The external interrupt 2 (INT2) mode register, IMOD2, is used to select INT2 and KSn pins as interrupt input. 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 "1" and a release signal for power-down mode is generated.

If one or more of the pins which are configured as key Interrupt (KS0–KS7) are in Low input or Low output state, the key Interrupt can not be occurred.

Table 10. IMOD1 and 2 Register Organization (4-Bit W)

| 0 | 0 | 0 | IMOD1.0 | Effect of IMOD1 Settings |
|---|---|---|---------|--------------------------|
| 0 | 0 | 0 | 0       | Rising edge detection    |
| 0 | 0 | 0 | 1       | Falling edge detection   |

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

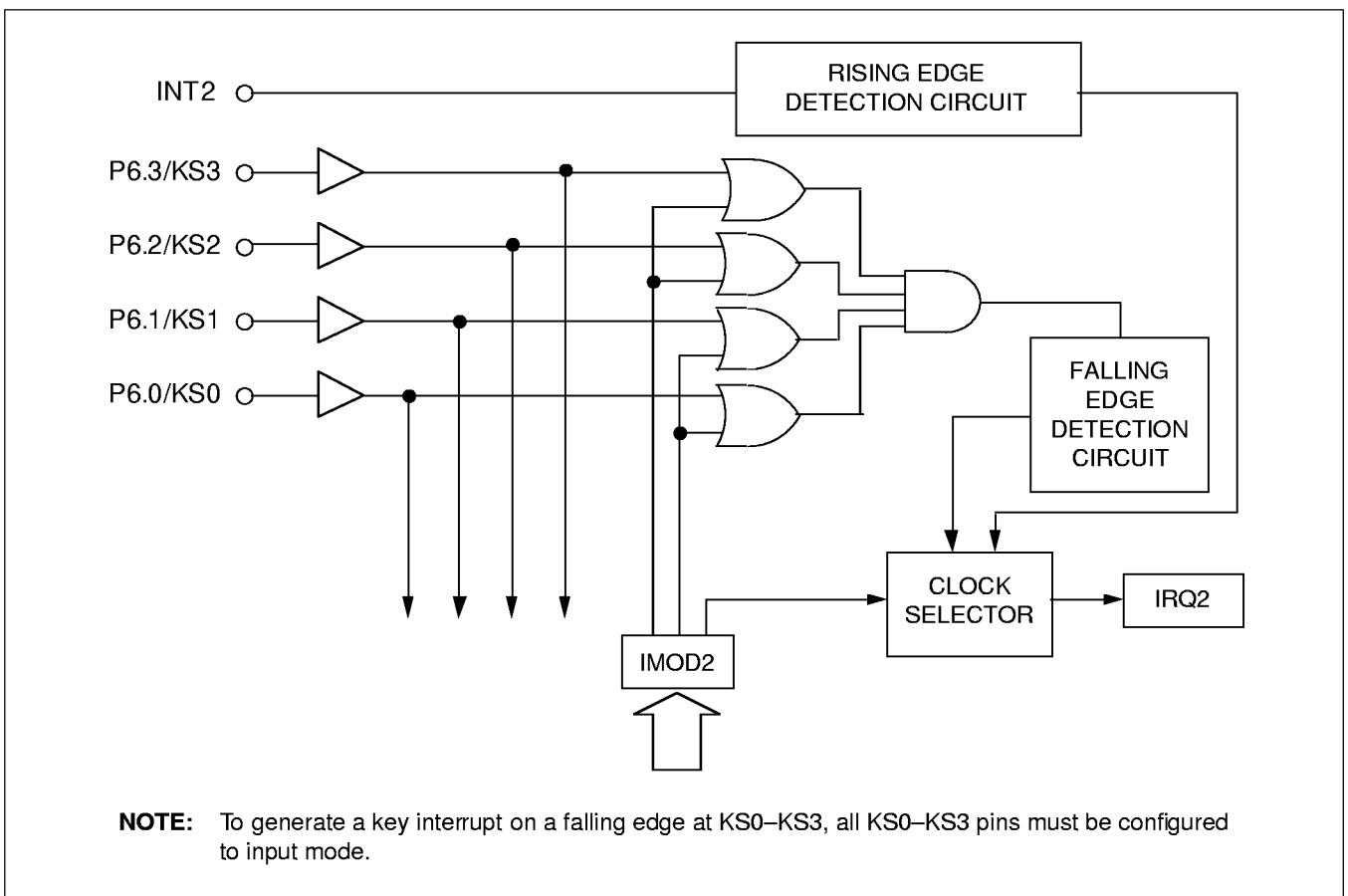


Figure 20. Curcuit Diagram for INT2 and KS0-KS3

### OSCILLATOR CIRCUITS

The KS57C3016 microcontroller has two oscillator circuits: a main system clock circuit, and a subsystem clock circuit. The power control register, PCON, is used to select normal CPU operating mode or one of two power-down modes. The main system clock frequency can be divided by 4, 8, or 64 by manipulating PCON register.

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, since they operate at very slow speeds and with very low power consumption (as low as 122  $\mu$ s at 32.768 kHz).

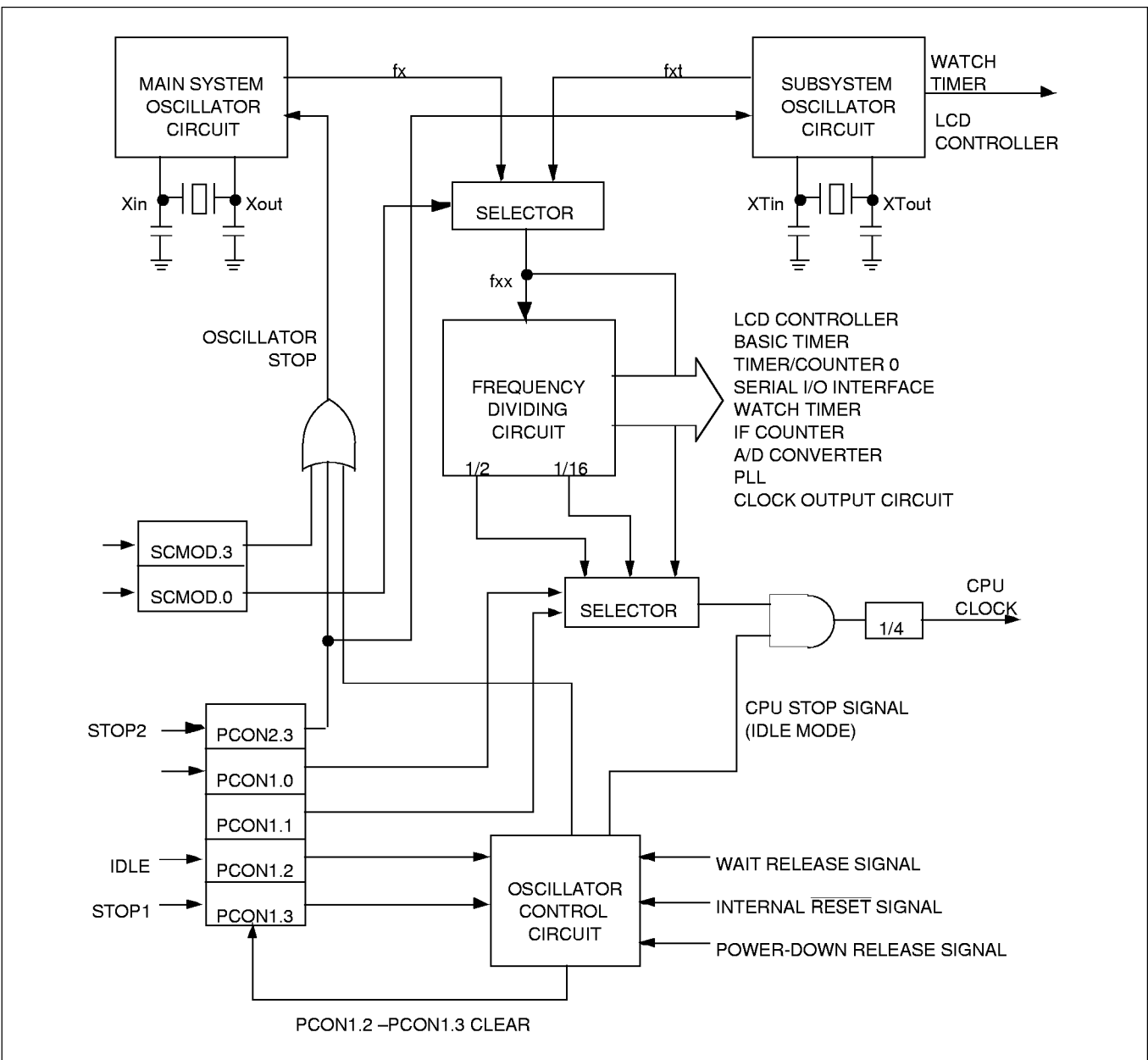


Figure 21. Clock Circuit Diagram

MAIN SYSTEM OSCILLATOR CIRCUITS

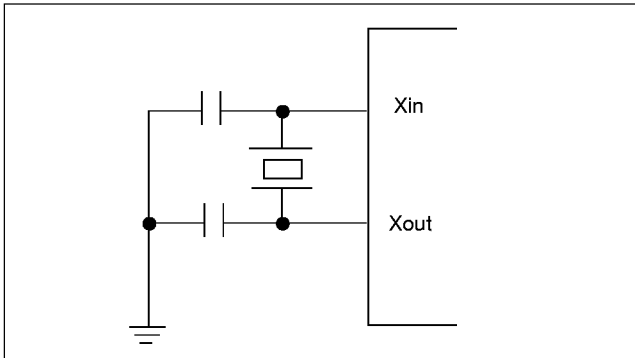


Figure 22. Crystal/Ceramic Oscillator (fx)

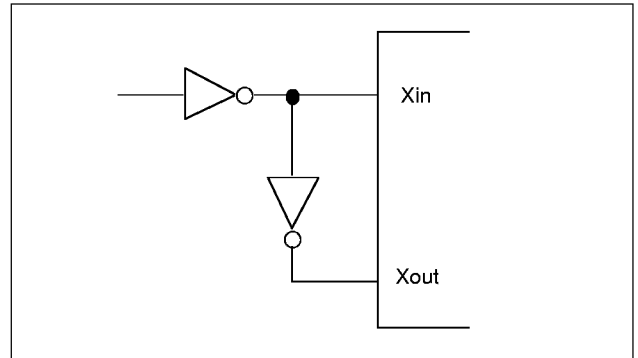


Figure 23. External Oscillator (fx)

SUBSYSTEM OSCILLATOR CIRCUITS

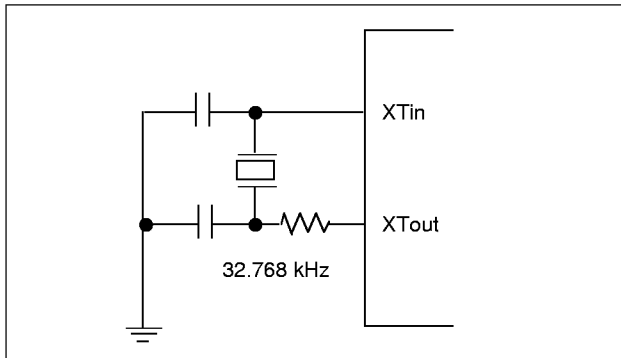


Figure 24. Crystal/Ceramic Oscillator (fxt)

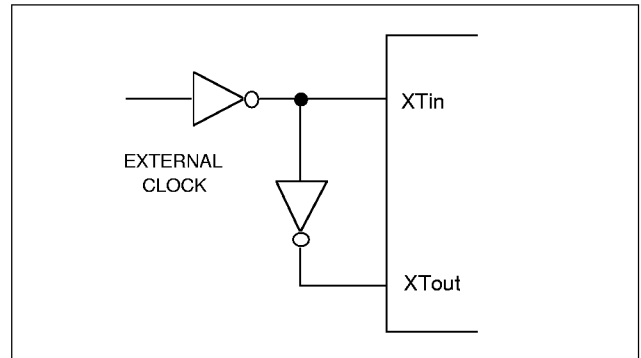


Figure 25. External Oscillator (fxt)

**POWER CONTROL REGISTER (PCON) and SUBSYSTEM CLOCK STOP CONTROL REGISTER (SUBSTP)**

The power control register, PCON, and the subsystem clock 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.

|               |      |          |        |        |        |
|---------------|------|----------|--------|--------|--------|
| <b>PCON</b>   | FB3H | PCON.3   | PCON.2 | PCON.1 | PCON.0 |
| <b>SUBSTP</b> | F8FH | SUBSTP.3 | 0      | 0      | 0      |

PCON.3 and PCON.2 can also be addressed by the STOP and IDLE instructions, respectively, to engage the Idle and Stop 1 power-down modes. Idle and Stop1 modes can be initiated by these instructions regardless of the current value of the enable memory bank flag (EMB). Stop2 mode is initiated by setting the SUBSTP.3 bit in the SUBSTP register.

**Table 11. 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 clocks 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                          | fx/4             |

** PROGRAMMING TIP — Setting the CPU Clock**

To set the CPU clock to 0.889  $\mu$ s 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.

**Table 12. Instruction Cycle Times for CPU Clock Rates**

| Selected CPU Clock | Resulting Frequency | Oscillation Source | Cycle Time (μs) |
|--------------------|---------------------|--------------------|-----------------|
| fx/64              | 70.312 kHz          | fx = 4.5 MHz       | 14.2            |
| fx/8               | 562.5 kHz           |                    | 1.77            |
| fx/4               | 1.125 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 used to select the CPU clock and to control main system clock oscillation. Only its least significant and most significant bits can be manipulated by 1-bit write instructions. Bits 2 and 1 are always logic zero.

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 main system clock oscillation. This can only be done by a STOP instruction.

**Table 13. 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, PCON, and the system clock mode register, SCMOD, determine whether a main system or a subsystem clock is selected as the CPU clock. This makes it possible to switch dynamically between the main and subsystem clocks and to modify operating frequencies.

### 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  $f_x/64$ ) and you want to switch from the  $f_x$  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  $f_x$  to  $f_{xt}$  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, 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 have 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  $15.3 \mu\text{s}$  at 4.19MHz after the standard oscillation stabilization interval of 31.3 ms has elapsed. Table 14 details the number of machine cycles that must elapse before a CPU clock switch modification goes into effect.

**Table 14. Elapsed Machine Cycles During CPU Clock Switch**

| AFTER<br>BEFORE |            | 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 | N/A               |            | 1 MACHINE CYCLE   |            | 1 MACHINE CYCLE       |            | N/A             |
|                 | PCON.1 = 1 | 8 MACHINE CYCLES  |            | N/A               |            | 8 MACHINE CYCLES      |            | N/A             |
|                 | PCON.0 = 0 | 8 MACHINE CYCLES  |            | N/A               |            | 8 MACHINE CYCLES      |            | N/A             |
|                 | PCON.1 = 1 | 16 MACHINE CYCLES |            | 16 MACHINE CYCLES |            | N/A                   |            | $f_x / 4f_{xt}$ |
| SCMOD.0 = 1     |            | N/A               |            | N/A               |            | $f_x / 4f_{xt}$ (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. Since the Xin 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, it is necessary to disable any interrupts which may occur during the time intervals shown in Table 14.
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 circuit is used to output clock pulses to the CLO pin. The clock output mode register (CLMOD) is used to enable or disable clock output to the CLO pin and to select the CPU clock source and frequency.

To output a frequency, set the clock output pin CLO/P2.0 to output mode and clear the pin's latch. Bit 2 in the CLMOD register must be "0".

**Table 15. 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       | fx/8                                 | 562.5 kHz, 4.096 kHz                      |
| 1                  | 0       | fx/16                                | 281.25 kHz, 2.048 kHz                     |
| 1                  | 1       | fx/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.

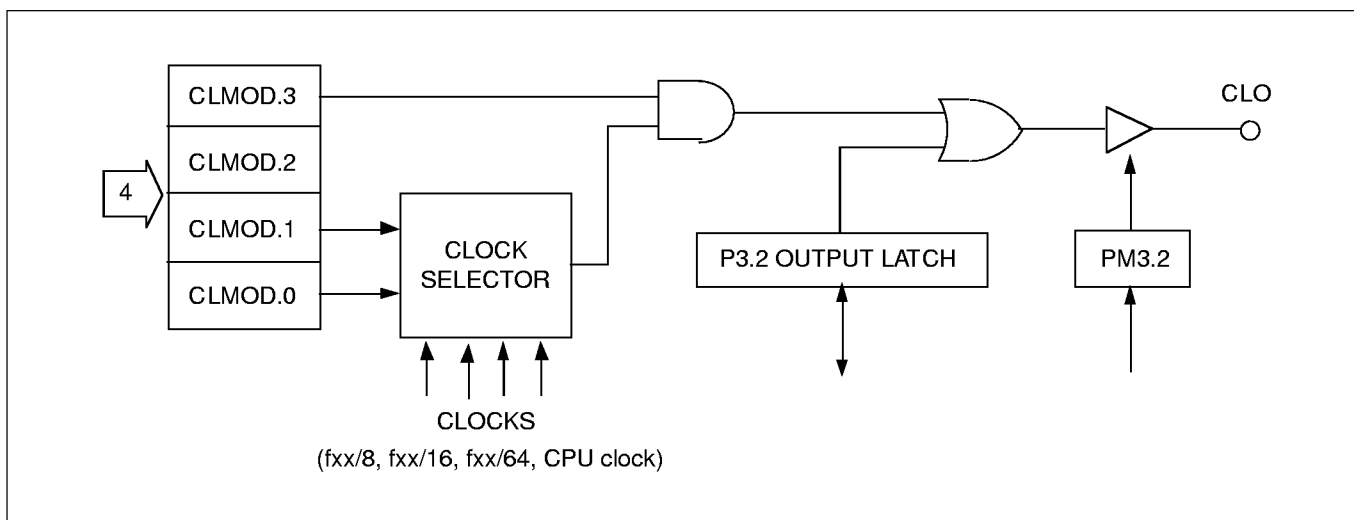


Figure 26. CLO Output Pin Circuit Diagram

### 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      ; P3.2 ← Output mode
BITR    P3.2        ; Clear P3.2 output latch
LD      A,#9H
LD      CLMOD,A

```

## POWER-DOWN

The KS57C3016 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. 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 peripheral

hardware components are powered-down. Stop2 mode is entered by setting the SUBSTP.3 bit. In stop2 mode, both main and subsystem clocks are stopped.

Only the PLL block is disabled in CE Low mode while other peripherals operate normally. The effect of power-down modes on specific peripheral hardware components is detailed in Table 16.

Idle or Stop1 modes are ended either by  $\overline{\text{RESET}}$  or by an interrupt that is enabled by the corresponding interrupt enable flag, IEx. (An exceptions to this rule is INT0.) Stop 2 mode can be ended by a RESET only.

**Table 16. 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 TCL0 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  |
| PWM                 | PWM output is disabled  | PWM output is disabled                                      | PWM output operates   | PWM output 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 instructions — it can only be changed by the external CE pin state.

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

The following sample code shows real-time clock and interrupt processing for key inputs to reduce 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 to low level
            LD      A,#3H
            LD      IMOD2,A    ; Select K0–K7 enable
            SMB     0
            BTR     IRQW
            BTR     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
CIDLE       RET
            IDLE
            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 17.

**Table 17. Unused Pin Connections for Reduced Power Consumption**

| Pin/Share Pin Names   | Recommended Connection  |
|---|---|
| P0.0 / $\overline{SCK}$<br>P0.1 / SO<br>P0.2 / SI<br>P0.3 / BICO  | Input mode: Connect to $V_{DD}$<br>Output mode: Do not connect                |
| P1.0 / INT0 – P1.2 / INT2   | Connect to $V_{DD}$   |
| P1.3 / INT4   | Connect to $V_{SS}$   |
| P2.0 / TCL0<br>P2.1 / TCL1<br>P2.2 / CAP<br>P2.3<br>P3.0 / TCLO0<br>P3.1 / TCLO1<br>P3.2 / CLO<br>P3.3 / BUZ<br>P4.0–P4.3<br>P5.0–P5.3<br>P6.0 / KS0 – P6.3 / KS3<br>P7.0 / KS4 – P7.3 / KS7<br>P8.0–P8.3<br>P9.0–P9.3<br>P10.0–P10.1 | Input mode: Connect to $V_{DD}$<br>Output mode: Do not connect                |
| SEG0–SEG27<br>SEG28 / P11.0 –<br>SEG39 / P13.3<br>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 $V_{DD}$                                     |
| $XT_{out}$  | Do not connect  |
| AMIF, FMIF  | Connect to $V_{SS}$   |
| TEST  | Connect to $V_{SS}$   |

**RESET**

A system reset operation can be initiated by  $\overline{\text{RESET}}$  or the CE pin. When a reset operation occurs, the system is initialized and program is executed from reset vector address. The CE reset occurs when the CE pin rises from low to high. CE resets can also be used for system initialization. However,  $\overline{\text{RESET}}$  signal is not generated automatically.

**Table 18. Hardware Register Values After  $\overline{\text{RESET}}$** 

| Hardware Component or Subcomponent             | If $\overline{\text{RESET}}$ Occurs During Operating Mode  | If $\overline{\text{RESET}}$ Occurs After Power-On   |
|--|--|--|
|  | RESET Pin  | RESET Pin  |
| Program counter (PC)                           | Lower six bits of address 0000H are transferred to PC13–8, and the contents of 0001H to PC7–0.                     | Lower six bits of address 0000H are transferred to PC13–8, and the contents of 0001H to PC7–0.                     |
| <b>Program Status Word (PSW):</b>              |  |  |
| 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  |
| <b>Data Memory (RAM):</b>                      |  |  |
| General registers E, A, L, H, X, W, Z, Y       | Values retained  | Undefined  |
| General-purpose registers                      | Values retained (note1)  | Undefined  |
| Bank selection registers (SMB, SRB)            | 0, 0   | 0, 0   |
| BSC register (BSC0–BSC3)                       | 0  | 0  |
| <b>Clocks:</b>                                 |  |  |
| Power control register (PCON)                  | 0  | 0  |
| Subsystem clock stop control register (SUBSTP) | 0  | 0  |
| Clock output mode register (CLMOD)             | 0  | 0  |
| System clock mode register (SCMOD)             | 0  | 0  |

**Note1:** The values of the 0F8H-0FDH are not retained when a  $\overline{\text{RESET}}$  signal is input.

**Table 18. Hardware Register Values After  $\overline{\text{RESET}}$  (Continued)**

| Hardware Component<br>or Subcomponent | If $\overline{\text{RESET}}$ Occurs During<br>Operating Mode | If $\overline{\text{RESET}}$ Occurs After<br>Power-On |
|---------------------------------------|--|---|
|                                       | $\overline{\text{RESET}}$ Pin                                | $\overline{\text{RESET}}$ Pin                         |
| <b>Interrupts:</b>                    |  |   |
| 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   |
| <b>I/O Ports:</b>                     |  |   |
| Output buffers                        | Off  | Off   |
| Output latches                        | 0  | 0   |
| Port mode flags (PM)                  | 0  | 0   |
| Pull-up resistor mode reg (PUMOD1/2)  | 0  | 0   |
| <b>Basic Timer:</b>                   |  |   |
| Count register (BCNT)                 | Undefined  | Undefined   |
| Mode register (BMOD)                  | 0  | 0   |
| Output enable flag (BOE)              | 0  | 0   |
| <b>Timer/Counter 0:</b>               |  |   |
| Count registers (TCNT0)               | 0  | 0   |
| Reference registers (TREF0)           | FFH  | FFH   |
| Mode registers (TMOD0)                | 0  | 0   |
| Output enable flags (TOE0)            | 0  | 0   |
| <b>Watch Timer:</b>                   |  |   |
| Watch timer mode register (WMOD)      | 0  | 0   |
| <b>LCD Driver/Controller:</b>         |  |   |
| LCD mode register (LMOD)              | 0  | 0   |
| LCD control register (LCON)           | 0  | 0   |
| Display data memory                   | Values retained  | Undefined   |
| Output buffers                        | Off  | Off   |

Table 18. Hardware Register Values After  $\overline{\text{RESET}}$  (Concluded)

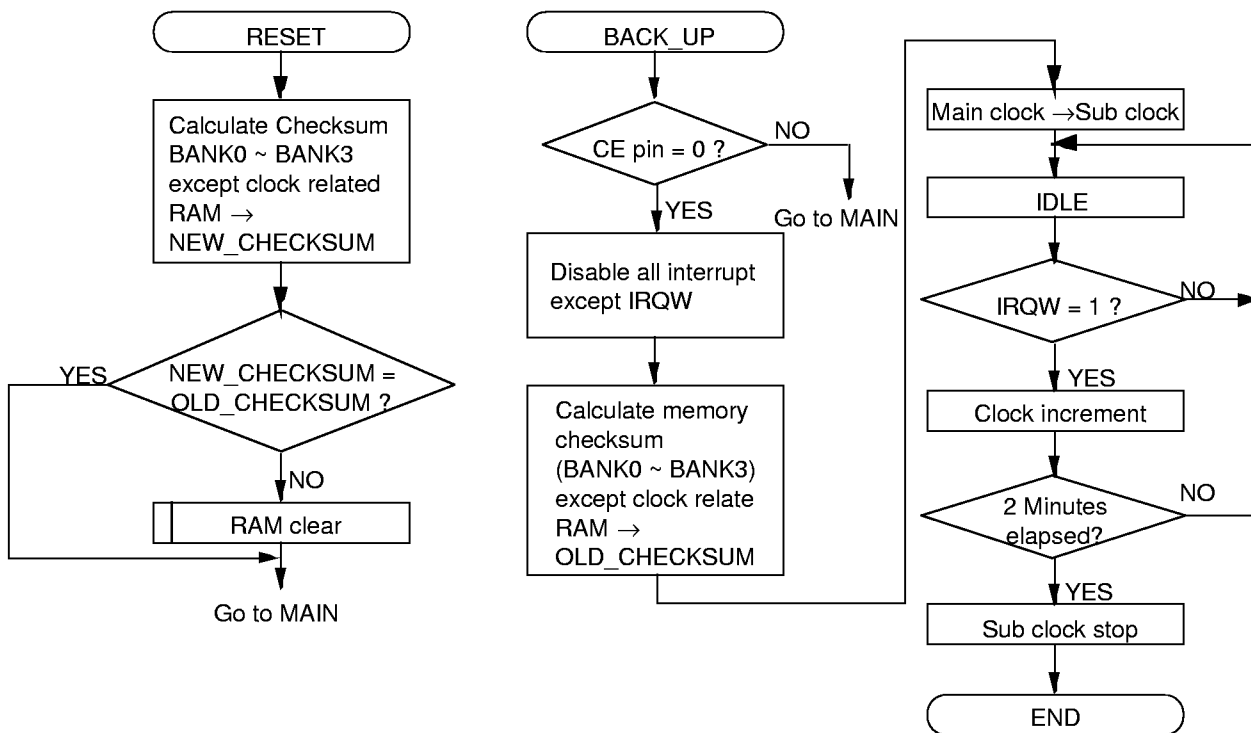
| Hardware Component<br>or Subcomponent       | If $\overline{\text{RESET}}$ Occurs During<br>Operating Mode | If $\overline{\text{RESET}}$ Occurs After<br>Power-On |
|---|--|---|
|   | $\overline{\text{RESET}}$ Pin                                | $\overline{\text{RESET}}$ Pin                         |
| <b>Serial I/O Interface:</b>                |  |   |
| SIO mode register (SMOD)                    | 0  | 0   |
| SIO interface buffer (SBUF)                 | Values retained  | Undefined   |
| <b>PLL</b>                                  |  |   |
| PLL mode register (PLMOD)                   | Values retained  | Undefined   |
| PLL Data register (PLLD0–PLLD3)             | Values retained  | Undefined   |
| PLL flag register (PLLREG)                  | (Note 1)   | (Note 2)  |
| <b>PWM</b>                                  |  |   |
| PWM mode register (PWMOD)                   | 0  | 0   |
| PWM counter                                 | 0  | 0   |
| PWM reference register<br>(PWMREF0–PWMREF5) | 0  | 0   |
| <b>IF Counter</b>                           |  |   |
| IF counter mode register (IFMOD)            | 0  | 0   |
| IF counter (IFCNT0, IFCNT1)                 | 0  | 0   |
| <b>A/D Converter</b>                        |  |   |
| A/D converter mode register (ADM0D)         | 0  | 0   |
| A/D converter data register (ADATA)         | 0  | 0   |

**NOTES:**

- |                    |                               |                  |
|--------------------|-------------------------------|------------------|
| 1. ULFG: undefined | CEFG: current state of CE pin | IFCFG: 0         |
| 2. ULFG: undefined | CEFG: current state of CE pin | IFCFG: undefined |



**PROGRAMMING TIP — Back-up routine and Reset routine**



**DATA BACK-UP ROUTINE**

**RESET ROUTINE FOR DATA BACK-UP**

**I/O PORTS**

The KS57C3016 has 14 ports. There are 4 input pins, 12 output pins, 23 configurable I/O pins, and 16 n-channel open-drain I/O pins, for a total of 55 I/O pins. Pin addresses for all ports except ports 11–13 are mapped in bank 15 of the RAM. Ports 11–13 pin addresses are in bank 1 of the RAM.

**PORT MODE FLAGS (PM FLAGS)**

Port mode flags (PM) are used to configure I/O ports to input or output mode by setting or clearing the corresponding I/O buffer. If a PM bit is "0", the corresponding I/O pin is set to input mode. If the PM bit is "1", the pin is set to output mode: PM0.0 for P0.0, PM0.1 for P0.1, and so on.


**PULL-UP RESISTOR MODE REGISTER (PUMOD)**

The 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 PUMOD bit is "1", a pull-up resistor is assigned to the corresponding I/O port: PUR3 for port 3, PUR2 for port 2, and so on. I/O ports 4, 5, 7 and 8 are an exception, however, as these port pins may only be assigned internal pull-up resistors via mask option.

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.

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

| PUMOD ID | Address | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------|---------|-------|-------|-------|-------|
| PUMOD1   | FDCH    | PUR3  | PUR2  | PUR1  | PUR0  |
|          | FDDH    | "0"   | PUR6  | "0"   | "0"   |
| PUMOD2   | FDEH    | "0"   | PUR10 | PUR9  | "0"   |

 **PROGRAMMING TIP — Enabling and Disabling I/O Port Pull-Up Resistors  
Configuring I/O Ports as Input or Output**

P6 enable pull-up resistors.  
Configure P7 as an output port:

```

BITS      EMB
SMB       15
LD        EA,#40H
LD        PUMOD1,EA      ; P6 enable, P0, 1, 2, 3 disable
LD        EA,#0F0H
LD        PMG0, EA       ; P7 ← Output

```

**Table 20. Port Mode Group Flags (8-Bit W)**

| PM Group ID | Address | Bit 3  | Bit 2  | Bit 1  | Bit 0  |
|-------------|---------|--------|--------|--------|--------|
| PMG0        | FE6H    | "0"    | "0"    | "0"    | "0"    |
|             | FE7H    | PM7.3  | PM7.2  | PM7.1  | PM7.0  |
| PMG1        | FE8H    | PM0.3  | PM0.2  | PM0.1  | PM0.0  |
|             | FE9H    | "0"    | "0"    | PM5    | PM4    |
| PMG2        | FEAH    | PM2.3  | PM2.2  | PM2.1  | PM2.0  |
|             | FEBH    | PM3.3  | PM3.2  | PM3.1  | PM3.0  |
| PMG3        | FECH    | PM6.3  | PM6.2  | PM6.1  | PM6.0  |
|             | FEDH    | PM8.3  | PM8.2  | PM8.1  | PM8.0  |
| PMG4        | FEEH    | PM9.3  | PM9.2  | PM9.1  | PM9.0  |
|             | FEFH    | PM10.3 | PM10.2 | PM10.1 | PM10.0 |

**PIN ADDRESSING FOR OUTPUT PORTS 11–13**

The addresses for the ports 11–13, 1-bit output pin latches are located in bank 1 of data memory instead of bank 15.

To address ports 11–13 output pins, use the settings EMB = 1 and SMB = 1. The LCD mode register, LMOD is used to control whether the pin address is used for LCD data output or for normal data output.

PORT 0 CIRCUIT DIAGRAM

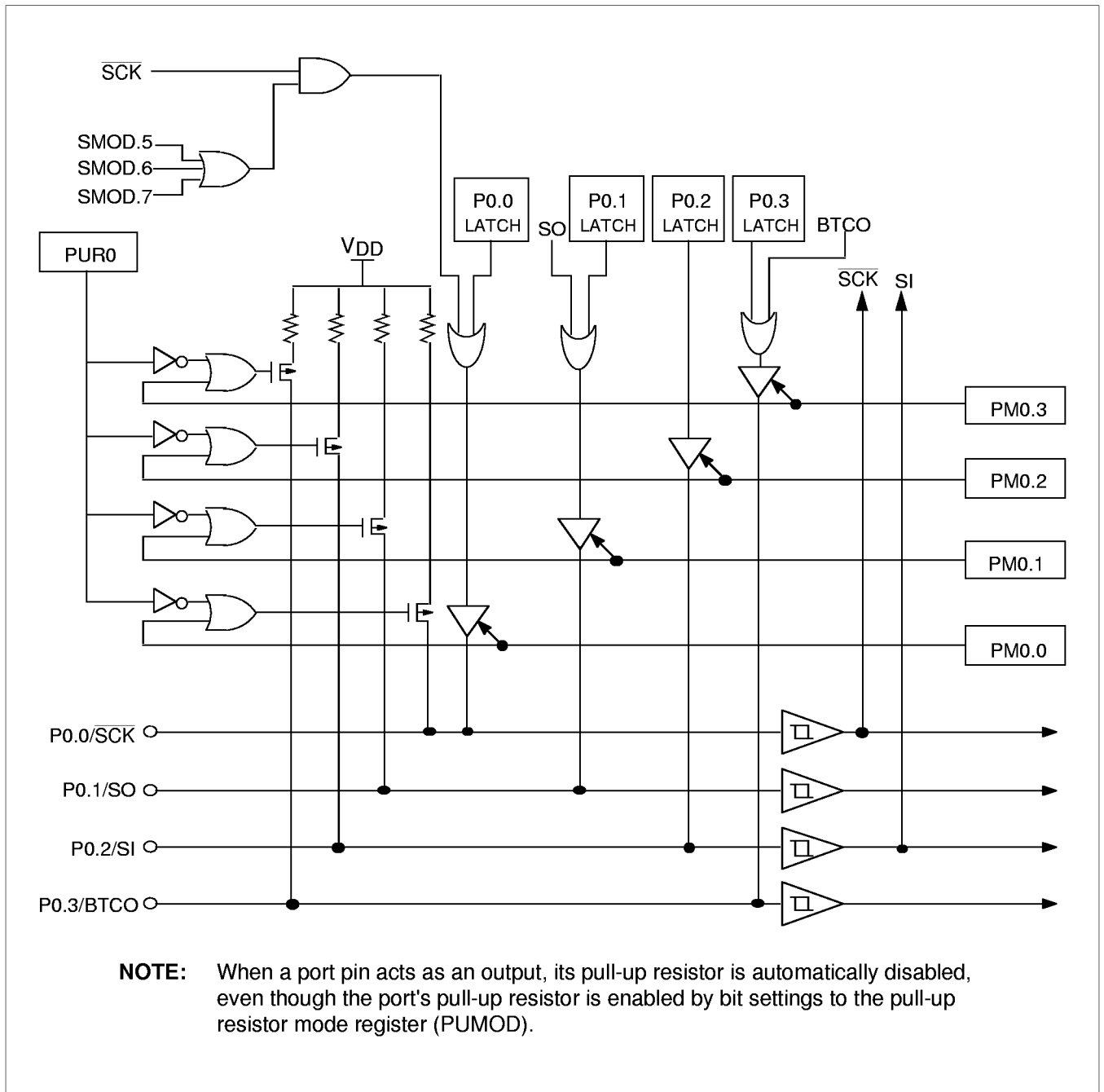


Figure 27 Port 0 Circuit Diagram

PORT 1 CIRCUIT DIAGRAM

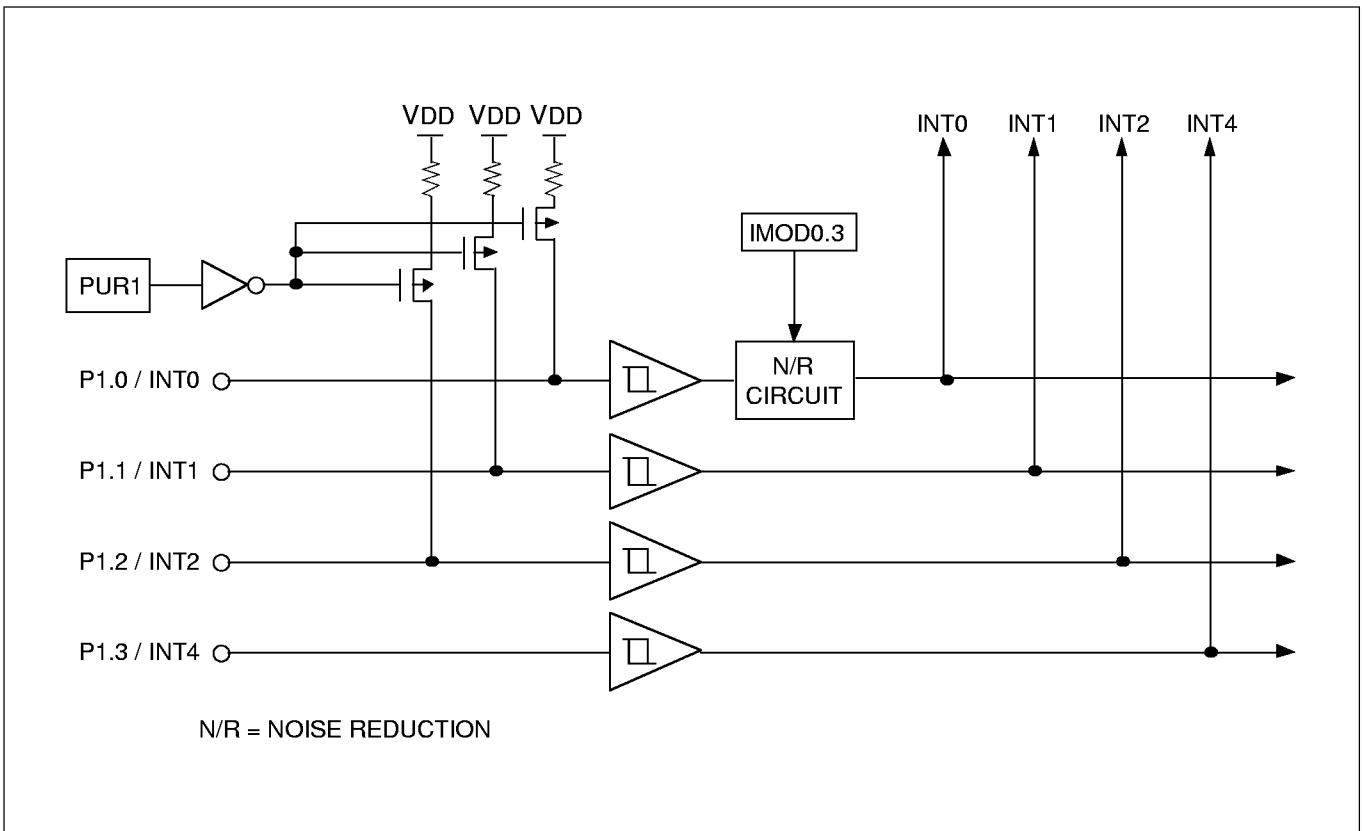


Figure 28. Port 1 Circuit Diagram

PORT 2, 3, 6, 9, 10 CIRCUIT DIAGRAM

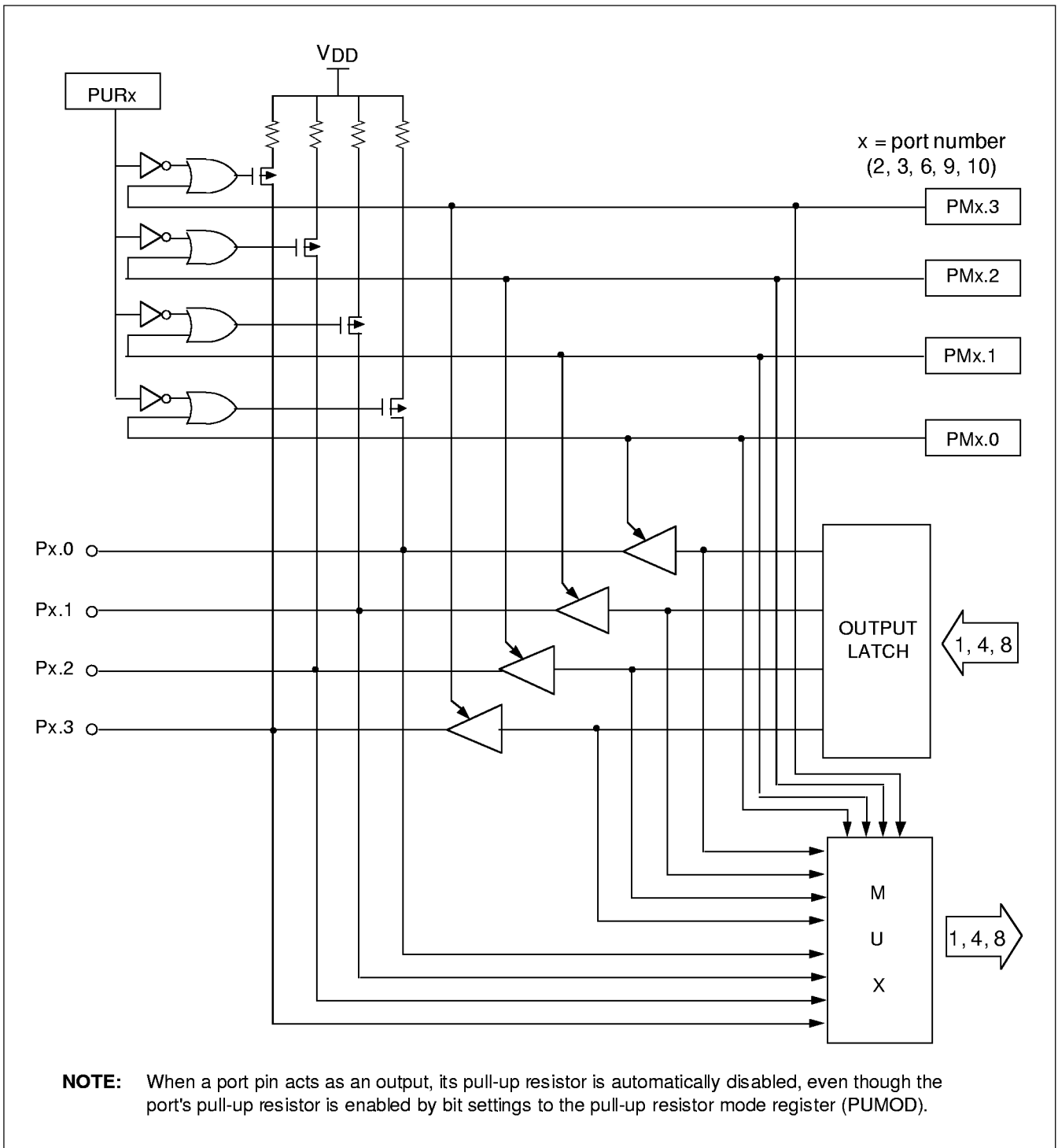


Figure 29. Port 2, 3, 6, 9 and 10 Circuit Diagram

PORT 4, 5 CIRCUIT DIAGRAM

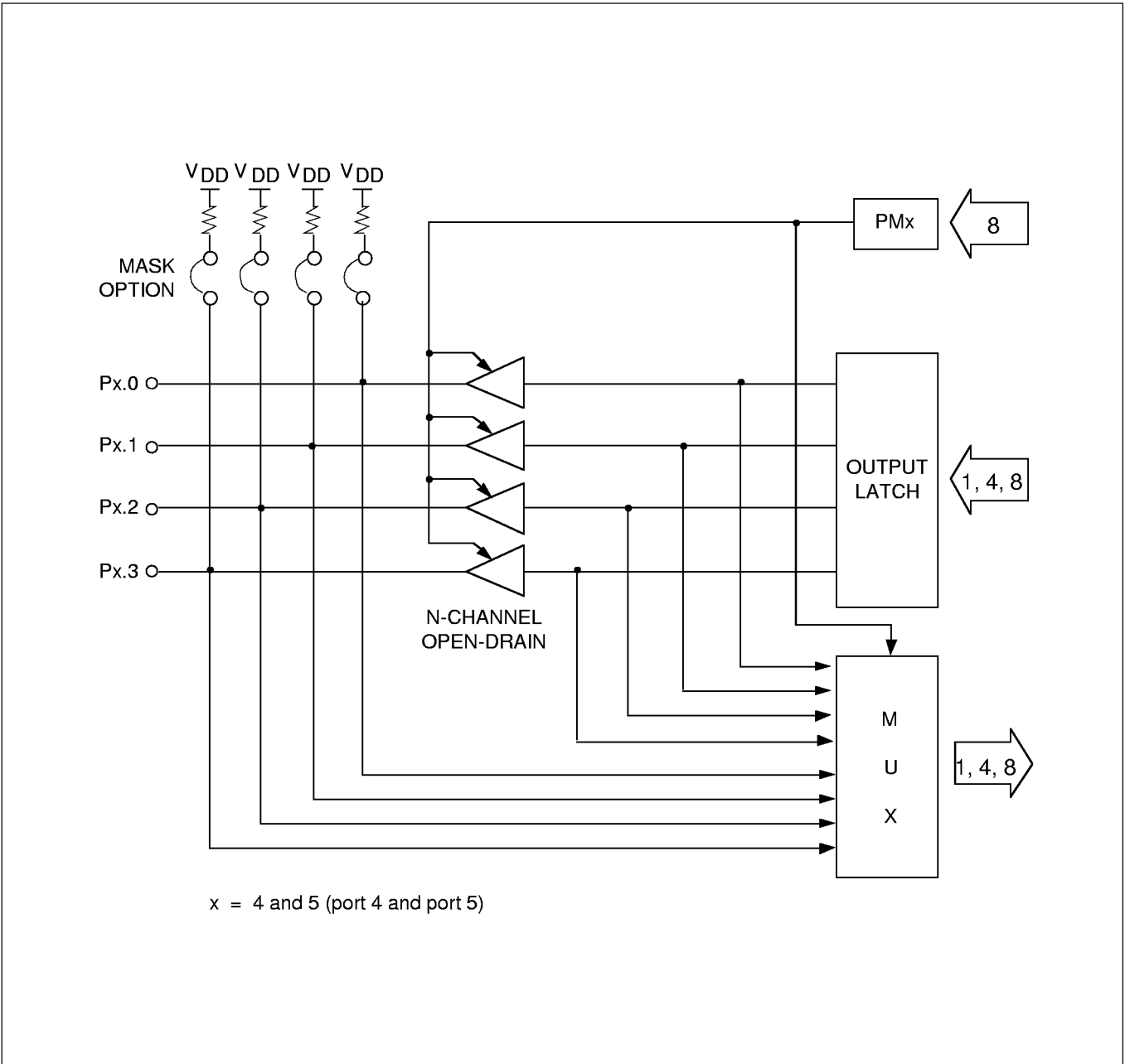


Figure 30. Port 4 and 5 Circuit Diagram

PORT 7, 8 CIRCUIT DIAGRAM

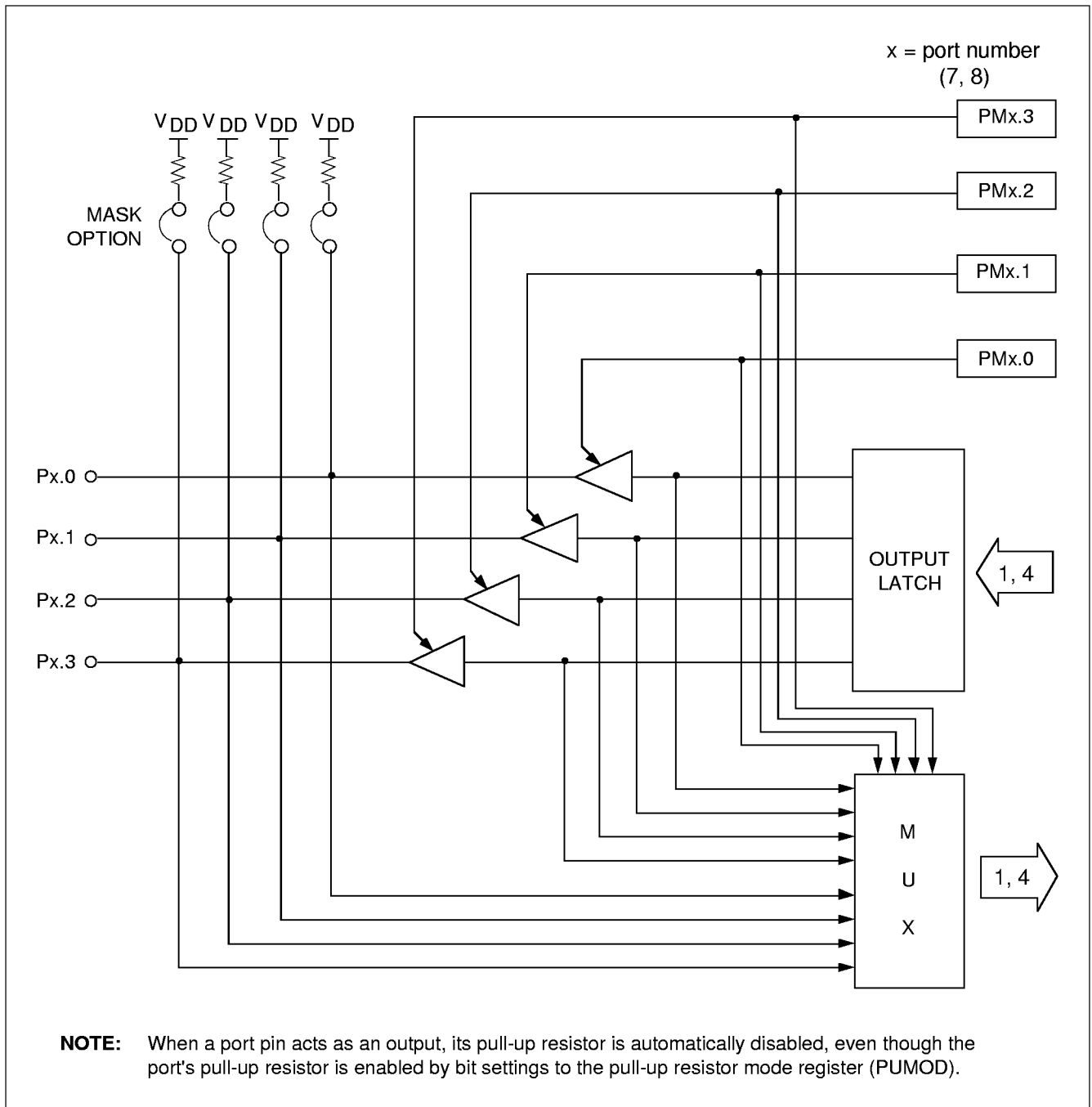


Figure 31. Ports 7 and 8 Circuit Diagram

### BASIC TIMER (BT)

The basic timer generates interrupt requests at precise intervals. You can use the basic timer as a "watchdog" timer for monitoring system events or use BT output to stabilize clock oscillation when Stop mode is released by an interrupt and following RESET.

#### Interval Timer Function

The measurement of elapsed time intervals is the basic timer's primary function. The standard interval is 256 BT clock pulses. To restart the basic timer, set bit 3 of the mode register BMOD to "1". 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. An overflow causes the BT interrupt request flag (IRQB) to be set to "1" to signal that the designated time interval has elapsed. An interrupt

request is then generated, BCNT is cleared to "0", and counting continues from 00H.

#### Watchdog Timer Function

The basic timer can also be used 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. To enable BTCO output operation, set the output latch for pin P0.3 to "0" and the port mode flag for P0.3 (PM0.3) to "1".

#### Oscillation Stabilization Interval Control

Setting bits 2–0 of the BMOD register determines the time interval (also referred to as 'wait time') required to stabilize clock signal oscillation when power-down mode is released by an interrupt. When a RESET signal is generated, the standard stabilization interval for system clock oscillation following a RESET is 29.1 ms at 4.5 MHz.

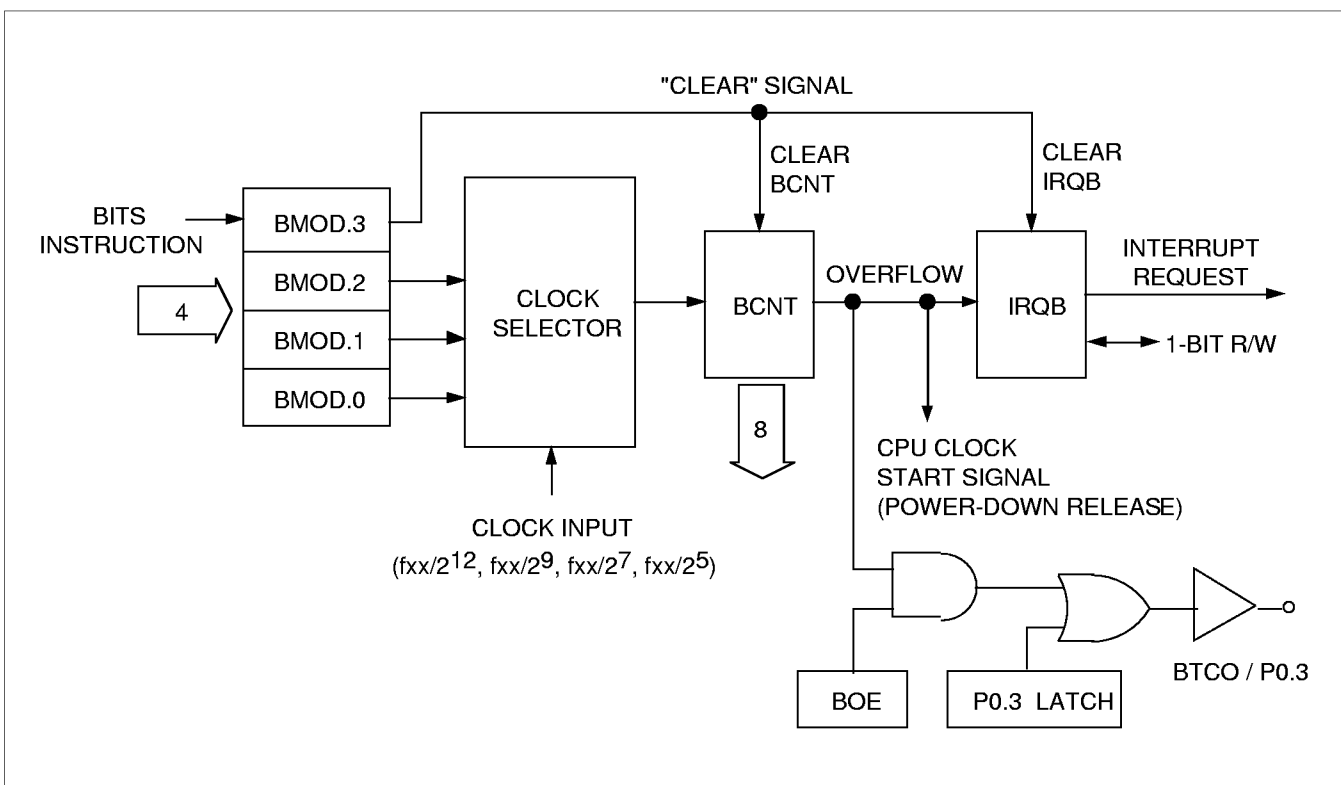


Figure 32. Basic Timer Circuit Diagram



**BASIC TIMER MODE REGISTER (BMOD)**

The basic timer mode register, BMOD, is used to select input frequency and oscillation stabilization time. The most significant bit of the BMOD register, BMOD.3, is used to start the basic timer again. When BMOD.3 is set to "1", the contents of the BT counter register (BCNT) and the BT interrupt request flag (IRQB) are both cleared to "0", and timer operation is restarted.

**Table 21. Basic Timer Mode Register (BMOD) Organization**

|               |  |
|---------------|--|
| <b>BMOD.3</b> | <b>Basic Timer Enable/Disable Control Bit</b>            |
| 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      | $f_{xx}/2^{12}$ (1.098 kHz) | $2^{20}/f_{xx}$ (233 ms)  |
| 0      | 1      | 1      | $f_{xx}/2^9$ (8.789 kHz)    | $2^{17}/f_{xx}$ (29.1 ms) |
| 1      | 0      | 1      | $f_{xx}/2^7$ (35.15 kHz)    | $2^{15}/f_{xx}$ (7.28 ms) |
| 1      | 1      | 1      | $f_{xx}/2^5$ (140.6 kHz)    | $2^{13}/f_{xx}$ (1.82 ms) |

**NOTES:**

1. Clock frequencies and stabilization intervals assume a system oscillator clock frequency ( $f_{xx}$ ) of 4.5 MHz.
2.  $f_{xx}$  = 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 table column 'Oscillation Stabilization' can also be interpreted as "Interrupt Interval Time."
4. The standard stabilization time for system clock oscillation following a RESET is 29.1 ms at 4.5 MHz. m

**BASIC TIMER COUNTER (BCNT)**

BCNT is an 8-bit counter register for the basic timer. When BCNT has incremented to hexadecimal 'FFH', 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 reads, 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/P0.3. 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.

|      |           |            |   |
|------|-----------|------------|---|
| F92H | 1-Bit R/W |            |   |
| TOE1 | TOE0      | <b>BOE</b> | 0 |

### PROGRAMMING TIP — Using the Basic Timer

- To read the basic timer count register (BCNT):

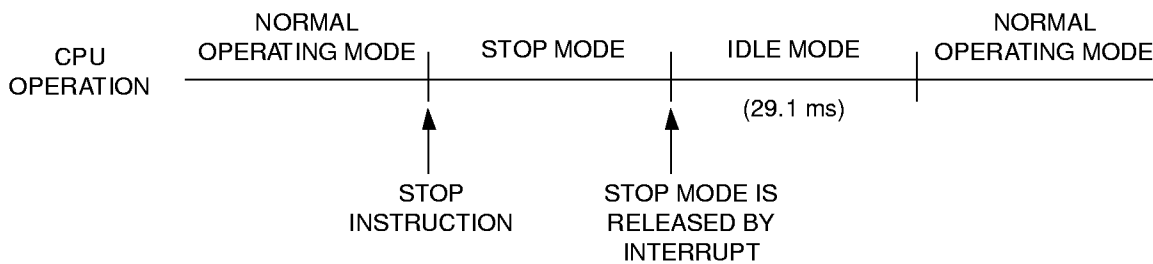
```

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

- When 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                               ; Set stop power-down mode
          NOP
          NOP
  
```



- 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"
  
```

- 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, TC generates an interrupt request. By counting signal transitions and comparing the current counter value with the reference register value, TC can be used to measure specific time intervals.

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 lets you adjust data transmission rates across the serial interface.

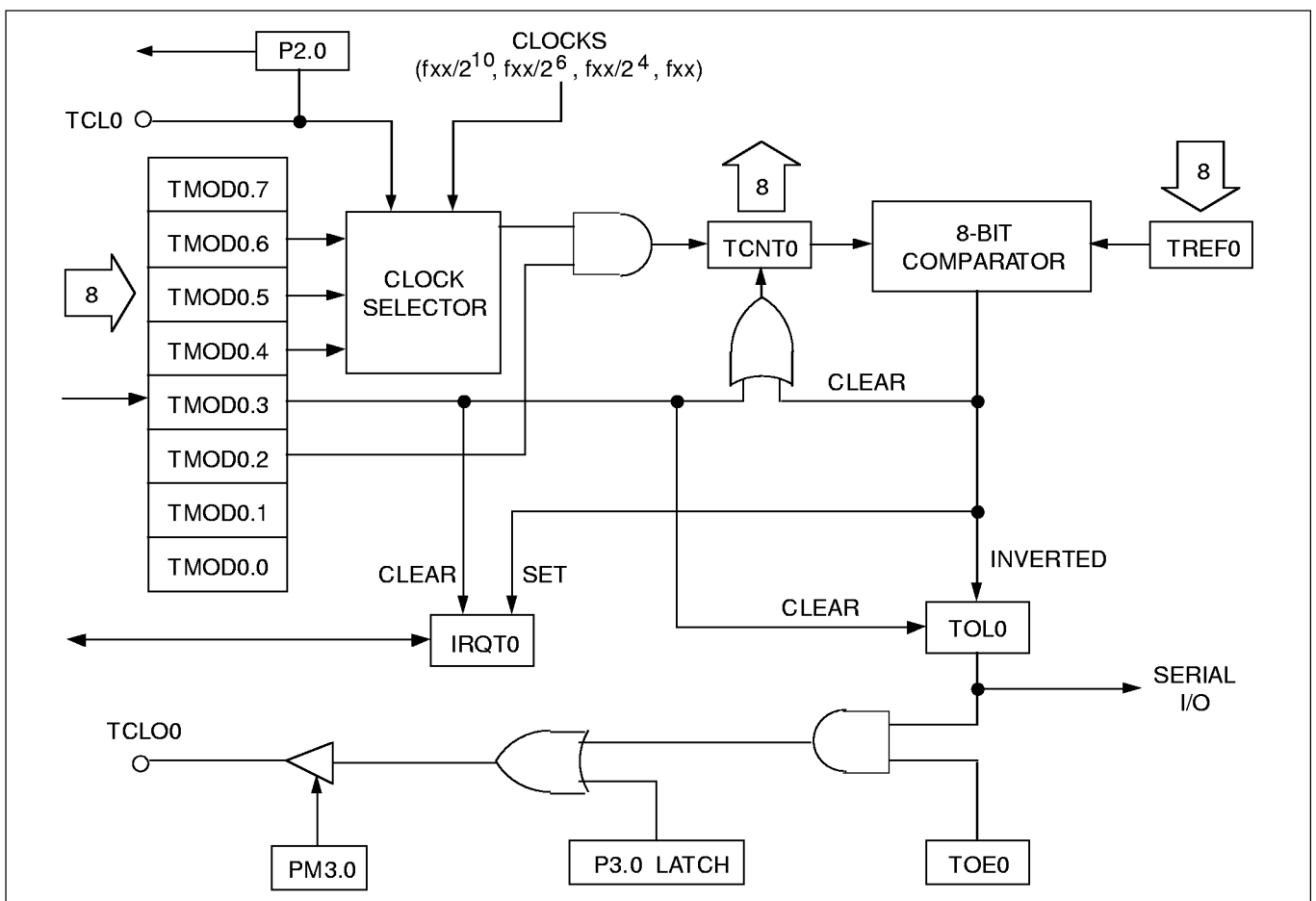


Figure 33. TC0 Circuit Diagram

**PROGRAMMABLE TIMER/COUNTER FUNCTION**

Timer/counter 0 can be programmed 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), the TC0 interrupt request flag (IRQT0) is set to logic one, the status of TOL0 is inverted, and the interrupt is generated. The content of TCNT0 is then cleared to 00H and TC0 continues counting.

** PROGRAMMING TIP — TC0 Signal Output to the TCLO0 Pin**

Output a 30 ms pulse width signal to the TCLO0 pin:

|      |          |                      |
|------|----------|----------------------|
| BITS | EMB      |                      |
| SMB  | 15       |                      |
| LD   | EA,#83H  |                      |
| LD   | TREF0,EA |                      |
| LD   | EA,#4CH  |                      |
| LD   | TMOD0,EA |                      |
| LD   | EA,#10H  |                      |
| LD   | PMG3,EA  | ; P3.0 ← output mode |
| BITR | P3.0     | ; P3.0 clear         |
| BITS | TOE0     |                      |

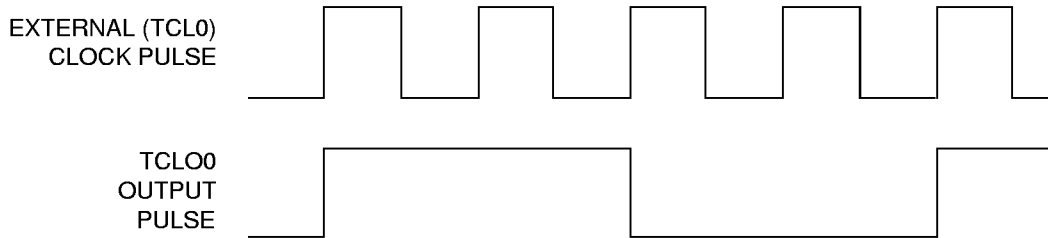
**TC0 EVENT COUNTER FUNCTION**

Timer/counter 0 can monitor or detect system 'events' by using the external clock input at the TCL0 pin as the counter source. 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, P2.0/TCL0 must be set to input mode.

Using timer/counter 0, a modifiable clock frequency can be output to the TC0 clock output pin, TCLO0. To enable the output to the TCLO0/P3.0, the I/O mode flag for P3.0 (PM3.0) must be set to output mode and output latch value for P3.0 must be set to "0" when timer output enable flag (TOE0) has been set to "1".

**PROGRAMMING TIP — External TCL0 Clock Output to the TCLO0 Pin**

Output external TCL0 clock pulse to the TCLO0 pin (divided by four):



```

BITS      EMB
SMB      15
LD        EA,#01H
LD        TREF0,EA
LD        EA,#0CH
LD        TMOD0,EA
LD        EA,#10H
LD        PMG3,EA      ; P3.0 ← output mode
BITR      P3.0         ; P3.0 clear
BITS      TOE0
    
```

**TC0 MODE REGISTER (TMOD0)**

TMOD0 is the 8-bit mode control register 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.

**Table 22. TC0 Mode Register (TMOD0) Organization**

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

Table 23. 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       | $f_{xx}/2^{10}$ (4.39 kHz)                   |
| 1       | 0       | 1       | $f_{xx}/2^6$ (70.3 kHz)                      |
| 1       | 1       | 0       | $f_{xx}/2^4$ (281 kHz)                       |
| 1       | 1       | 1       | $f_{xx} = 4.5$ 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; then restart TC0 counting operation:

```

BITS    EMB
SMB     15
BITS    TMOD0.3

```

#### TC0 REFERENCE REGISTER (TREF0)

TREF0 is used to store a reference value to be compared to the incrementing TCNT0 register in order to identify an elapsed time interval.

Use the following formula to calculate the correct value to load to the TREF0 reference register:

TC0 timer interval =

$$(\text{TREF0 value} + 1) \times \frac{1}{\text{TMOD0frequencysetting}}$$


Assuming TREF0 value  $\neq$  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.

| F92H |             | 1-Bit R/W |   |
|------|-------------|-----------|---|
| 0    | <b>TOE0</b> | BOE       | 0 |

When you set the TOE0 flag to "1", the contents of TOL0 can be output to the TCLO0 pin.

 **PROGRAMMING TIP — Setting a TC0 Timer Interval**

To set a 30 ms timer interval for TC0, given  $f_{xx} = 4.5 \text{ MHz}$ , 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 =  $f_{xx}/2^{10}$ , and TREF0 is set to FFH):
2. Calculate the TREF0 value:

$$30 \text{ ms} = \frac{\text{TREF0value}+1}{4.39\text{kHz}}$$

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

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

3. Load the value 83H to the TREF0 register:

```

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

### WATCH TIMER

Watch timer functions include real-time and watch-time measurement and interval timing for the system clock. To start watch timer operation, set bit 2 of the watch timer mode register, WMOD.2, to "1". The watch timer starts, the interrupt request flag IRQW is automatically set to "1", and interrupt requests commence in 0.5-second intervals. Because the watch timer functions as a quasi-interrupt instead of a vectored interrupt, the IRQW flag should be cleared to "0" by program software as soon as a requested interrupt service routine has been executed.

The watch timer can generate a steady 2 kHz, 4 kHz, 8 kHz, or 16 kHz signal to the BUZ pin. To generate a BUZ signal, the output latch for I/O port 3.3 is cleared to "0" and the port 3.3 output mode flag (PM3.3) set to 'output' mode. By setting WMOD.1 to "1", the watch timer will function in high-speed mode, generating an interrupt every 3.91 ms. High-speed mode is useful for timing events for program debugging sequences.

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

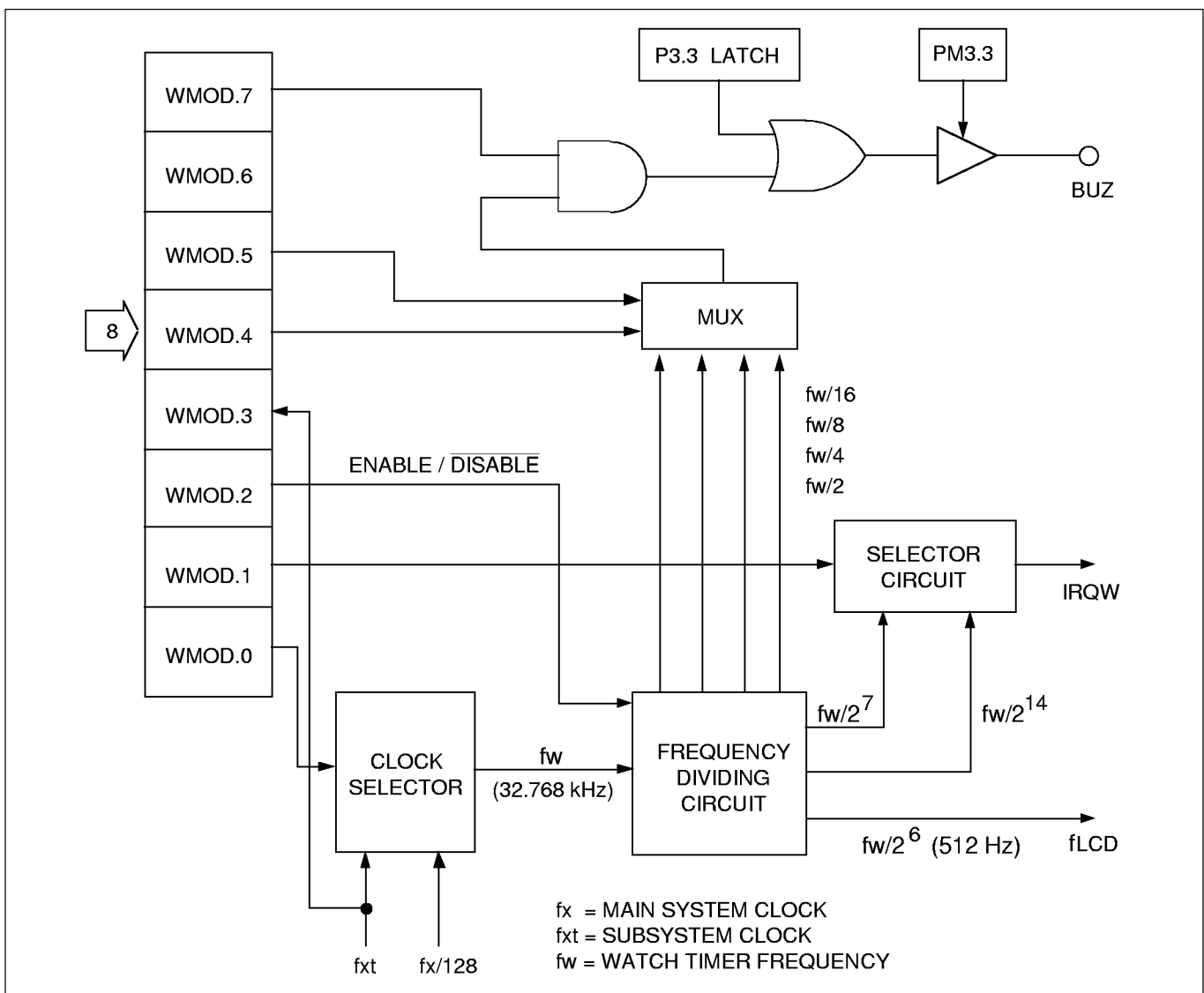


Figure 34. Watch Timer Circuit Diagram



**WATCH TIMER MODE REGISTER (WMOD)**

The watch timer mode register WMOD is used to select specific watch timer operations.

**Table 24. Watch Timer Mode Register (WMOD) Organization (8-Bit W)**

| 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 <sub>in</sub> pin is low (read-only bit)  |         |
|             | 1      |   | Input level to XT <sub>in</sub> pin is high (read-only bit) |         |
| WMOD.2      | 0      |   | Disable watch timer; clear frequency dividing circuits      |         |
|             | 1      |   | Enable watch timer  |         |
| WMOD.1      | 0      |   | Normal mode; sets IRQW to 0.5 s                             |         |
|             | 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 4.5 MHz.

**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,#80H
LD        PMG3,EA      ; P3.3 ← output mode
BITR     P3.3
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

To operate the A/D converter, one of the four analog input channels is selected by writing the appropriate value to the ADC mode register.

To start the converter, the ADSTR flag in the control register AFLAG must be set 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.

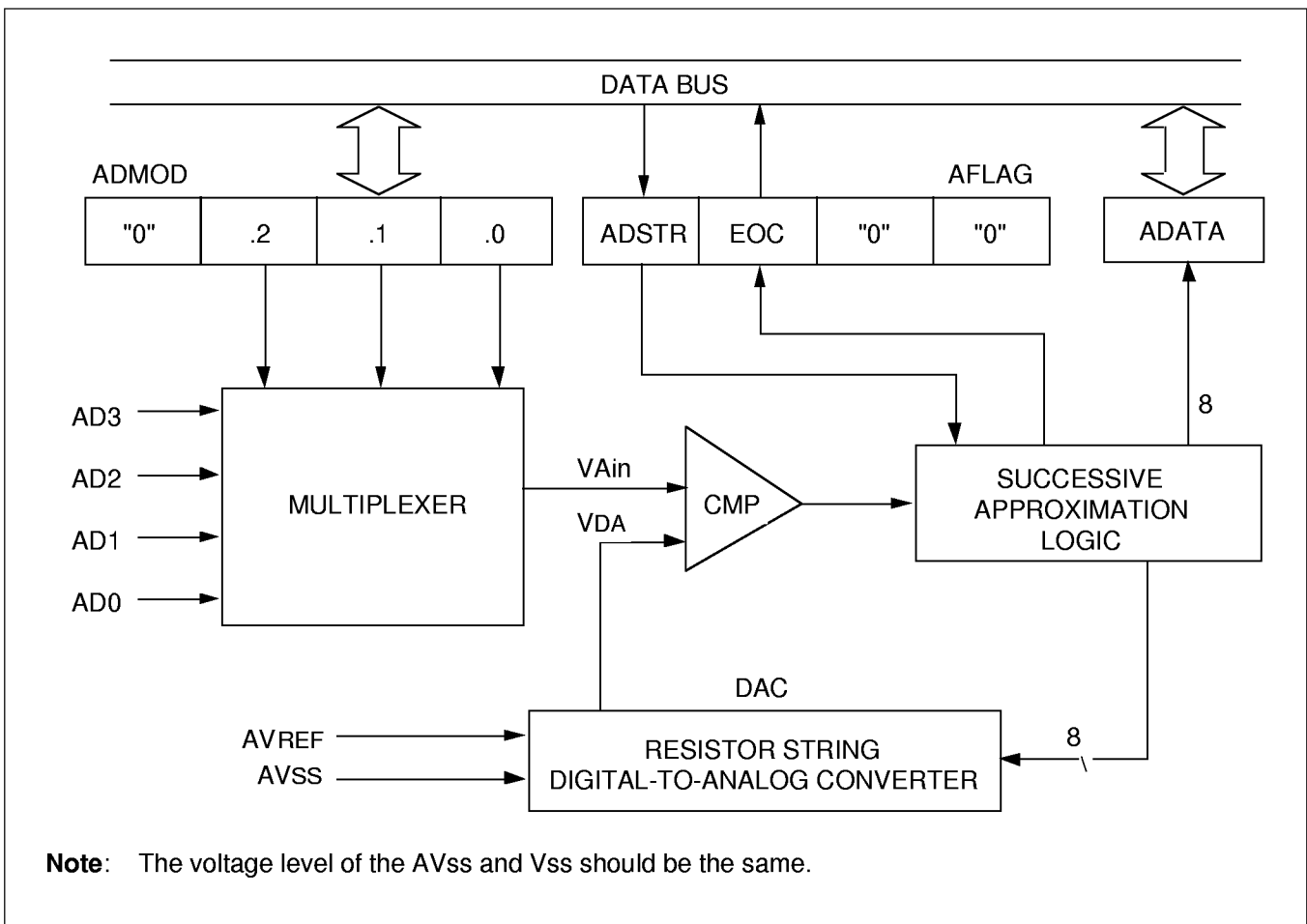


Figure 35. A/D Converter Circuit Diagram

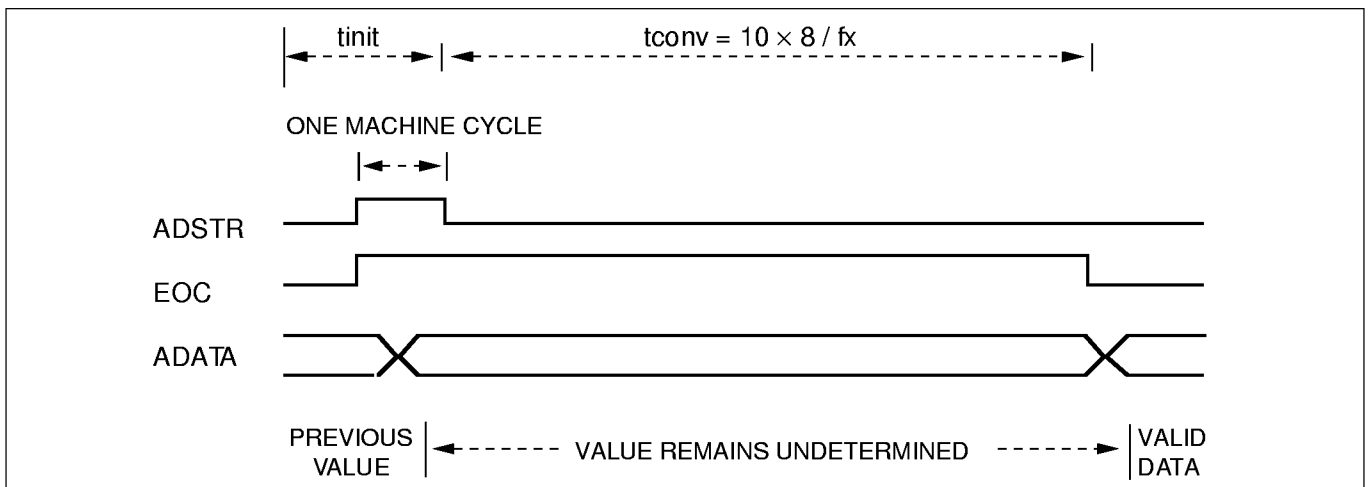


Figure 36. A/D Converter Timing Diagram

**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,  $V_{DA}$ . The  $V_{DA}$  values are input from the DAC to the comparator where they are compared to the multiplexed external analog source voltage,  $V_{Ain}$ . Since the DAC has 8-bit resolution, it generates the 256-step analog reference voltage.

**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 used to select one of four analog channels as the analog data input source. Input channels AD0–AD3 may be used either for analog input to the A/D converter, or as normal input ports. Because 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. Bit 3 in the ADMOD register is always "0".

Table 25. A/D Converter Mode Register Settings (1, 4-Bit R/W)

| ADMOD.2 | ADMOD.1 | ADMOD.0 | Effect of ADMOD Bit Setting |
|---------|---------|---------|-----------------------------|
| 1       | 0       | 0       | Select input channel AD0    |
| 0       | 0       | 1       | Select input channel AD1    |
| 0       | 1       | 0       | Select input channel AD2    |
| 0       | 1       | 1       | Select input channel AD3    |

**NOTE:** If ADMOD.2–ADMOD.0 = 0, disable analog input channel selection.

**ADC CONTROL REGISTER (AFLAG)**

The A/D converter control register, AFLAG, contains the control flags used to start the A/D converter and to monitor its operational status.

| FDBH  |     |     |     |
|-------|-----|-----|-----|
| ADSTR | EOC | "0" | "0" |

A conversion is started by setting the ADSTR bit in the AFLAG register. ADSTR is write-only and is 1-bit and 4-bit addressable. The EOC flag (End Of Conversion) can be read to determine the status the conversion operation.

When a conversion is completed, EOC is to indicate that the conversion result can now be read. EOC can also be set by an ADSTR manipulation. While EOC is "1", the ADC cannot start a new conversion. 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 AD0, AD1 and AD2 pins are used as A/D input pins and the P9.3 / AD3 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              ; AD0 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     AD0BUF,EA             ; AD0BUF ← AD0 conversion data
        LD     A,#1H
        LD     ADMOD,A              ; AD1 pin select for A/D conversion
        BITS   ADSTR                ; A/D conversion start
AD1CK    BTSF   EOC                  ; A/D conversion end check
        JR     AD1CK                ; A/D conversion not completed
        LD     EA,ADATA              ; AD conversion end
        LD     AD1BUF,EA             ; AD1BUF ← AD1 conversion data
        LD     A,#2H
        LD     ADMOD,A              ; AD2 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     AD2BUF,EA             ; AD2BUF ← AD2 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.

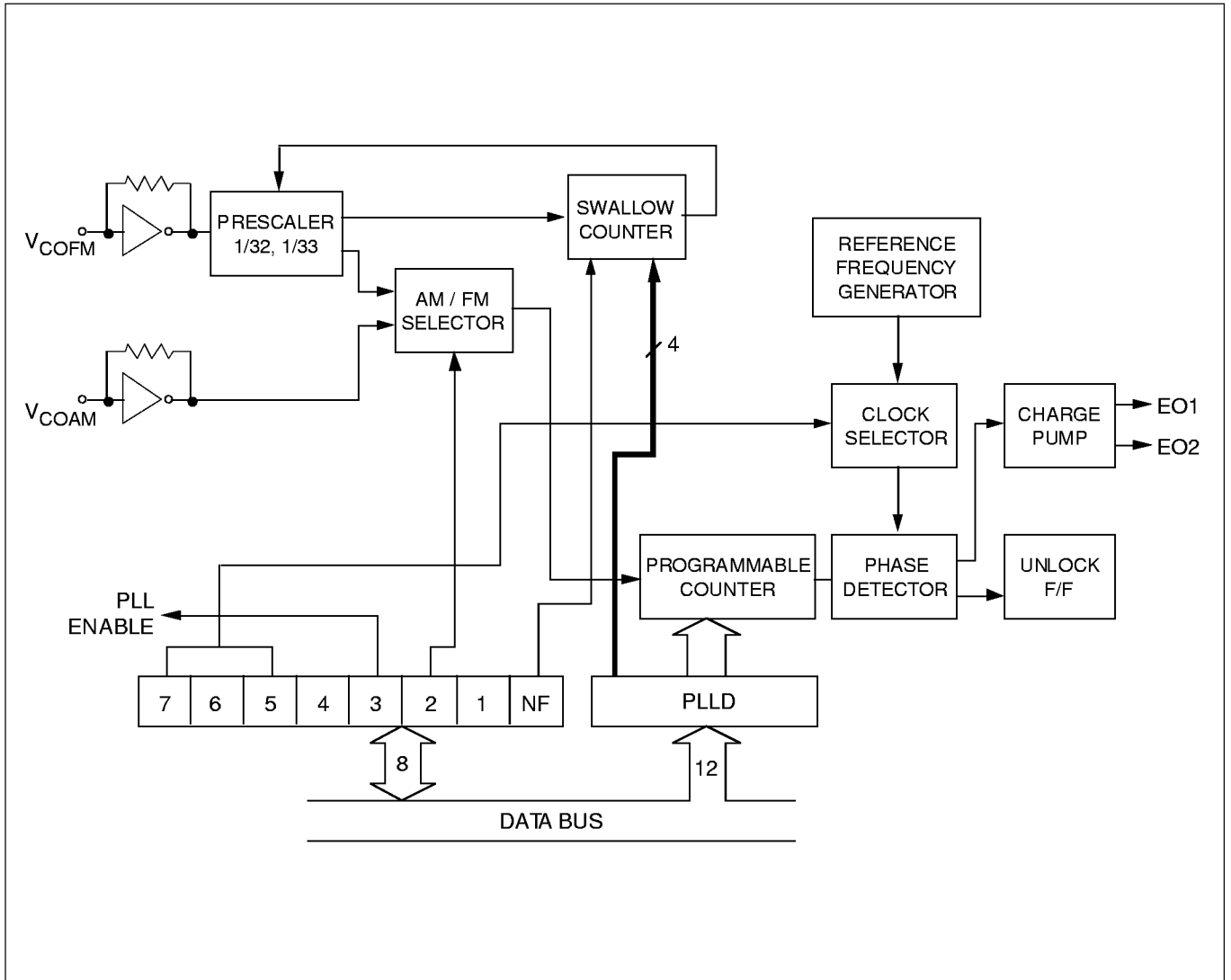


Figure 37. PLL Frequency Synthesizer Block Diagram

**PLL FREQUENCY SYNTHESIZER FUNCTIONS**

The PLL frequency synthesizer divides the signal frequency at the V<sub>COAM</sub> or V<sub>COFM</sub> pin using the programmable divider. It then outputs the phase difference between the divided frequency and reference frequency at the EO1 and EO2 pins.

**NOTE**

The PLL frequency synthesizer operates only when the CE pin is high level; it enters the disable mode when the CE pin is low.

**Input Selection Circuit**

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.

**Programmable Divider**

The programmable divider divides the frequency of the signal from the V<sub>COAM</sub> and V<sub>COFM</sub> pins in accordance with the values contained in the swallow counter and 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 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). The PLLD register is manipulated using 4-bit and 8-bit RAM control instructions.

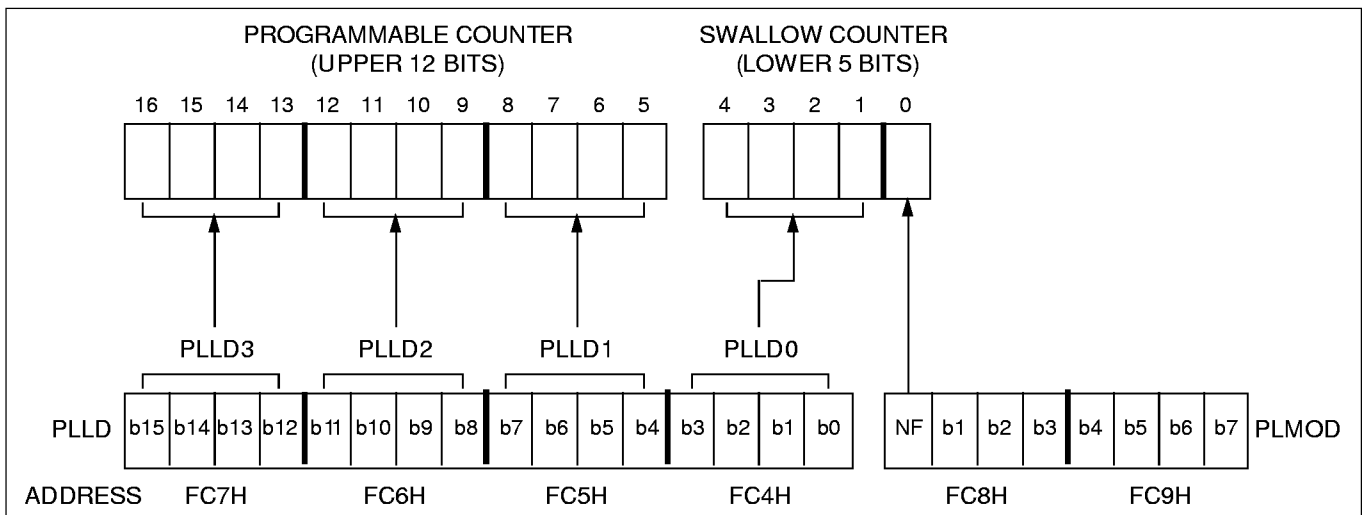


Figure 38. PLLD Register Configuration

**Frequency Division Systems**

In the direct frequency division system, the upper 12 bits are valid. In the pulse swallow system, 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 systems, as set in the PLL data register, are shown below:

- Direct frequency division (AM)

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

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

- Pulse swallow system (FM)

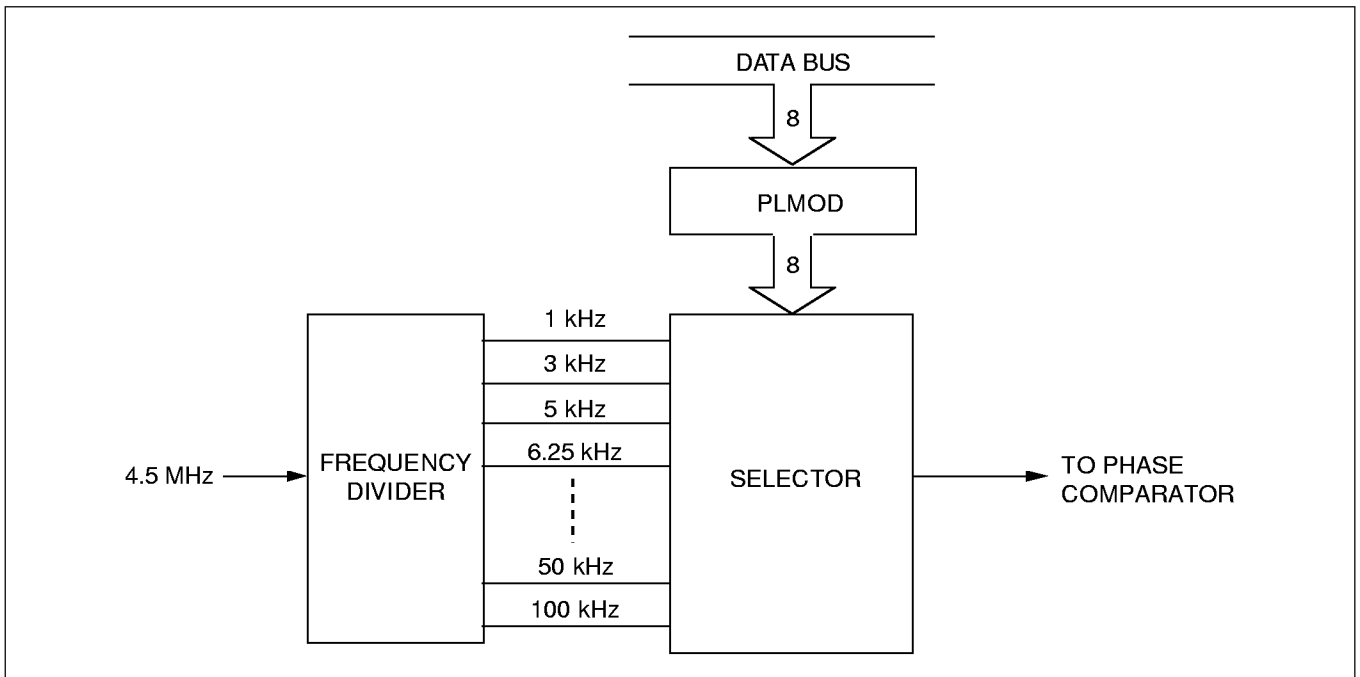
$$fr = \frac{fV_{COFM}}{N}$$

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

**REFERENCE FREQUENCY GENERATOR**

The reference frequency generator produces reference frequencies which are then compared by the phase comparator.

The reference frequency generator divides a crystal oscillation frequency of 4.5MHz and generates the reference frequency (fr) for the PLL frequency synthesizer.



**Figure 39. Reference Frequency Generator**

**PLL MODE REGISTER (PLMOD)**

The PLL mode register (PLMOD) is used to start and stop PLL operation.

|               |         |         |         |         |
|---------------|---------|---------|---------|---------|
| <b>PLMOD1</b> | PLMOD.3 | PLMOD.2 | 0       | NF      |
| <b>PLMOD2</b> | PLMOD.7 | PLMOD.6 | PLMOD.5 | PLMOD.4 |

**Table 26. PLMOD Organization (4-Bit W)**

**PLL Enable Bit**

|                |   |              |
|----------------|---|--------------|
| <b>PLMOD.2</b> | 0 | PLL disabled |
|                | 1 | PLL enabled  |

**Frequency Division Method Selection Bit**

| PLMOD.3 | Frequency Division Method   | Selected Pin   | Input Voltage        | Input Frequency | Division Value                           |
|---------|-----------------------------|--|----------------------|-----------------|--|
| 0       | Direct method for AM        | V <sub>COAM</sub> selected;<br>V <sub>COFM</sub> pulled down | 300 mV <sub>PP</sub> | 0.5 – 30 MHz    | 16 to (2 <sup>12</sup> – 1)              |
| 1       | Pulse swallow method for FM | V <sub>COFM</sub> selected;<br>V <sub>COAM</sub> pulled down | 300 mV <sub>PP</sub> | 15 – 200 MHz    | 2 <sup>10</sup> to (2 <sup>17</sup> – 2) |

| PLMOD.7 | PLMOD.6 | PLMOD.5 | PLMOD.4 | Selected Reference Frequency |
|---------|---------|---------|---------|------------------------------|
| 0       | 0       | 0       | 0       | 1 kHz                        |
| 0       | 0       | 0       | 1       | 3 kHz                        |
| 0       | 0       | 1       | 0       | 5 kHz                        |
| 0       | 0       | 1       | 1       | 6.25 kHz                     |
| 0       | 1       | 0       | 0       | 9 kHz                        |
| 0       | 1       | 0       | 1       | 10 kHz                       |
| 0       | 1       | 1       | 0       | 12.5 kHz                     |
| 0       | 1       | 1       | 1       | 25 kHz                       |
| 1       | 0       | 0       | 0       | 50 kHz                       |
| 1       | 0       | 0       | 1       | 100 kHz                      |

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



**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 ( $f_r$ ) output from the reference frequency generator.

The charge pump outputs the phase comparator's output from error output pins EO1 and EO2. The relation between the error output pin output, divided frequency  $f_N$ , and reference frequency  $f_r$  is shown below:

- $f_r > f_N =$  Low level output
- $f_r < f_N =$  High level output
- $f_r = f_N =$  Floating level

When PLL operation is started by setting PLMOD register, PLL unlock flag (ULFG) in the PLL flag register (PLLREG) has unlock state information between 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 unlock state. If ULFG = "0", the PLL lock state is selected.

|        |      |       |      |
|--------|------|-------|------|
| PLLREG |      |       | F9DH |
| ULFG   | CEFG | IFCFG | 0    |

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

PLL operation is decided by CE (chip enable) pin state. The PLL frequency synthesizer is disabled and the error output pin is set to floating state while the CE pin is low. When CE pin is high level, PLL is operating normally.

The chip enable flag (CEFG) in the PLLREG register has information about CE pin state. When the CE pin changes its low state to high, CEFG flag is set to logic one and CE reset operation occurs. When the CE pin changes its high state to low, CEFG flag is set to logic zero and 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, the following selections must be made 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 bit 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. Calculate the frequency division value as follows:

$$N = \frac{f_{V_{COAM}}}{f_r}$$

Where  $f_{V_{COAM}}$  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 frequency: 9 kHz  
 Intermediate frequency: + 450 kHz

The frequency division value  $N$  is calculated as follows:

$$N = \frac{f_{V_{COAM}}}{f_r} = \frac{1422+450}{9} = 208 \text{ (decimal)}$$

$$= 0D0H \text{ (hexadecimal)}$$

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 0 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 bit 3.
2. The V<sub>COFM</sub> 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:

$$N = \frac{f_{V_{COFM}}}{f_r}$$

where  $f_{V_{COFM}}$  is the input frequency at the V<sub>COFM</sub> pin, and  $f_r$  is the reference frequency

**Example:**

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

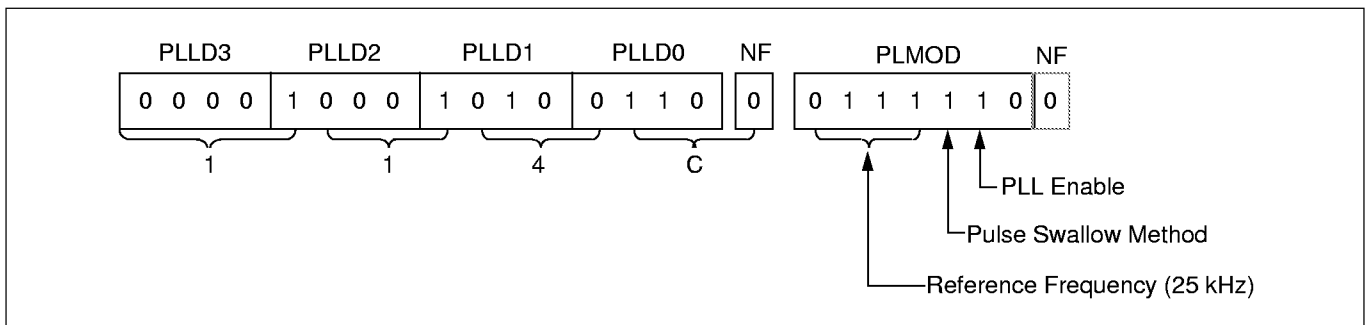
- Receive frequency: 100.0 MHz
- Reference frequency: 25 kHz
- Intermediate frequency: 10.7 MHz

The frequency division value N is calculated as follows:

$$N = \frac{f_{V_{COFM}}}{f_r} = \frac{(100.0+10.7) \times 10^6}{25 \times 10^3} = 4428 \text{ (decimal)}$$

$$= 114CH \text{ (hexadecimal)}$$

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



In the above example, each time NF bit value (LSB) is inverted, V<sub>CO</sub> oscillation frequency varies by 25 kHz.

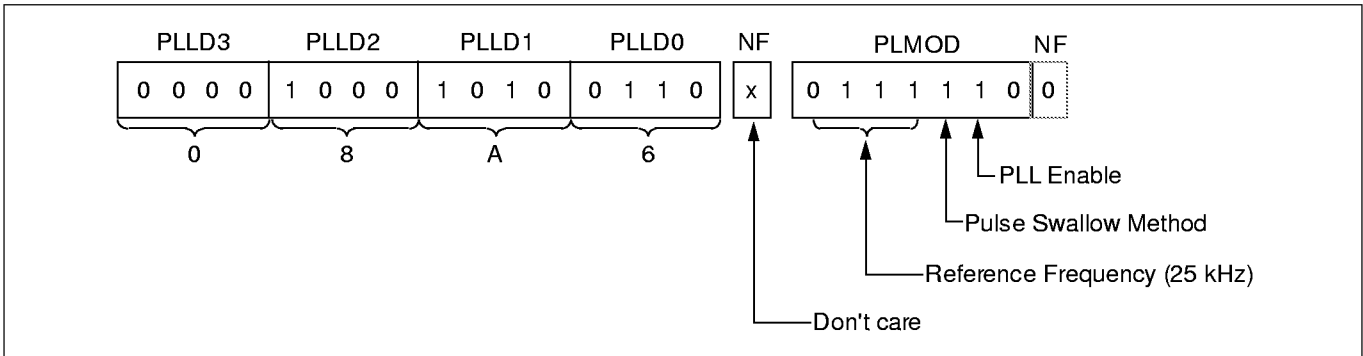
To simplify programming, it is therefore useful not to use the NF bit. In this way, you can easily calculate the reference frequency in 50-kHz (2 x 25 kHz) multiples.

**Pulse Swallow Method (Continued)**

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

$$N = \frac{f_{VCOFM}}{fr} = \frac{(100.0+10.7) \times 10^6}{2 \times 25 \times 10^3} = 2214 \text{ (decimal)}$$

$$= 8A6H \text{ (hexadecimal)}$$



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 KS57C3016 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 2. When the AMIF pin input signal is directly connected to the IFC, it is not divided.

By setting the 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.

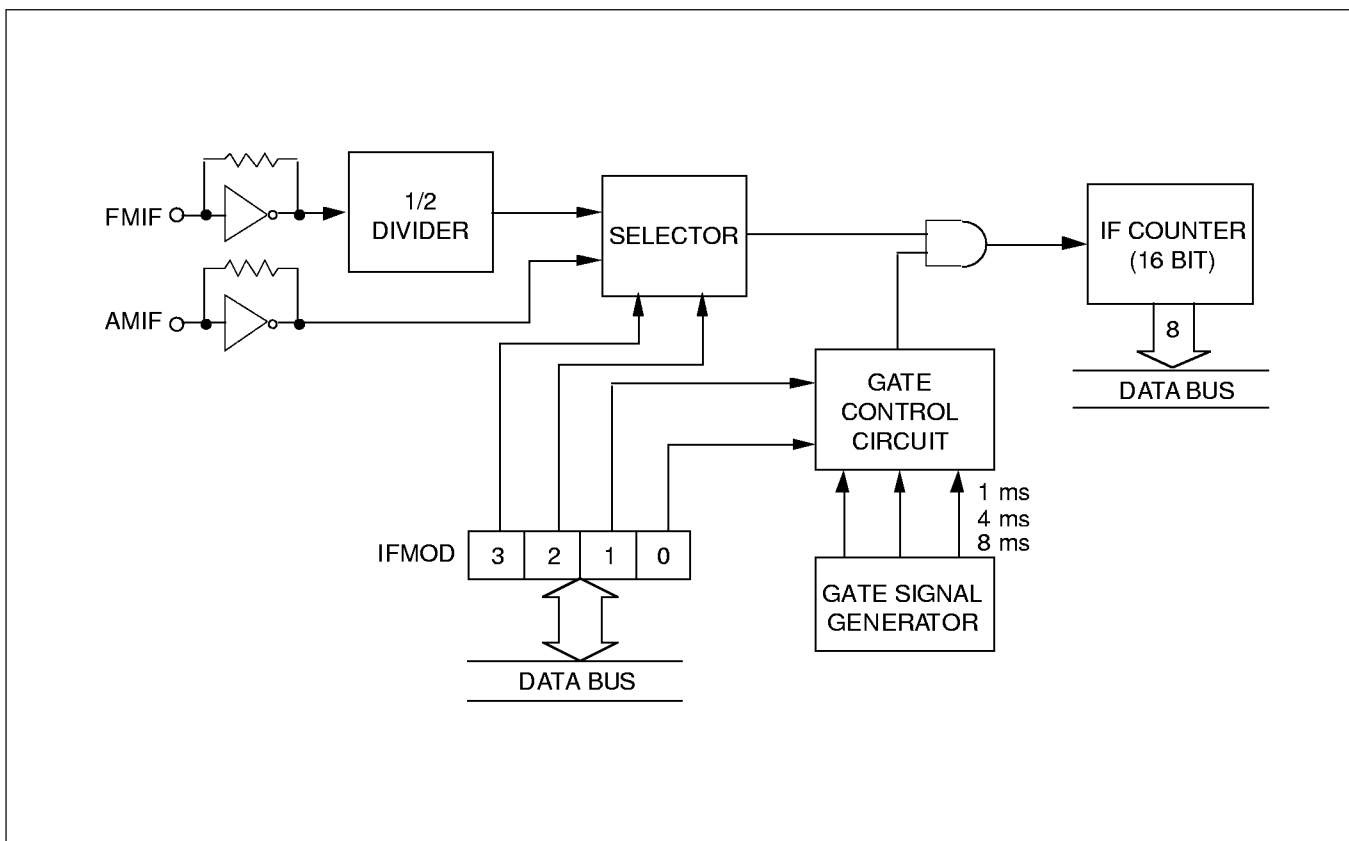


Figure 40. IF Counter Block Diagram

**IFC MODE REGISTER (IFMOD)**

The IFC mode register (IFMOD) 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.

**Table 27. IFMOD Organization (4-Bit R/W)****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 ms               |
| 0       | 1       | 4 ms               |
| 1       | 0       | 8 ms               |
| 1       | 1       | Open               |

**PLL FLAG REGISTER (PLLREG)**

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.

| PLLREG |      |       | 4-Bit R |
|--------|------|-------|---------|
| ULFG   | CEFG | IFCFG | 0       |

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

**GATE TIMES (1-ms, 4-ms, and 8-ms)**

After setting IFMOD, the IFC gate is opened for a 1-ms, 4-ms, or 8-ms period, 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 n shows gate timings with a 1 kHz internal clock.

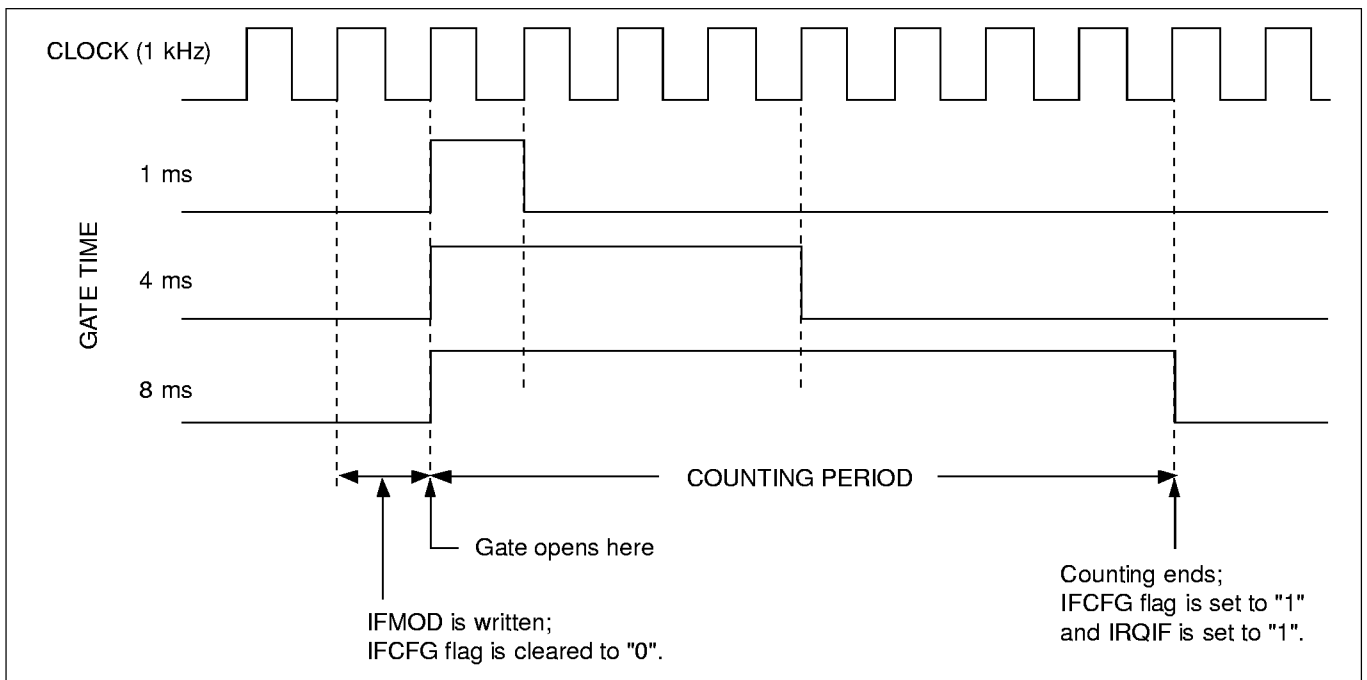


Figure 41. Gate Timing (1, 4, or 8 ms)

**Gate Time Selected as Open**

If "open" is selected by setting IFMOD and IFC counts the input signal during the open period of the gate, the gate will be closed at when IFMOD is rewritten.

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

- Set the gate time to non-open (1-, 4-, or 8-ms) by bits IFMOD.1 and IFMOD.0.
- Disable IFC operation by clearing bits IFMOD.3 and IFMOD.2 to "0". This method lets the gate stay open, and stops the counting operation.

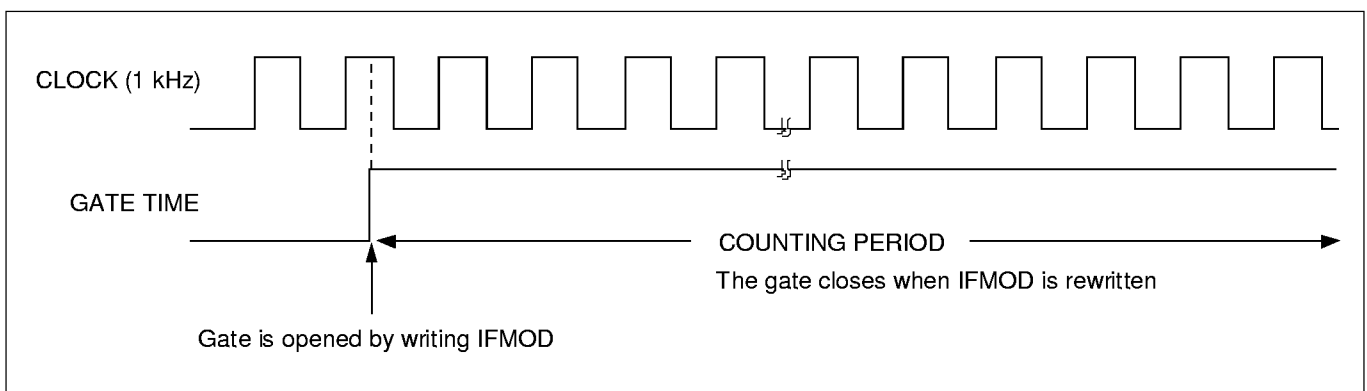
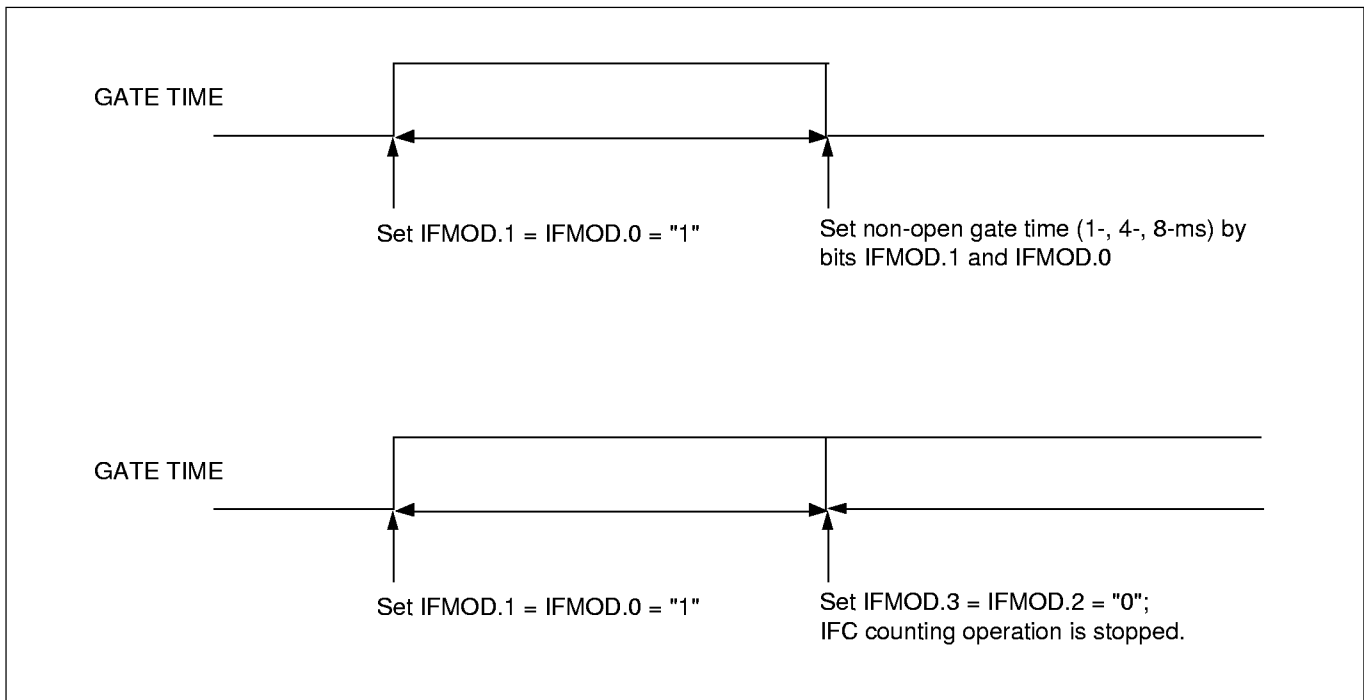


Figure 42. Gate Timing (Open)



**Figure 43. Gate Timing for Non-Open and Stop Counting Operation**

### Gate Time Errors

A gate time error occurs when the gate signals are not synchronized to 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 38).

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.

### 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".



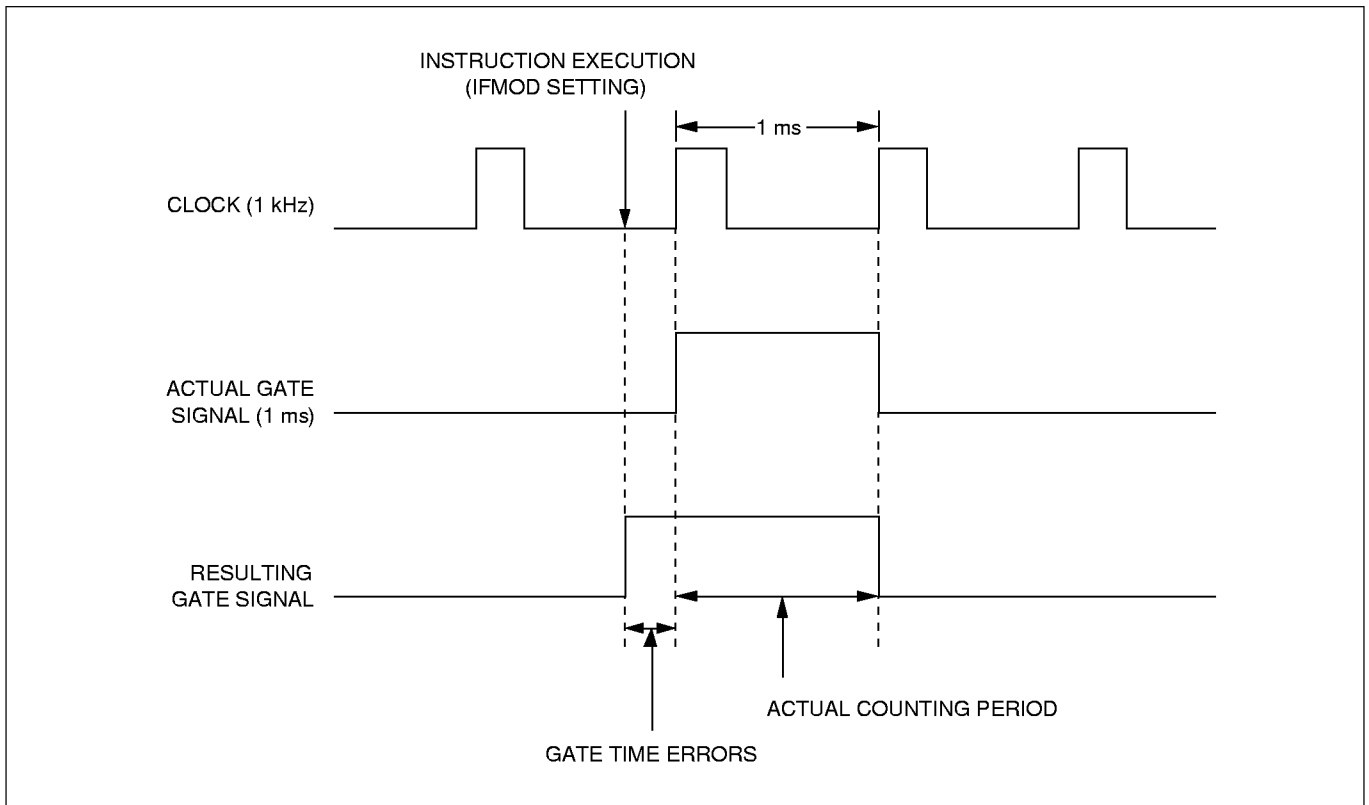


Figure 44. Gate Error Timing (Open)

**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 writing "0s" to IFMOD.2 and IFMOD.3.

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. Reset clears the counter to zero.

|               |          |          |          |          |          |          |          |          |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>IFCNT0</b> | IFCNT0.7 | IFCNT0.6 | IFCNT0.5 | IFCNT0.4 | IFCNT0.3 | IFCNT0.2 | IFCNT0.1 | IFCNT0.0 |
| <b>IFCNT1</b> | IFCNT1.7 | IFCNT1.6 | IFCNT1.5 | IFCNT1.4 | IFCNT1.3 | IFCNT1.2 | IFCNT1.1 | IFCNT1.0 |

**IF COUNTER (IFC) OPERATION (Continued)**

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:

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

when  $T_G$  = Gate time (1 ms, 4 ms, 8 ms)

— AMIF pin input frequency:

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

when  $T_G$  = Gate time (1 ms, 4 ms, 8 ms)

**Table 28. IF Counter Frequency Ranges**

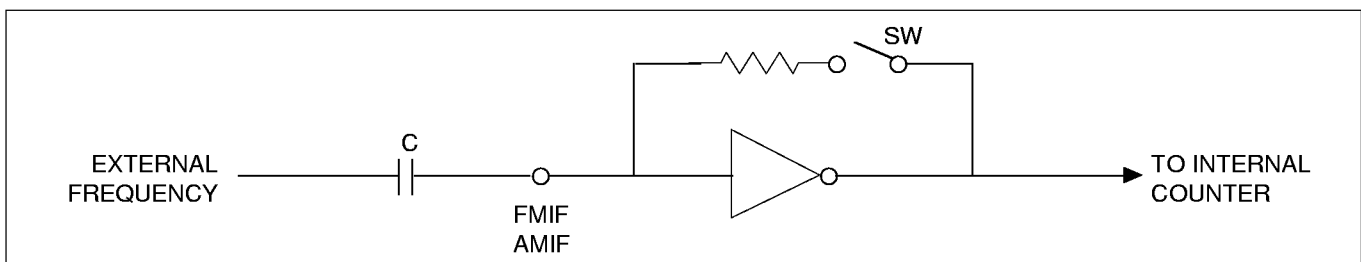
| Pin  | Voltage Level   | Frequency Range  |
|------|-----------------|------------------|
| AMIF | 300 m Vpp (min) | 0.1 MHz to 1 MHz |
| FMIF | 300 m Vpp (min) | 5 MHz to 15 MHz  |

**INPUT PIN CONFIGURATION**

The AMIF and FMIF pins have built-in AC amplifiers (see Figure 44). 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 45. 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     IFCFG    ; 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 frequency.

FMIF input frequency ( $f_{FMIF}$ ): 10.7 MHz

Gate time ( $T_G$ ): 8 ms

IFC counter value (N):

$$\begin{aligned}
 N &= (f_{FMIF} / 2) \times T_G \\
 &= 10.7 \times 10^6 / 2 \times 8 \times 10^{-3} \\
 &= 42800 \\
 &= A720H
 \end{aligned}$$

|              |        |   |   |   |   |   |   |   |        |   |   |   |   |   |   |   |
|--------------|--------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| <b>Bin</b>   | 1      | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0      | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| <b>Dec</b>   | A      |   |   |   | 7 |   |   |   | 2      |   |   |   | 0 |   |   |   |
| <b>IFCNT</b> | 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):

$$\begin{aligned}
 N &= (f_{AMIF}) \times T_G \\
 &= 450 \times 10^3 \times 8 \times 10^{-3} \\
 &= 3600 \\
 &= E10H
 \end{aligned}$$

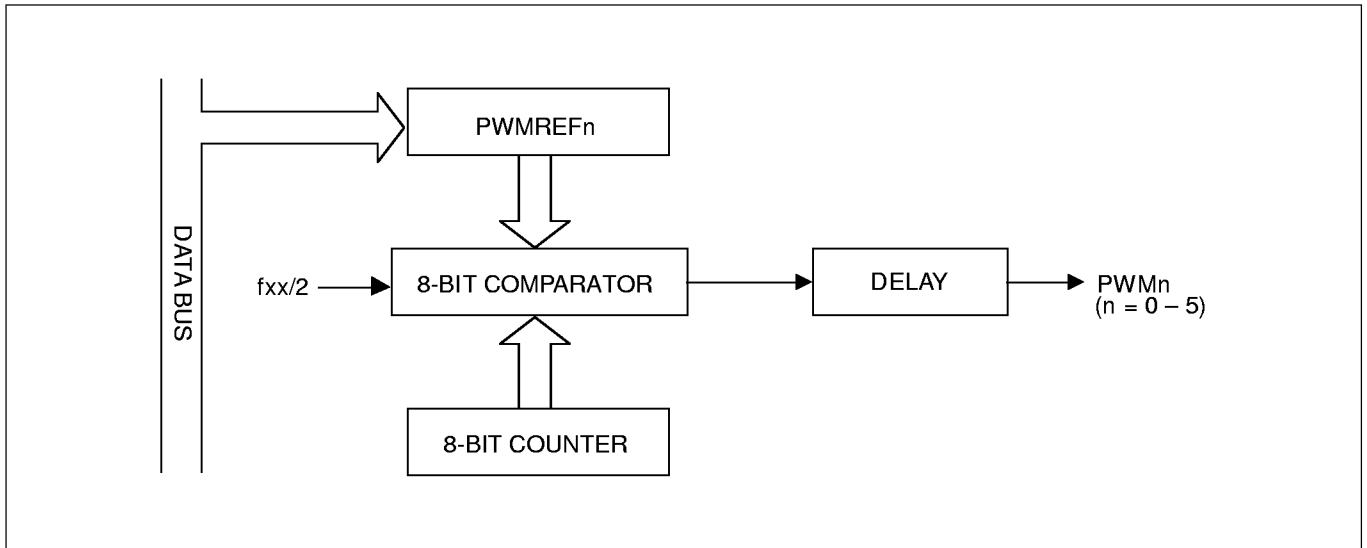
|              |        |   |   |   |   |   |   |   |        |   |   |   |   |   |   |   |
|--------------|--------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|
| <b>Bin</b>   | 0      | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0      | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| <b>Dec</b>   | 0      |   |   |   | E |   |   |   | 1      |   |   |   | 0 |   |   |   |
| <b>IFCNT</b> | IFCNT1 |   |   |   |   |   |   |   | IFCNT0 |   |   |   |   |   |   |   |

**PULSE WIDTH MODULATION**

The KS57C3016 has six pulse width modulated output channels, each with 8-bit resolution. These channels generate pulses of programmable length and interval. Since  $f_{xx}/2$  is used as the clock source, the repeating frequency is calculated as follows:

$$f_{PWM} = \frac{f_{xx}}{2 \times 2^8} \text{ (8789 Hz at 4.5 MHz)}$$

If the system clock has a 4.5 MHz oscillation frequency, the minimum pulse width will be 444 ns.



**Figure 46. PWM Block Diagram**

**PWM MODE REGISTER (PWMOD)**

The 8-bit PMW mode register PWMOD is used to control the PWM outputs. When bit PWMOD.0 is set to "1", PWM output starts. PWM output is stopped, and the counter is cleared when PWMOD.0 is cleared to "0". Clearing PWMOD.0 to "0" forces the PWM output to low level; the PWM reference register (PWMREFn) retains the current value.

All PWM outputs are disabled at once when PWMOD.1 is cleared to "0". This does not stop PWM operation, however, but simply turns off the PWM signal output.

Individual PWM outputs are enabled or disabled by setting PWMOD.2 through PWMOD.7. PWMOD is manipulated using 8-bit RAM control instructions only, except for PWMOD.0 and PWMOD.1, which are also bit-addressable.

Table 29. PWMOD Organization

| Bit Name | Bit Setting | Effect of PWMOD Bit Setting           |
|----------|-------------|---------------------------------------|
| PWMOD.0  | 0           | PWM output stops (counter is cleared) |
|          | 1           | Normal PWM output mode                |
| PWMOD.1  | 0           | Disable all PWM outputs               |
|          | 1           | Enable all PWM outputs                |
| PWMOD.2  | 0           | PWM0 disable                          |
|          | 1           | PWM0 enable                           |
| PWMOD.3  | 0           | PWM1 disable                          |
|          | 1           | PWM1 enable                           |
| PWMOD.4  | 0           | PWM2 disable                          |
|          | 1           | PWM2 enable                           |
| PWMOD.5  | 0           | PWM3 disable                          |
|          | 1           | PWM3 enable                           |
| PWMOD.6  | 0           | PWM4 disable                          |
|          | 1           | PWM4 enable                           |
| PWMOD.7  | 0           | PWM5 disable                          |
|          | 1           | PWM5 enable                           |

### PWM FUNCTION DESCRIPTION

The KS57C3016 PWM module consists of a PWM mode register, an 8-bit counter, six PWM reference registers, and two delay logic circuits. The PWMOD register and the counter are common to all six PWM channels.

The 8-bit counter counts modulus 256, that is, from 0–255, inclusive. The value of the 8-bit counter is compared to the contents of the six PWM reference registers, PWMREF<sub>n</sub> (n = 0–5). When the reference register value equals the counter value, the PWM output goes low. When the counter reaches zero, the PWM output is forced high. The low-to-high ratio (Duty) of the PWM output is  $\text{PWMREF} \div 256$ .

All PWM outputs remain inactive during the first 256 input clock signals. Then, when the counter value changes from FFH back to 00H, the PWM outputs are forced to high level. The pulse width ratio (duty cycle) is defined by the contents of the reference register and is programmed in increments of 1:256. The 8-bit PWM data register is read and written using 8-bit RAM control instructions only.

PWM output can be held at low level by continuously loading the reference register with 00H. By continuously loading the reference register with FFH, you can hold the PWM output to high level, except for the last pulse of the clock source, which sends the output low (see Figure 46).

Table 30. PWM Reference Register Duty Values

| Reference Register Value (PWMREFn) | Duty              |
|------------------------------------|-------------------|
| 0000 0000                          | 0 /256 (0 %)      |
| 0000 0001                          | 1 /256 (0.39 %)   |
| 0000 0010                          | 2 /256 (0.78 %)   |
| ⋮                                  | ⋮                 |
| 1000 0000                          | 128 /256 (50 %)   |
| 1000 0001                          | 129 /256 (50.4 %) |
| ⋮                                  | ⋮                 |
| 1111 1110                          | 254 /256 (99.2 %) |
| 1111 1111                          | 255 /256 (99.6 %) |

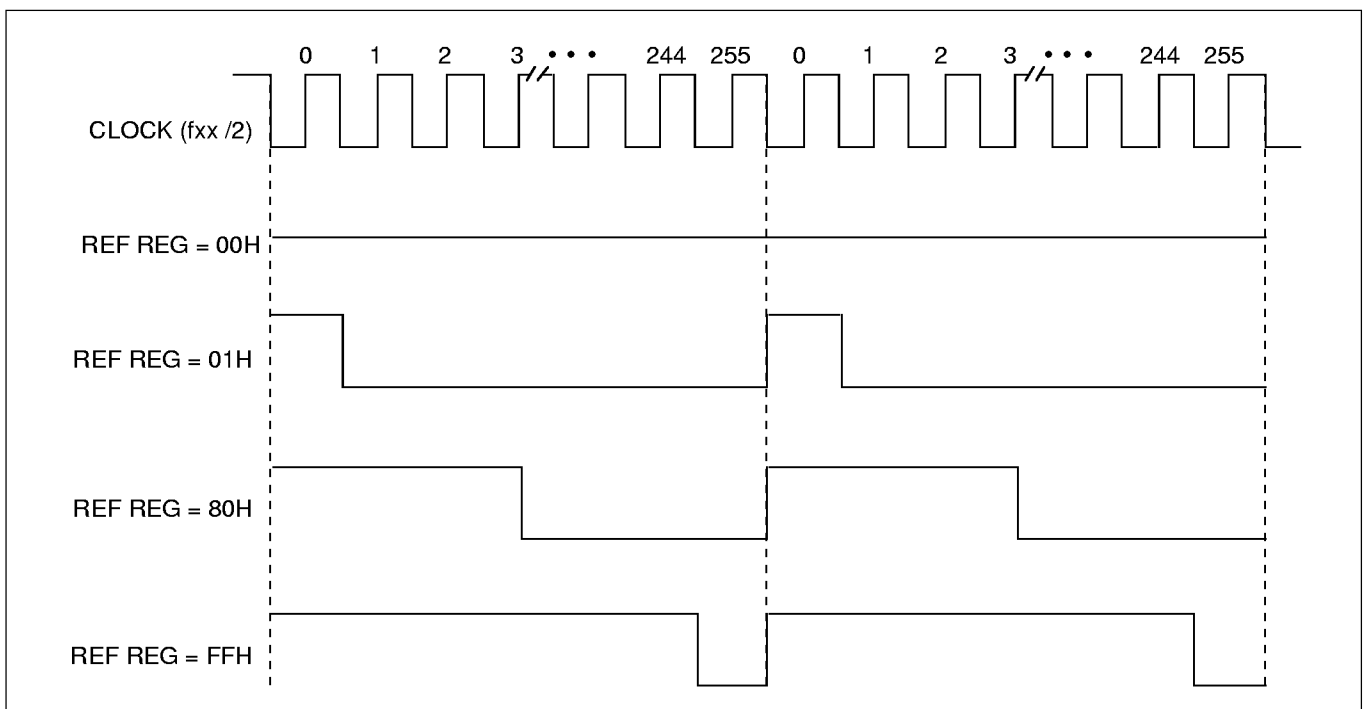


Figure 47. PWM Output Waveforms

## OUTPUT DELAY TIMING

There are two delay circuits for PWM outputs. These circuits are used to reduce noise levels when all six PWM output pins change their states simultaneously. Figure n shows the delay timing for each PWM output pin. PWM0/1, PWM2/3, and PWM4/6 have the same delay times. The system clock (f<sub>xx</sub>) is used for the delay circuit and each pair of PWM outputs are delayed by 1/2 clock pulse.

The PWM output signal starts whenever the counter value is 00H. If you modify PWM mode register or data register values while the PWM module is operating, the modified PWM frequency is output when the counter overflows. PWM output retains the previous duty ratio until the counter completes its current operation and starts the next counting operation.

## PWM OUTPUT PORTS

The PWM outputs are alternate functions assigned to n-channel, open drain I/O port 7 and port 8. Pull-up resistors can be assigned by mask option to these ports. Port 7 and 8 do not support loads greater than 9 volts.

To use ports 7 and 8 as a PWM output, ports 7 and 8 should be configured as output mode and its output data is cleared to zero. To use a PWM output as a normal I/O pin, the PWM output first needs to be disabled by the corresponding PWMOD setting. You should then configure ports 7 and 8 to input or output mode by port mode register. A reset operation initializes ports 7 and 8 to input mode.

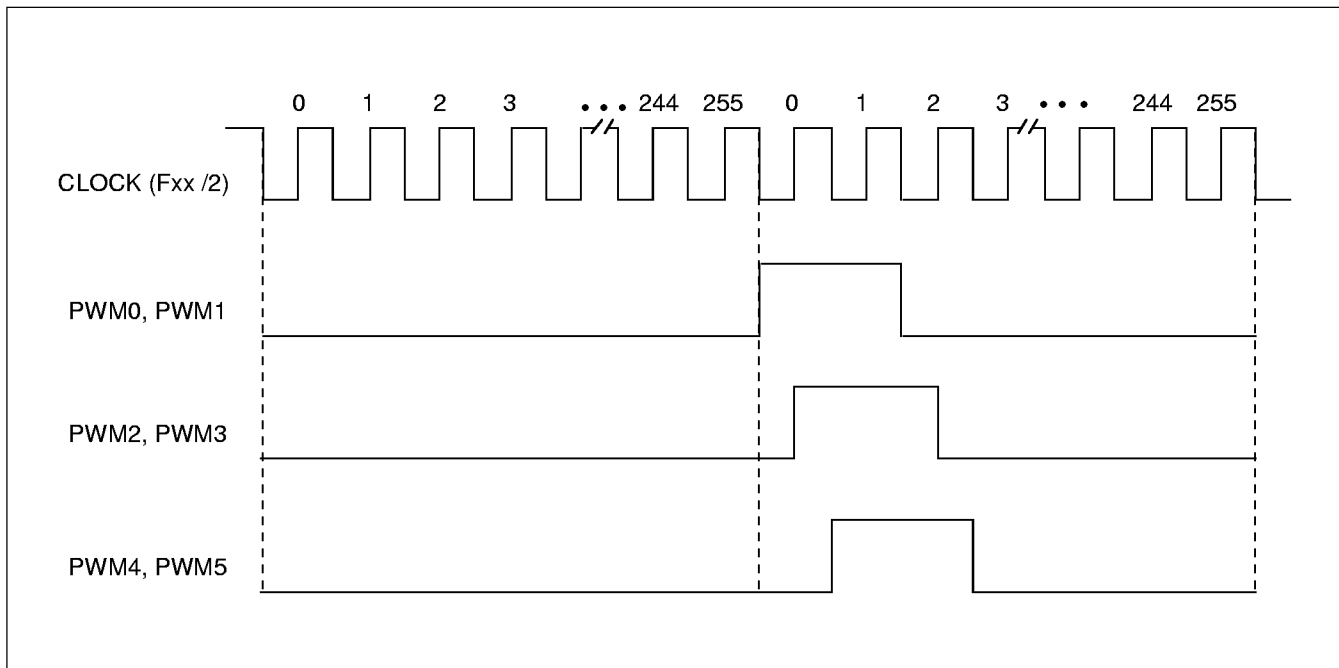


Figure 48. PWM Output Delay Timing



**LCD CONTROLLER/DRIVER**

The KS57C3016 microcontroller can directly drive an up-to-16-digit (128-segment) LCD panel. 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 the Stop1 and Idle power-down modes.

**LCD RAM ADDRESS AREA**

RAM addresses 1E0H–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 are sent out through segment pins SEG0–SEG31 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.

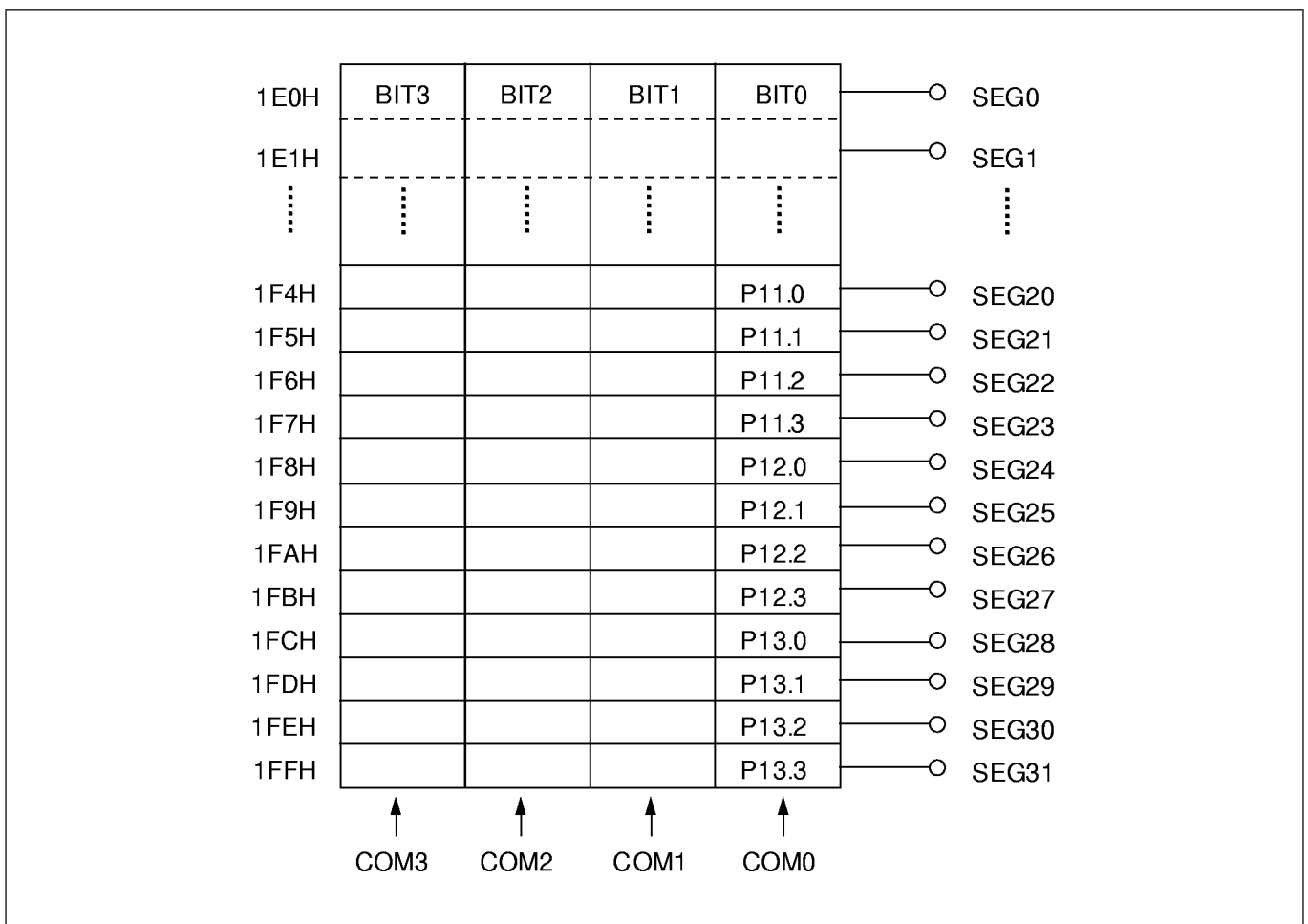


Figure 49. LCD Display Data RAM Organization

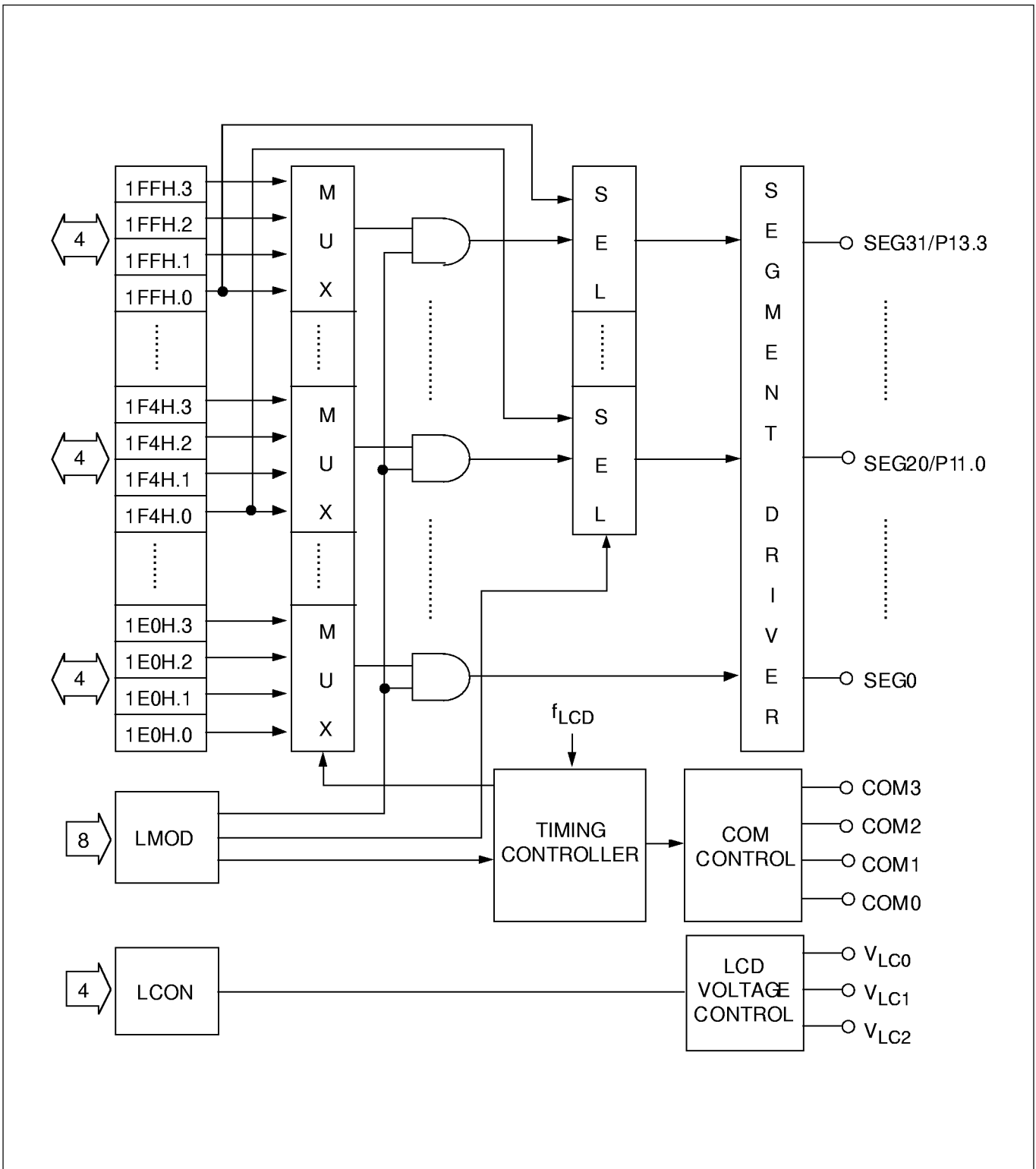


Figure 50. LCD Circuit Diagram

**LCD CONTROL REGISTER (LCON)**

The LCON register is used to turn the LCD display on and off and to control the flow of current to dividing resistors in the LCD circuit. When LCON.0 is logic zero, the LCD display is turned off and the current to the dividing resistors is cut off, regardless of the current LMOD.3 value.

**Table 31. LCD Control Register (LCON) Organization (4-Bit W)**

| LCON Bit | Setting | Description  |
|----------|---------|--|
| LCON.3   | 0       | This bit is used for internal testing only; set to logic zero.   |
| LCON.2   | 0       | Always set to logic zero.  |
| LCON.1   | 0       | Always set to logic zero.  |
| LCON.0   | 0       | LCD output low; turn display off, cut off current to dividing resistors, and output ports 11–13 latch contents.  |
|          | 1       | If LMOD.3 = "0": LCD output low; turn display off; output ports 11–13 latch contents;<br>If LMOD.3 = "1": COM and SEG output in display mode; turn display on output ports 11–13 latch contents. |

**Table 32. Relationship of LCON.0 and LMOD.3 Bit Settings**

| LCON.0 | LMOD.3 | COM0–COM3                              | SEG0–SEG31                             | P11.0–P13.3  |
|--------|--------|--|--|--|
| 0      | x      | Output low; LCD display off            | Output low; LCD display off            | Output latch contents; cut off current to dividing resistors |
| 1      | 0      | Output low; LCD display off            | Output low; LCD display off            | Output latch contents; LCD display off                       |
|        | 1      | COM output corresponds to display mode | SEG output corresponds to display mode | Output latch contents; LCD display on                        |

**NOTE:** 'x' means 'don't care.'

**LCD MODE REGISTER (LMOD)**

The LCD mode control register LMOD is used to control display mode; LCD clock, segment or port output, and display on/off. 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 when the LCD display is turned on. The LCD display can continue to operate during Idle and Stop modes if a subsystem clock is used as the watch timer source.

**Table 33. 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:**

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

**Table 34. Maximum Number of Display Digits Per Duty Cycle**

| LCD Duty | LCD Bias | COM Output Pins | Maximum Digit Display<br>(× 8 Segment Pins) |
|----------|----------|-----------------|---|
| Static   | Static   | COM0            | 4   |
| 1/2      | 1/2      | COM0–COM1       | 8   |
| 1/3      | 1/2      | COM0–COM2       | 12  |
| 1/3      | 1/3      | COM0–COM2       | 12  |
| 1/4      | 1/3      | COM0–COM3       | 16  |

**Table 35. LCD Mode Control Register (LMOD) Organization (8-Bit W)**

| LMOD.7 | LMOD.6 | LCD Output Segments and 1-Bit Output Pins                      |
|--------|--------|--|
| 0      | 0      | Segments 20–23, 24–27, and 28–31                               |
| 0      | 1      | Segments 20–23 and 24–27; 1-bit output at P13.0–P13.3          |
| 1      | 0      | Segments 20–23; 1-bit output at P12.0–P12.3 and P13.0–P13.3    |
| 1      | 1      | 1-bit output only at P11.0–P11.3, P12.0–P12.3, and P13.0–P13.3 |

| 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'.

**LCD DRIVE VOLTAGE**

The LCD display is turned on only when the voltage difference between the common and segment signals is greater than  $V_{LCD}$ . The LCD display is turned off when the difference between the common and segment signal voltages is less than  $V_{LCD}$ .

**NOTE**

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

**LCD VOLTAGE DIVIDING RESISTORS**

On-chip voltage dividing resistors for the LCD circuit can be configured by mask option selection. Figure n 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.

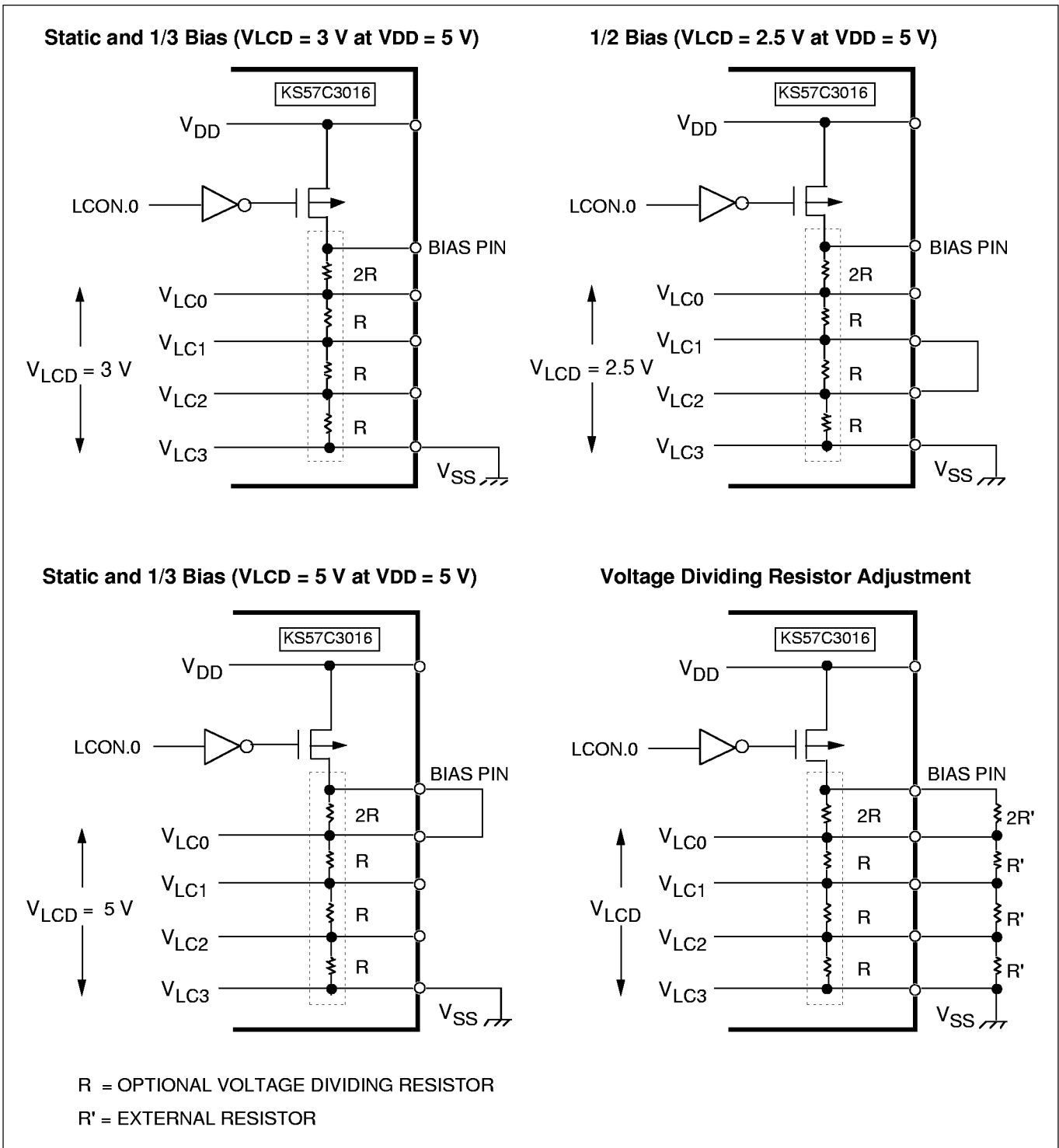


Figure 51. Voltage Dividing Resistor Circuit Diagrams

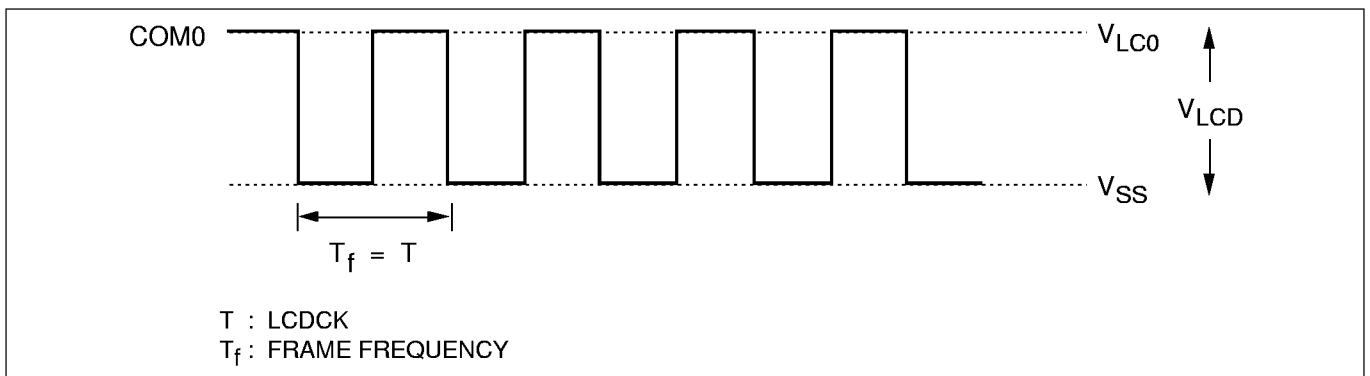
**COMMON (COM) SIGNALS**

The common signal output pin selection (COM pin selection) varies according to the selected duty cycle.

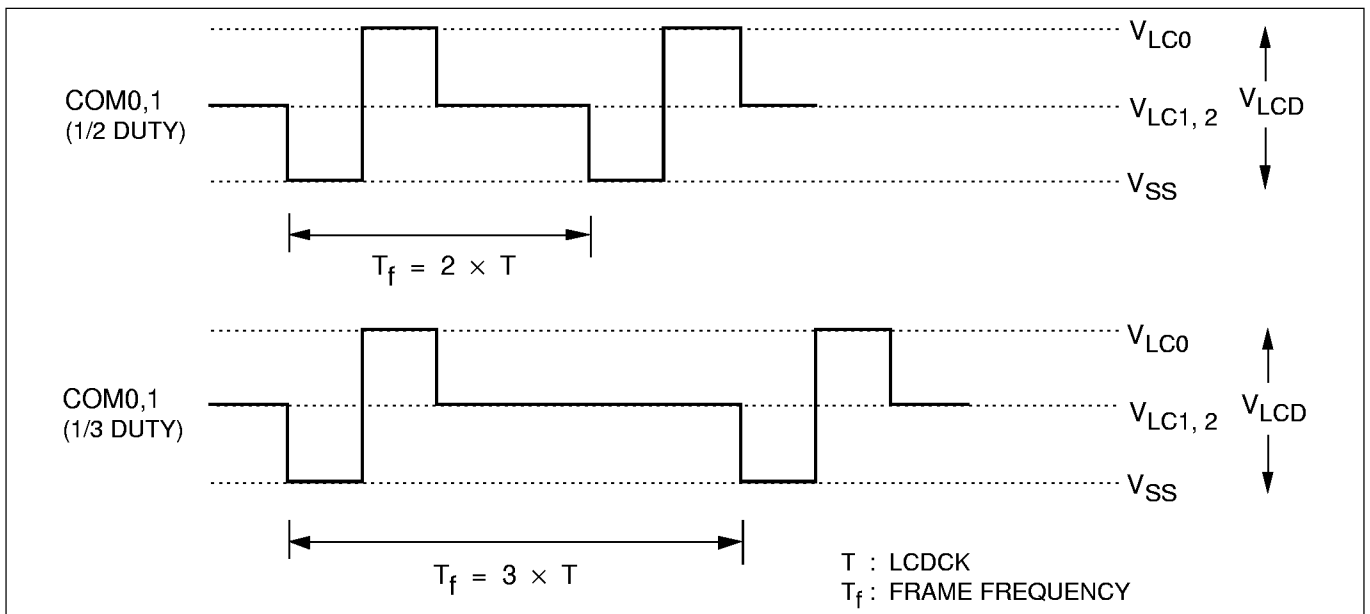
**Table 36. Common Signal Pins Used Per Duty Cycle**

| 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 52. LCD Common Signal Waveform (Static)**



**Figure 53. LCD Common Signal Waveforms at 1/2 Bias (1/2, 1/3 Duty)**

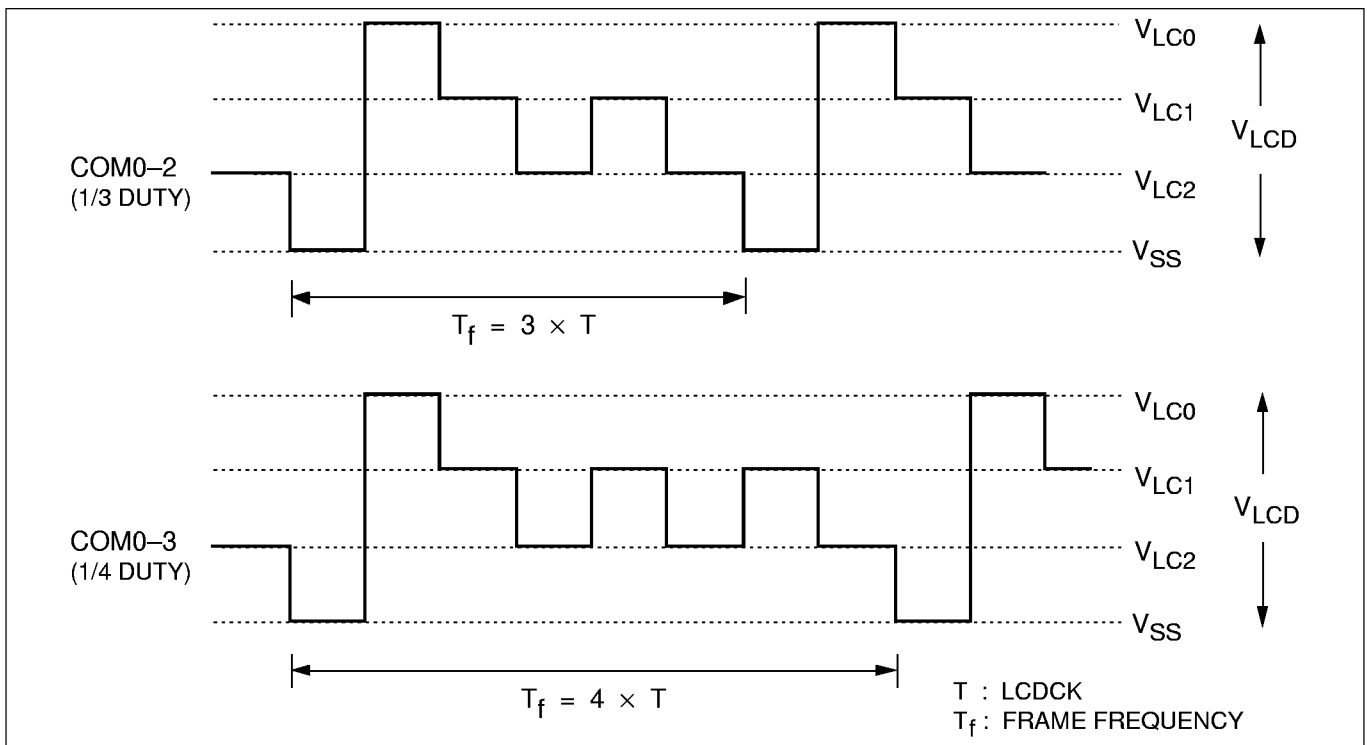


Figure 54. LCD Common Signal Waveforms at 1/3 Bias (1/3, 1/4 Duty)

**SEGMENT (SEG) SIGNALS**

The 40 LCD segment signal pins are connected to corresponding display RAM locations at 1E0H-1FFH. Bits 0-3 of the display RAM are synchronized with the common signal output pins COM0, COM1, COM2, and COM3.

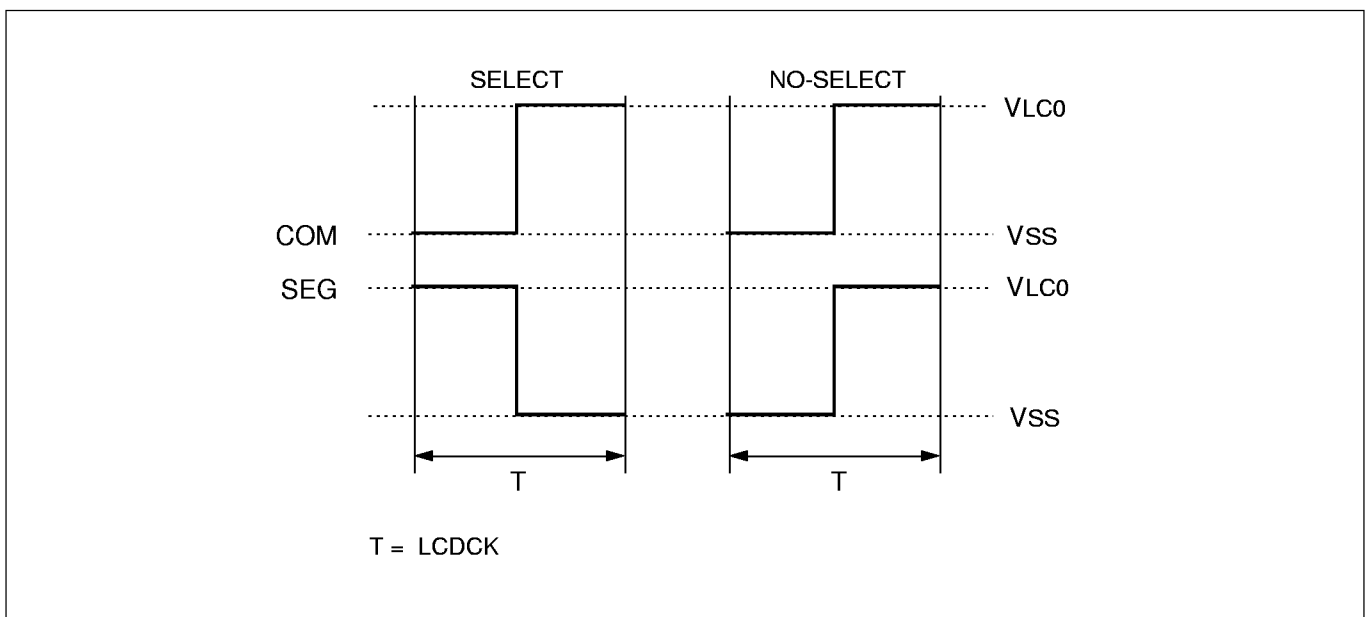


Figure 55. Select/No-Select Bias Signals in Static Display Mode



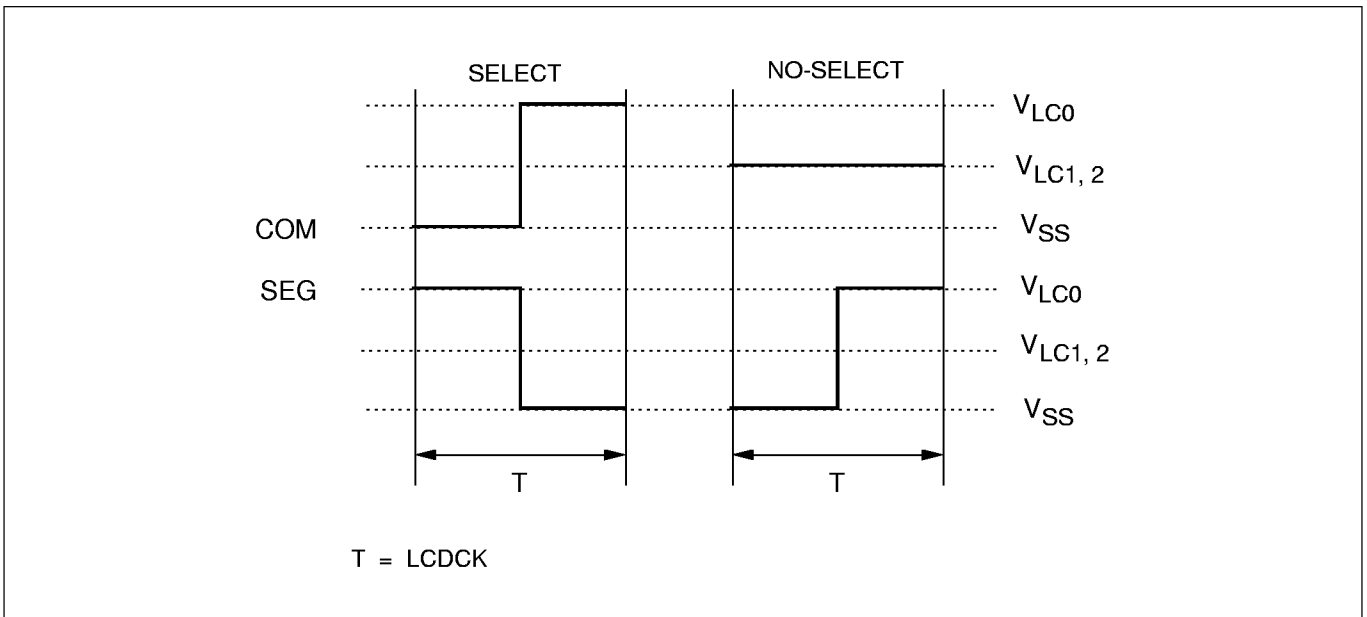


Figure 56. Select/No-Select Bias Signals in 1/2 Bias Display Mode

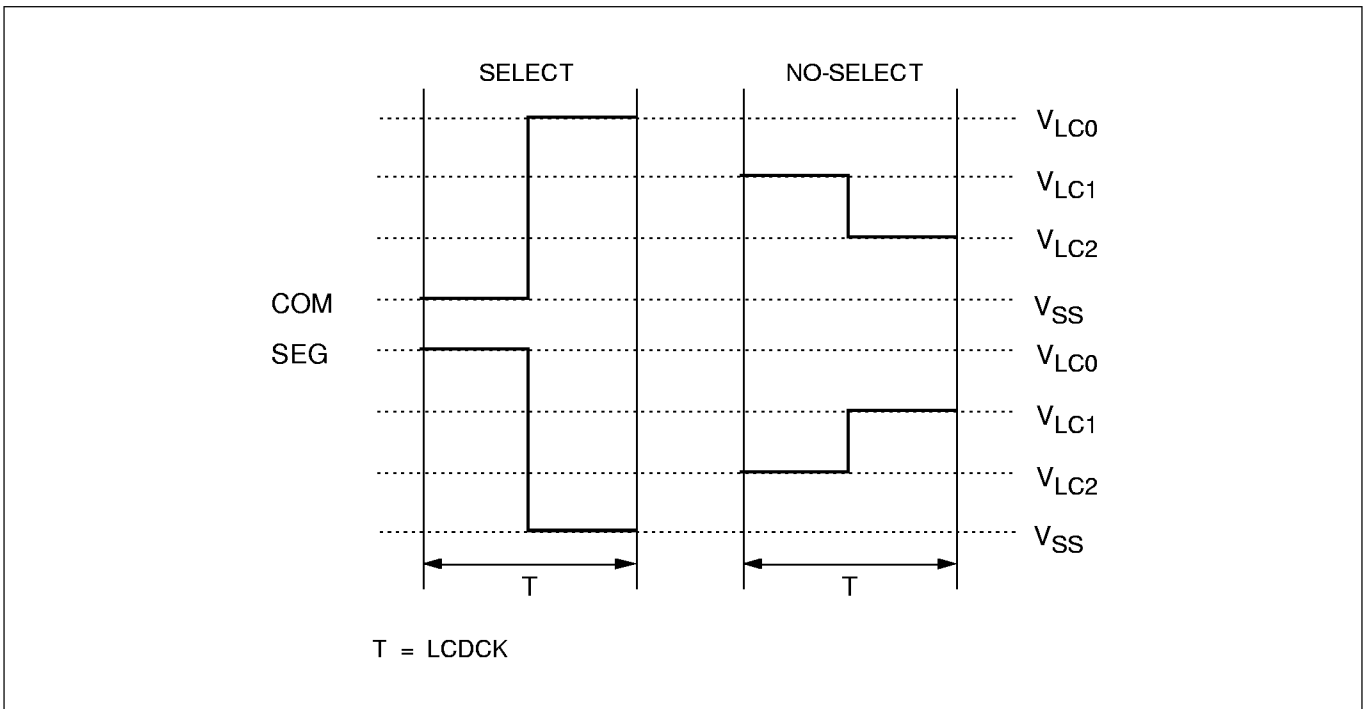


Figure 57. Select/No-Select Signals in 1/3 Bias Display Mode

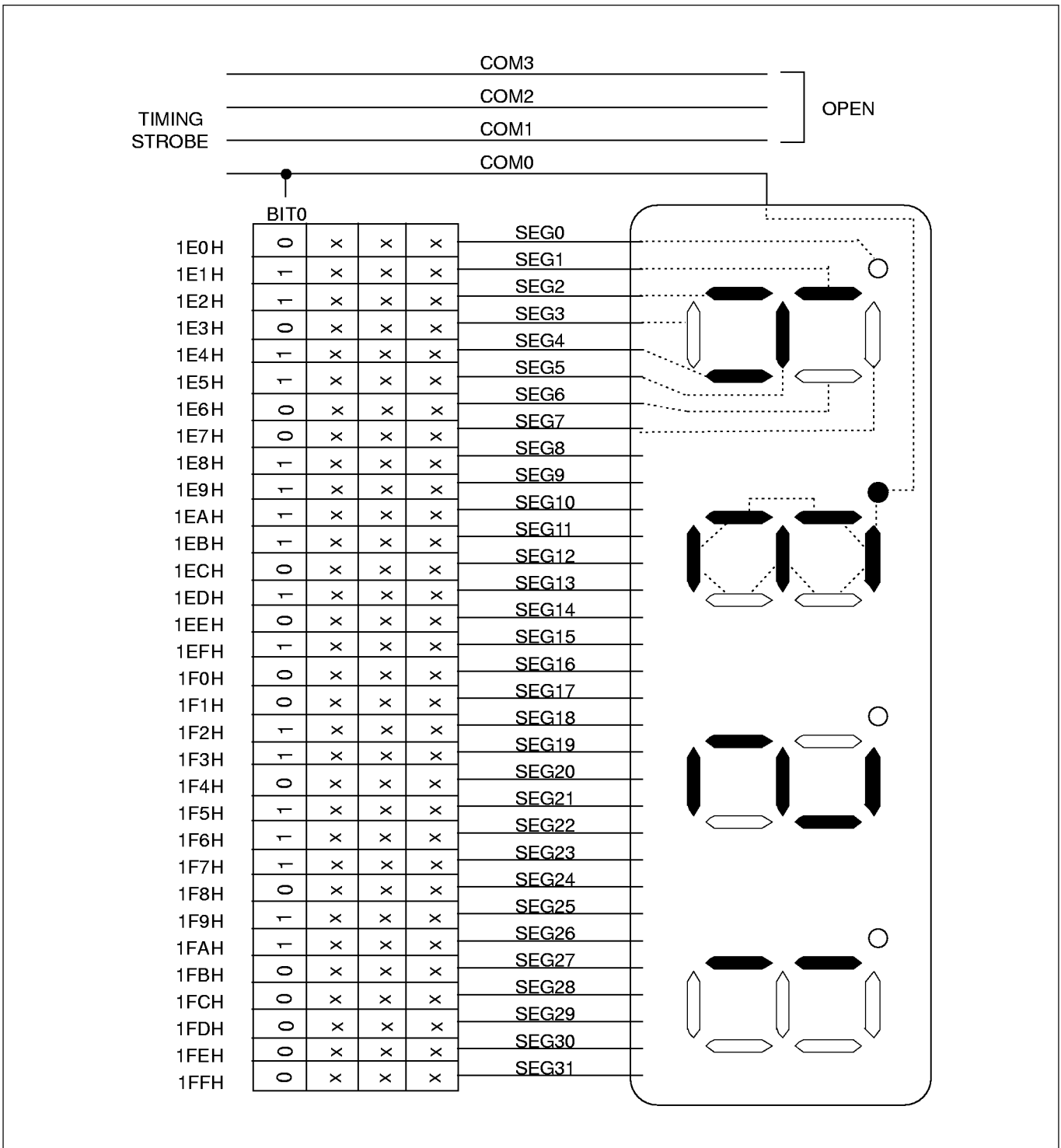


Figure 58. LCD Connection Example (Static)

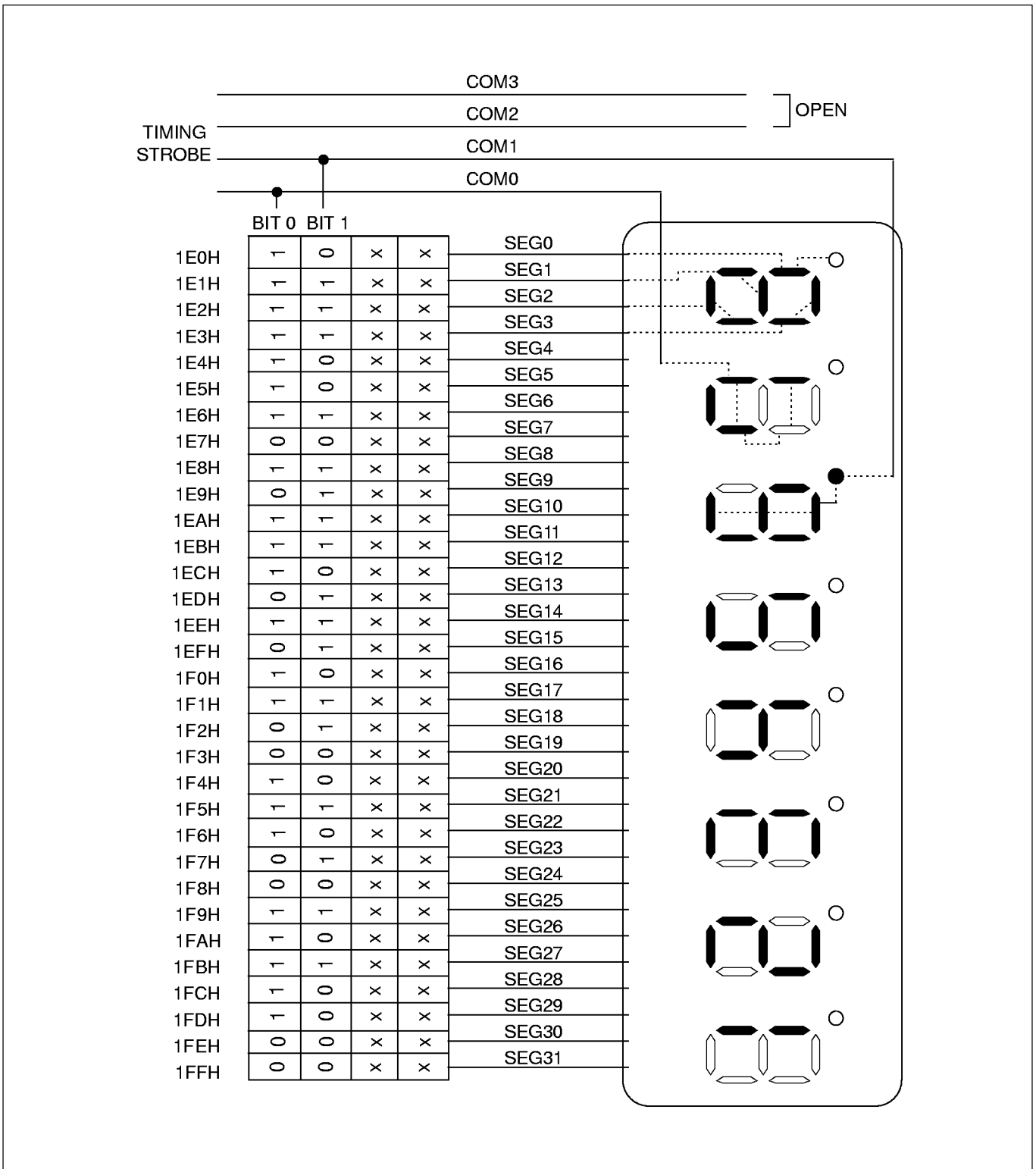


Figure 59. LCD Connection Example (1/2 Duty, 1/2 Bias)

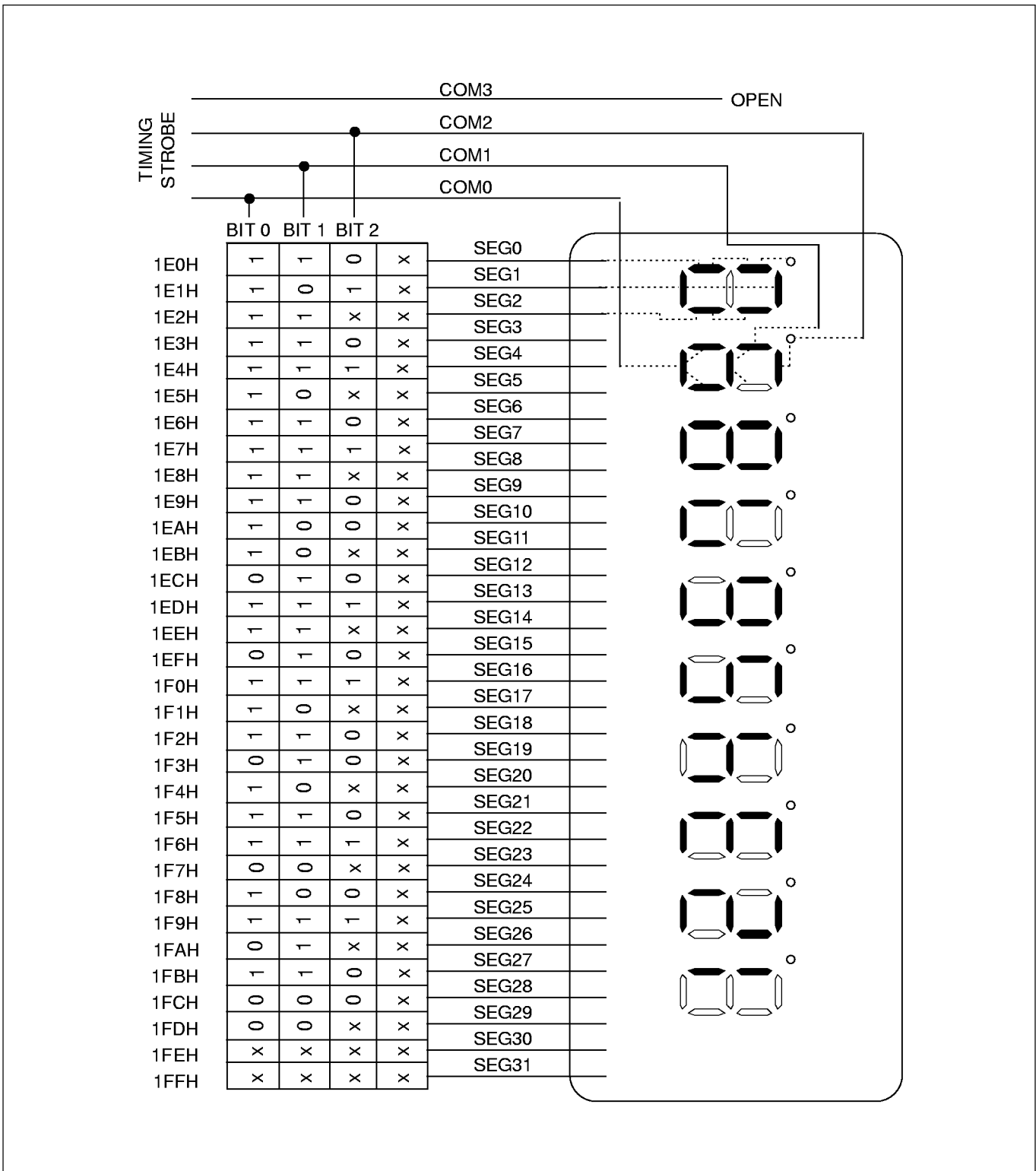


Figure 60. LCD Connection Example (1/3 Duty, 1/3 Bias)

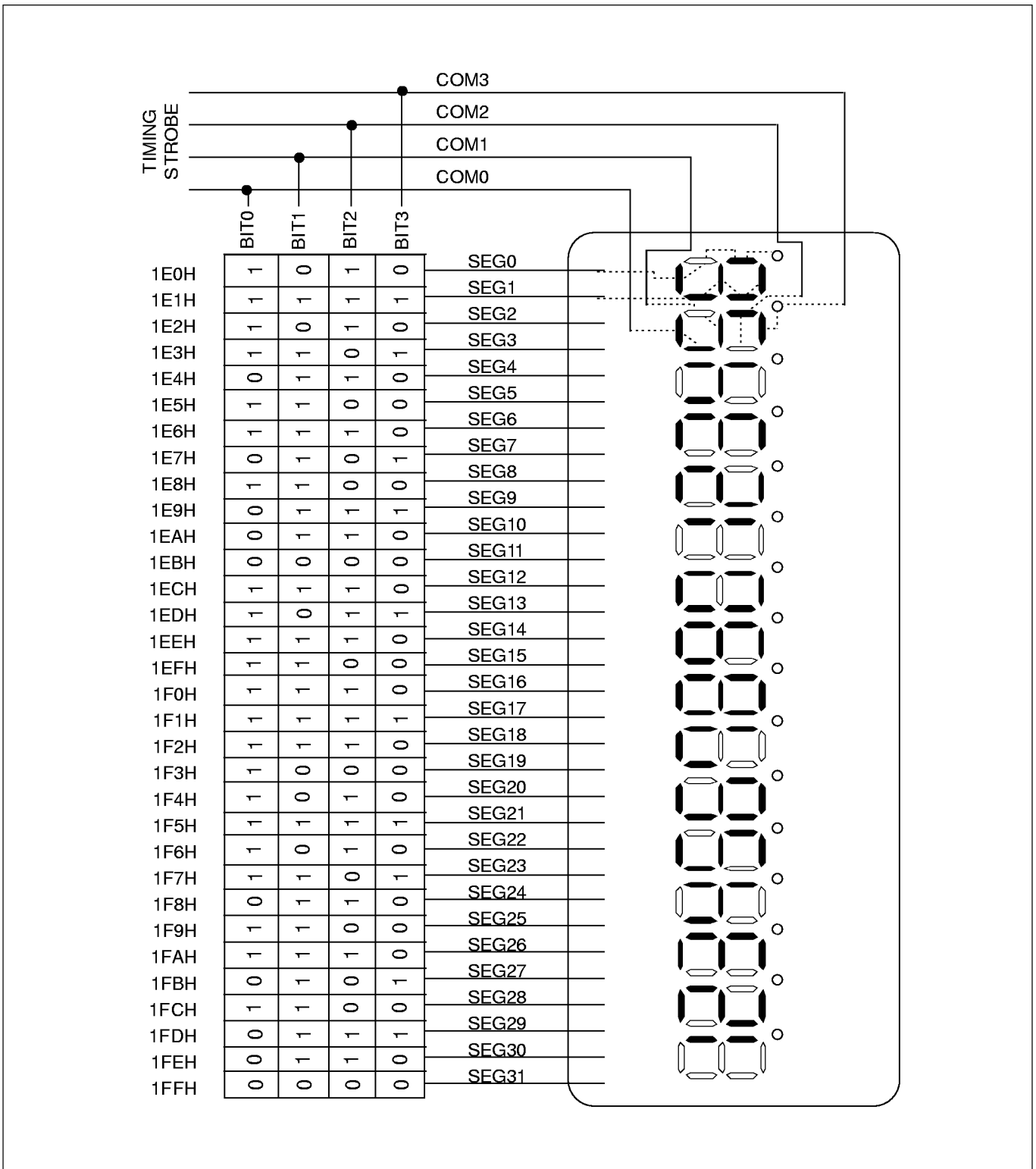


Figure 61. LCD Connection Example (1/4 Duty, 1/3 Bias)

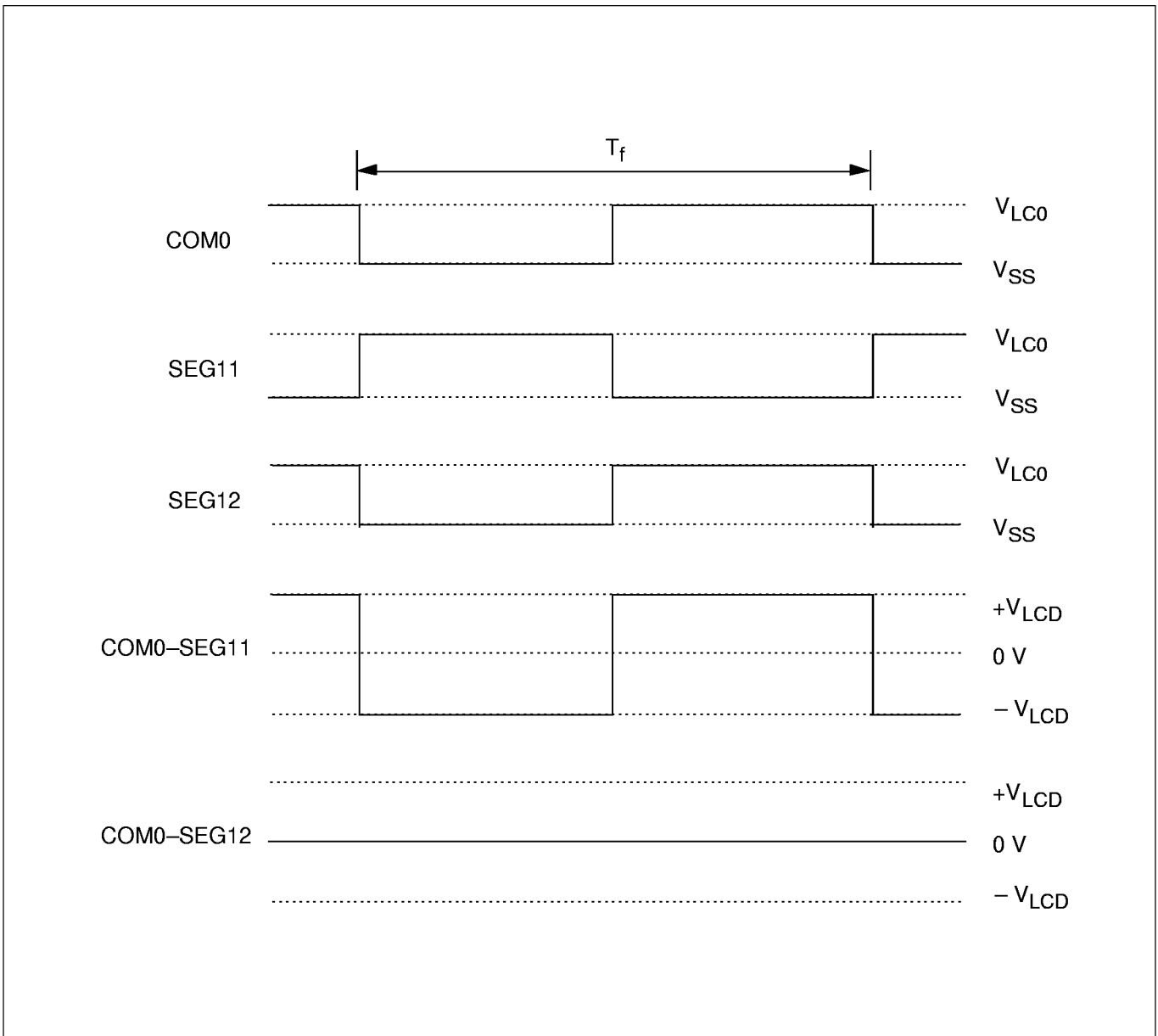


Figure 62. LCD Signal Waveforms in Static Mode

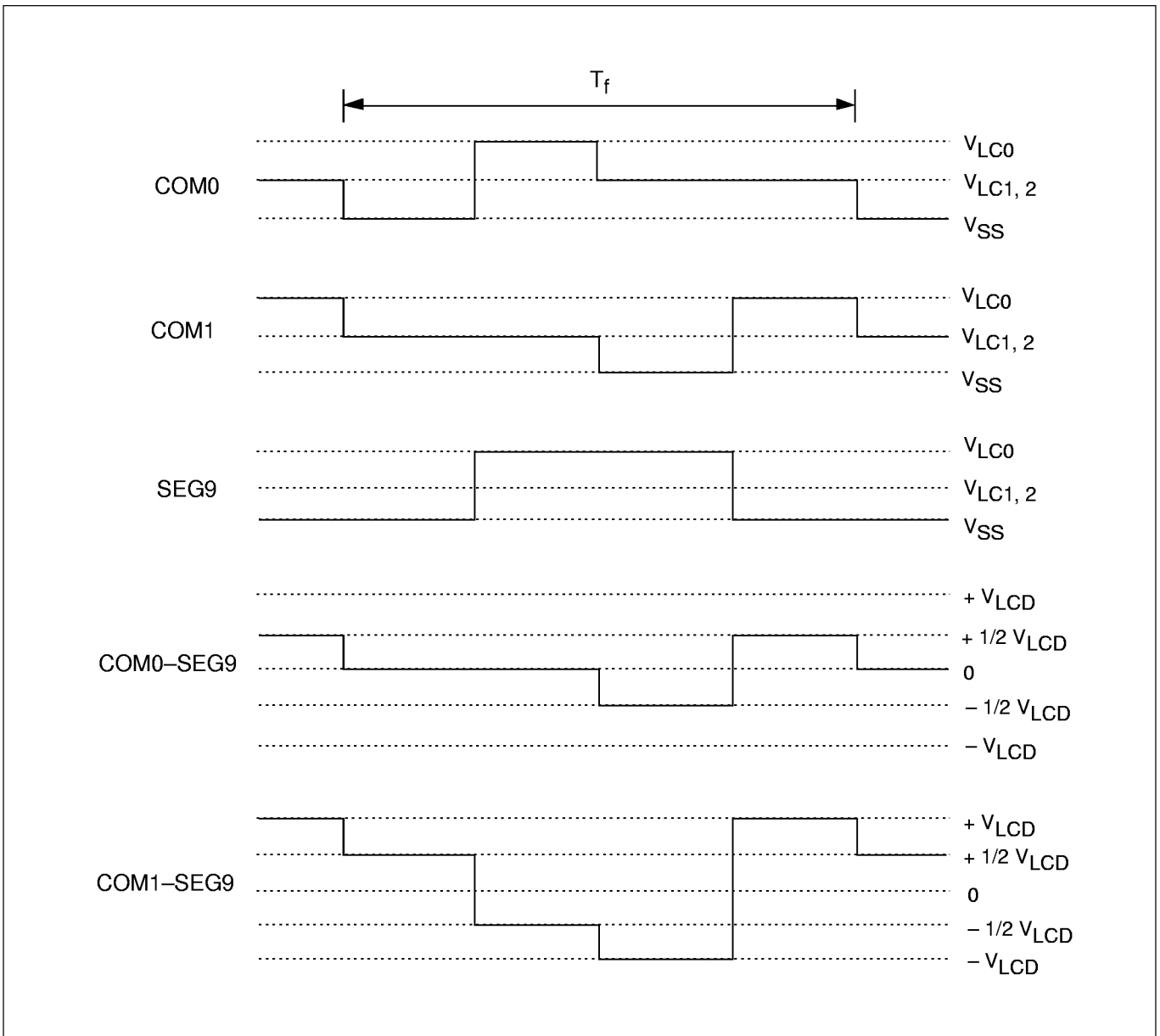


Figure 63. LCD Signal Waveforms at 1/2 Duty, 1/2 Bias

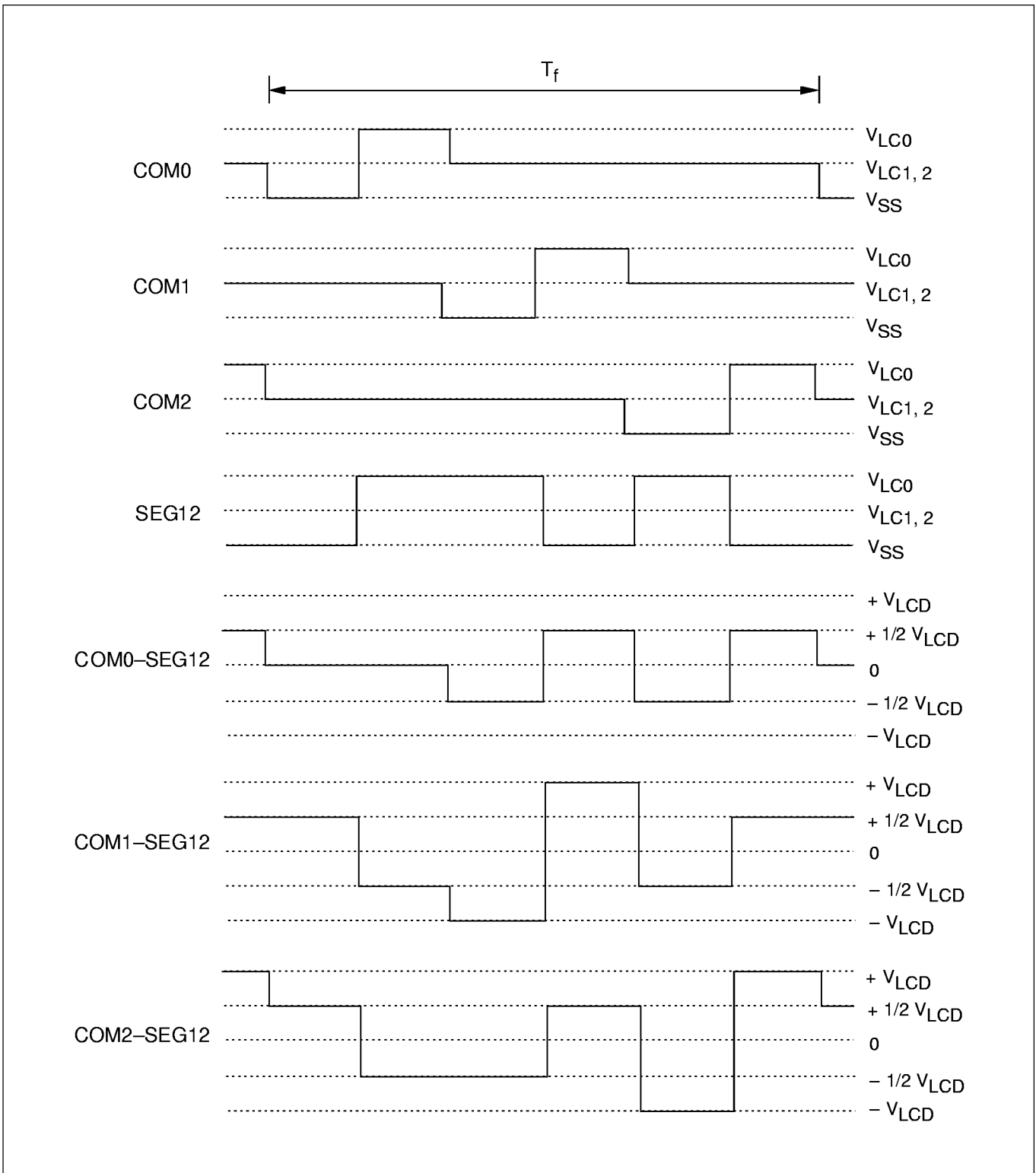


Figure 64. LCD Signal Waveforms at 1/3 Duty, 1/2 Bias



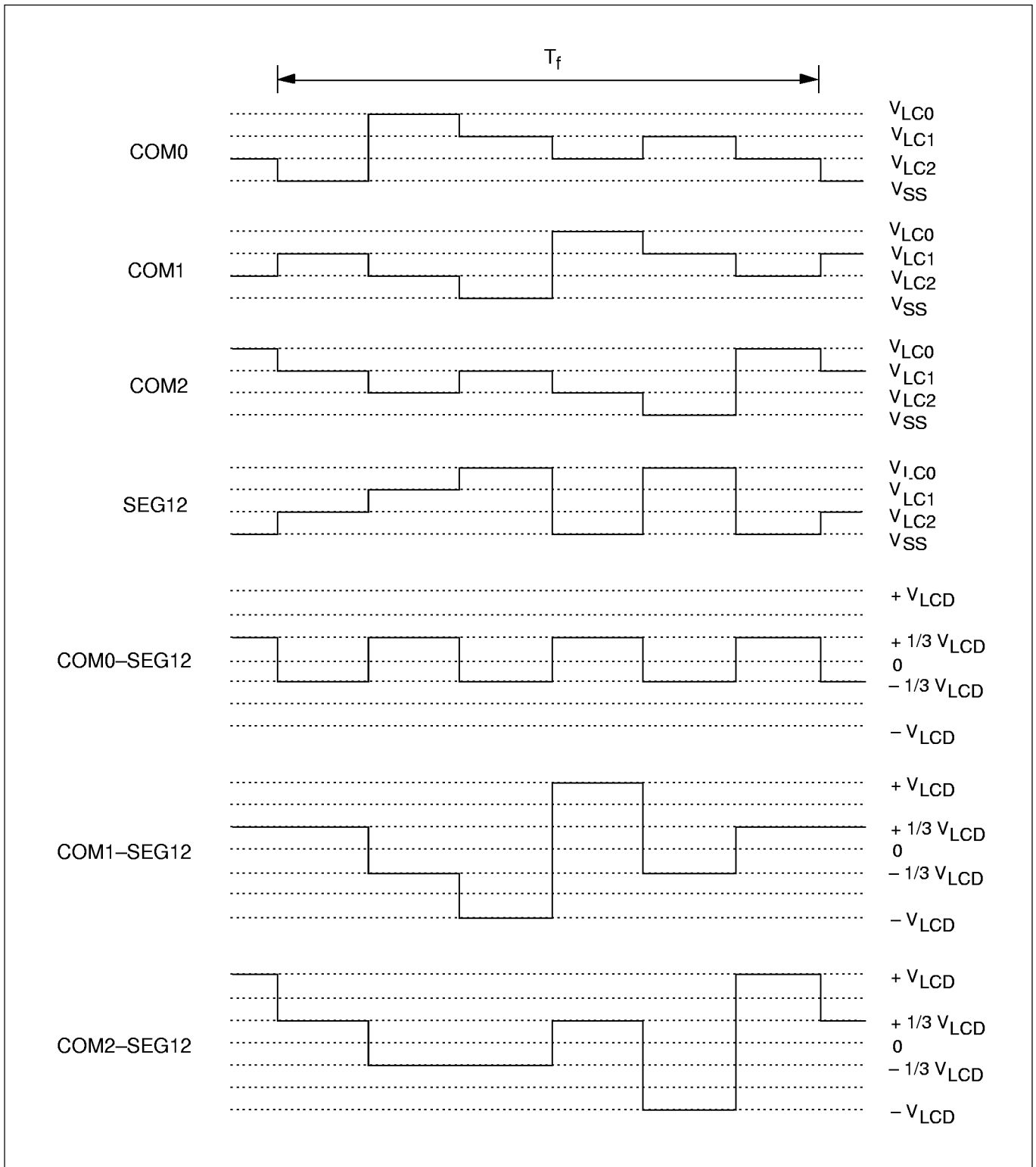


Figure 65. LCD Signal Waveforms at 1/3 Duty, 1/3 Bias

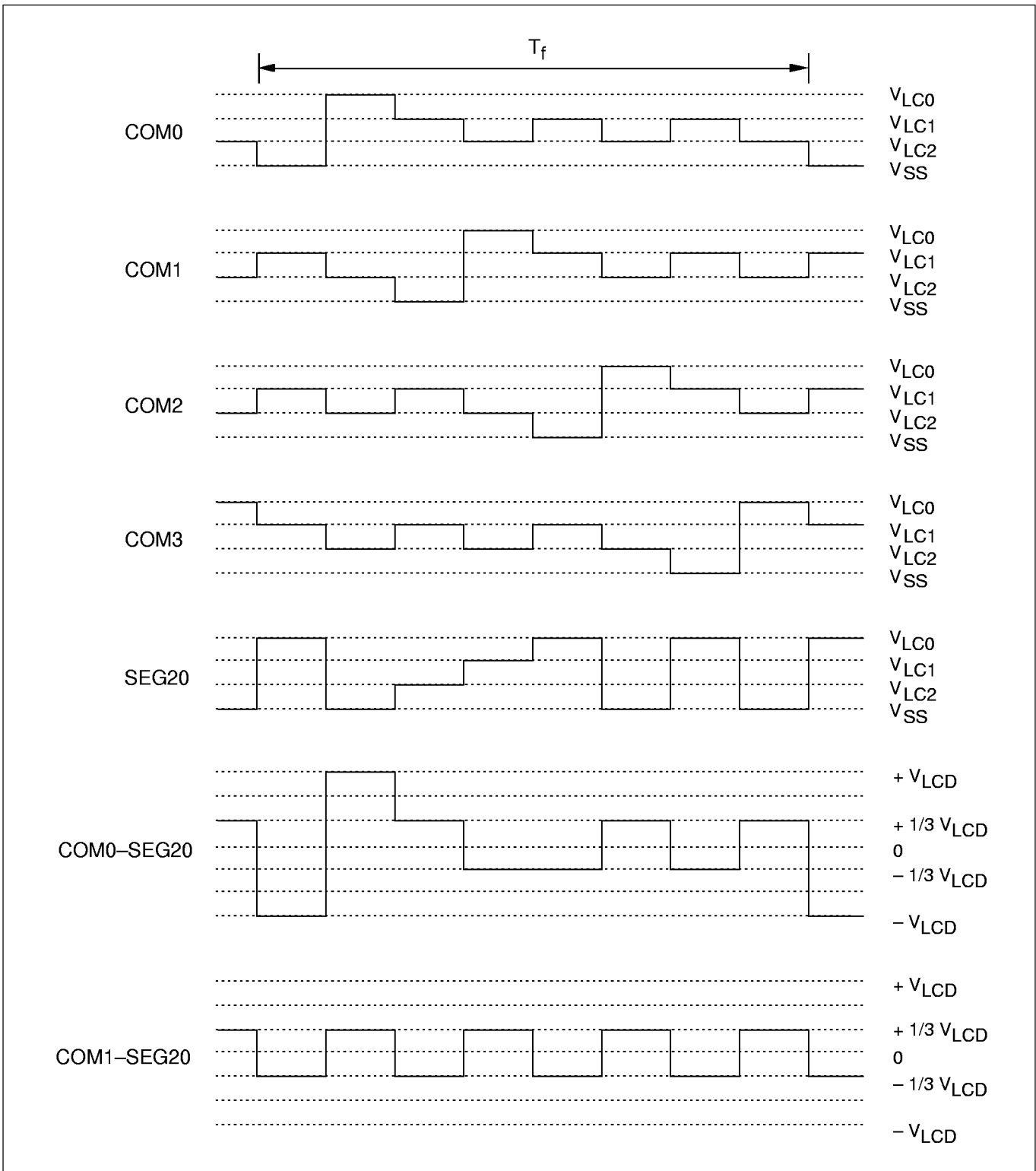


Figure 66. LCD Signal Waveforms at 1/4 Duty, 1/3 Bias



**SERIAL I/O MODE REGISTER (SMOD)**

The serial I/O mode register (SMOD) specifies the operation mode of the serial interface. SMOD register settings enable you to select either MSB-first or LSB-first serial transmission, and to operate in transmit-and-receive mode or receive-only mode.

When SMOD.3 is set to "1", the contents of the serial interface interrupt request flag, IRQS, and the 3-bit serial clock counter are cleared, and SIO operations are initiated. When the SIO transmission starts, SMOD.3 is cleared to "0".

**SERIAL I/O BUFFER REGISTER (SBUF)**

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 at the rate of one bit for each falling edge of the SIO clock. Receive data is simultaneously input from the SI pin to SBUF at the rate of one bit for each rising edge of the SIO clock.

When receive-only mode is used, incoming data is input to the SIO buffer at the rate of one bit for each rising edge of the SIO clock. SBUF can be read or written using 8-bit RAM control instructions.

**Table 37. SIO Mode Register (SMOD) Organization**

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

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

**NOTES:**

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

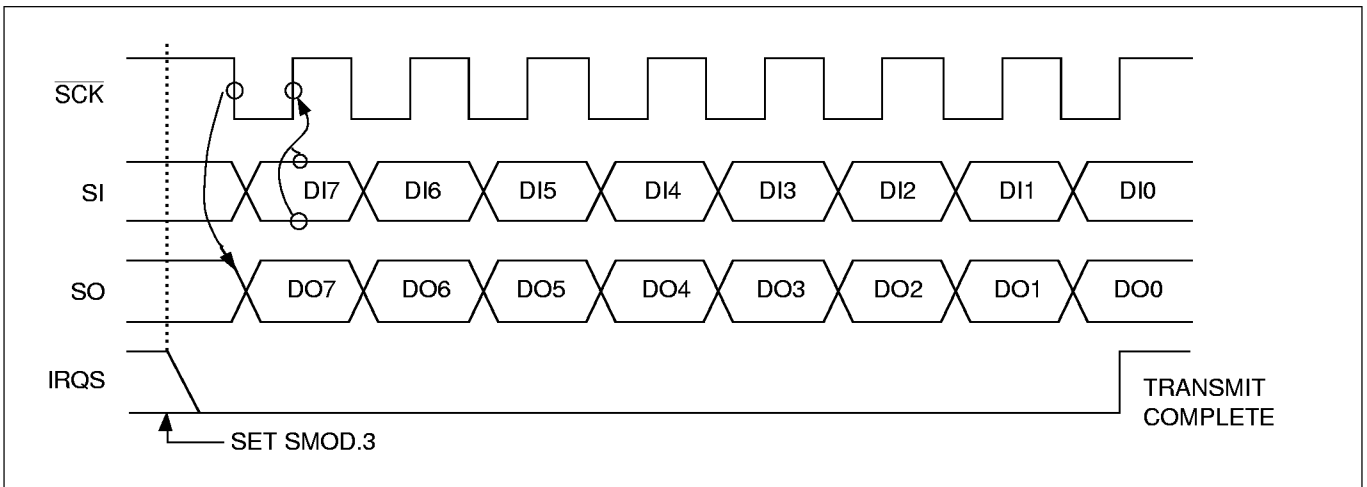


Figure 68. SIO Timing in Transmit/Receive Mode

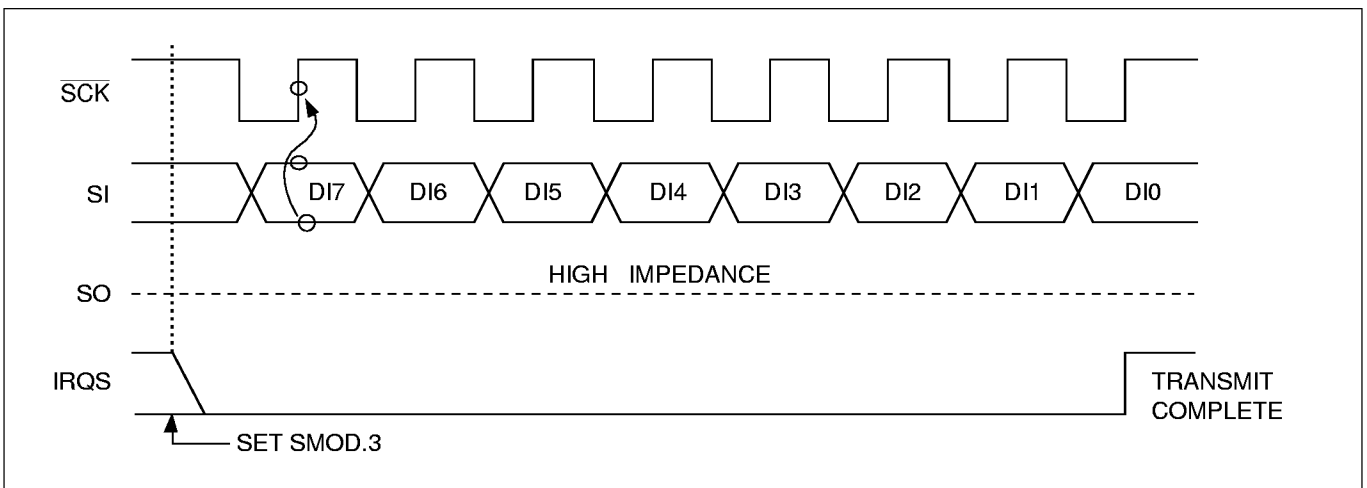


Figure 69. SIO Timing in Receive-Only Mode

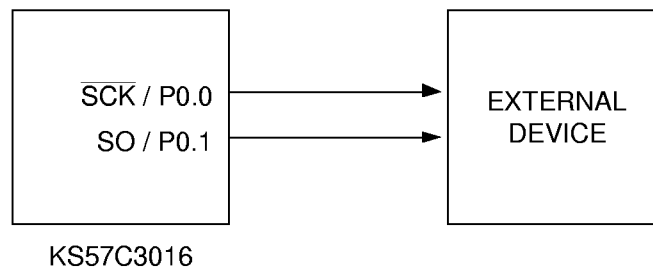
### PROGRAMMING TIP — Setting Transmit/Receive Modes for Serial I/O

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

```

BITS      EMB
SMB       15
LD        EA,#03H
LD        PMG1,EA      ; P0.0 /  $\overline{SCK}$  and P0.1 / 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        PMG1,EA      ; P0.0 /  $\overline{SCK}$  and P0.1 / SO ← Output, P0.2 / SI ← Input
LD        EA,TDATA     ; TDATA address = Bank0(20H–7FH)
LD        SBUF,EA
LD        EA,#4FH
LD        SMOD,EA     ; SIO start
BITR      IES          ; SIO Interrupt Enable
STEST     BTSTZ      IRQS
JR        STEST
LD        EA,SBUF
LD        RDATA,EA    ; RDATA address = Bank0 (20H–7FH)

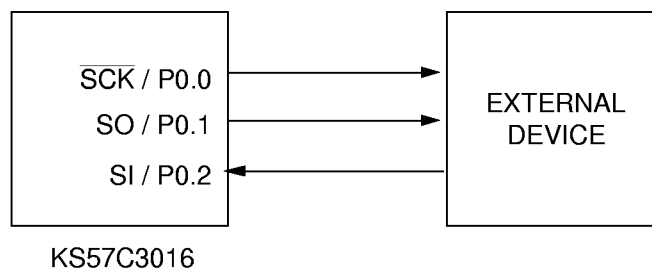
```

**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        PMG1,EA      ; P0.0 /  $\overline{\text{SCK}}$  and P0.1 / SO ← Output, P0.2 / SI ← Input
LD        EA,TDATA     ; TDATA address = Bank0 (20H–7FH)
LD        SBUF,EA
LD        EA,#8FH
LD        SMOD,EA     ; SIO start
EI
BITS      IES          ; SIO Interrupt Enable
.
.
.
INTS      PUSH        SB      ; Store SMB, SRB
          PUSH        EA      ; Store EA
          BITR        EMB
          LD          EA,TDATA ; EA ← Transmit data
          ; TDATA address = Bank0 (20H–7FH)
          ; Transmit data ↔ Receive data
          XCH         EA,SBUF  ; RDATA address = Bank0 (20H–7FH)
          LD          RDATA,EA ; SIO start
          BITS      SMOD.3
          POP         EA
          POP         SB
          IRET
    
```



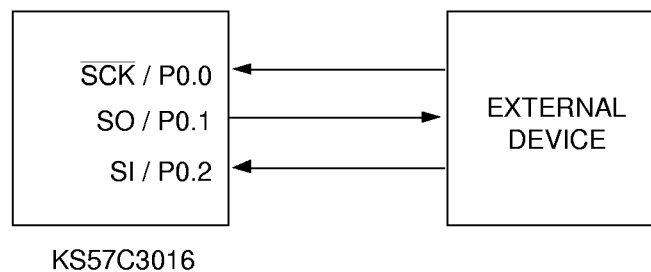
### PROGRAMMING TIP — Setting Transmit/Receive Modes for Serial I/O (Concluded)

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

```

BITR      EMB
LD        EA,#02H
LD        PMG1,EA          ; P0.1/ SO ← Output, P0.0/  $\overline{\text{SCK}}$  and P0.2/ SI← Input
LD        EA,TDATA        ; TDATA address = Bank0 (20H–7FH)
LD        SBUF,EA
LD        EA,#0FH
LD        SMOD,EA        ; SIO start
EI
BITS      IES            ; SIO Interrupt Enable
.
.
.
INTS      PUSH          SB          ; Store SMB, SRB
          PUSH          EA          ; Store EA
          BITR
          LD            EA,TDATA    ; EA ← Transmit data
          ; TDATA address = Bank0 (20H–7FH)
          XCH          EA,SBUF      ; Transmit data ↔ Receive data
          LD            RDATA,EA    ; RDATA address = Bank0 (20H–7FH)
          BITS          SMOD.3      ; SIO start
          POP          EA
          POP          SB
          IRET

```





**ELECTRICAL DATA**

**Table 38. Absolute Maximum Ratings**

(T<sub>A</sub> = 25 °C)

| Parameter             | Symbol           | Conditions   | Rating  | Units |
|-----------------------|------------------|--|---|-------|
| Supply Voltage        | V <sub>DD</sub>  | —  | − 0.3 to + 7.0  | V     |
| Input Voltage         | V <sub>I1</sub>  | Applies to I/O ports 4, 5, 7, and 8. (Pull-up resistors are individually assignable to pins at ports 4, 5, 7 and 8, or they can remain open-drain) | − 0.3 to V <sub>DD</sub> + 0.3<br>(With pull-up resistor)<br>− 0.3 to + 9.0<br>(Open-drain) | V     |
|                       | V <sub>I2</sub>  | All I/O ports except 4, 5, 7, and 8  | − 0.3 to V <sub>DD</sub> + 0.3  |       |
| Output Voltage        | V <sub>O</sub>   | —  | − 0.3 to V <sub>DD</sub> + 0.3  | V     |
| Output Current High   | I <sub>OH</sub>  | One I/O port active  | − 15  | mA    |
|                       |                  | All I/O ports active   | − 30  |       |
| Output Current Low    | I <sub>OL</sub>  | One I/O port active  | + 30 (Peak value)   | mA    |
|                       |                  |  | + 15 *  |       |
|                       |                  | Total for ports 0, 2–10  | + 100 (Peak value)  |       |
|                       |                  |  | + 60 *  |       |
| Operating Temperature | T <sub>A</sub>   | —  | − 40 to + 85  | °C    |
| Storage Temperature   | T <sub>stg</sub> | —  | − 65 to + 150   | °C    |

\* The values for Output Current Low ( I<sub>OL</sub> ) are calculated as Peak Value ×  $\sqrt{\text{Duty}}$  .

**Table 39. D.C. Electrical Characteristics**

(T<sub>A</sub> = − 40 °C to + 85 °C, V<sub>DD</sub> = 2.7 V to 6.0 V)

| Parameter          | Symbol           | Conditions   | Min                   | Typ | Max                | Units |
|--------------------|------------------|--|-----------------------|-----|--------------------|-------|
| Input High Voltage | V <sub>IH1</sub> | All input pins except those specified below for V <sub>IH2</sub> –V <sub>IH4</sub> | 0.7V <sub>DD</sub>    | —   | V <sub>DD</sub>    | V     |
|                    | V <sub>IH2</sub> | Ports 0, 1, 6, 7, and RESET  | 0.8V <sub>DD</sub>    |     | V <sub>DD</sub>    |       |
|                    | V <sub>IH3</sub> | Ports 4, 5, 7, and 8 with pull-up resistors assigned                               | 0.7V <sub>DD</sub>    |     | V <sub>DD</sub>    |       |
|                    |                  | Ports 4, 5, 7, and 8 are open-drain  | 0.7V <sub>DD</sub>    |     | 9                  |       |
|                    | V <sub>IH4</sub> | X <sub>in</sub> , X <sub>out</sub> , and XT <sub>in</sub>                          | V <sub>DD</sub> − 0.5 |     | V <sub>DD</sub>    |       |
| Input Low Voltage  | V <sub>IL1</sub> | All input pins except those specified below for V <sub>IL2</sub> –V <sub>IL3</sub> | —                     | —   | 0.3V <sub>DD</sub> | V     |
|                    | V <sub>IL2</sub> | Ports 0, 1, 6, 7, 9, 10, and RESET   |                       |     | 0.2V <sub>DD</sub> |       |
|                    | V <sub>IL3</sub> | X <sub>in</sub> , X <sub>out</sub> , and XT <sub>in</sub>                          |                       |     | 0.4                |       |

Table 39. D.C. Electrical Characteristics (Continued)

(T<sub>A</sub> = -40 °C to +85 °C, V<sub>DD</sub> = 2.7 V to 6.0 V)

| Parameter                   | Symbol            | Conditions   | Min                   | Typ | Max | Units |
|-----------------------------|-------------------|--|-----------------------|-----|-----|-------|
| Output High Voltage         | V <sub>OH1</sub>  | V <sub>DD</sub> = 4.5 V to 6.0 V<br>I <sub>OH</sub> = -1 mA<br>Ports 0, 2-10   | V <sub>DD</sub> - 1.0 | —   | —   | V     |
|                             |                   | I <sub>OH</sub> = -100 μA  | V <sub>DD</sub> - 0.5 |     |     |       |
|                             | V <sub>OH2</sub>  | V <sub>DD</sub> = 4.5 V to 6.0 V<br>I <sub>OH</sub> = -100 μA<br>Ports 11-13 only  | V <sub>DD</sub> - 2.0 |     |     |       |
|                             |                   | I <sub>OH</sub> = -30 μA   | V <sub>DD</sub> - 1.0 |     |     |       |
| Output Low Voltage          | V <sub>OL1</sub>  | V <sub>DD</sub> = 4.5 V to 6.0 V<br>I <sub>OL</sub> = 1.6 mA<br>Ports 4, 5, 7, and 8 only  | —                     | 0.8 | 2   | V     |
|                             |                   | I <sub>OL</sub> = 1.6 mA<br>Ports 0, 2, 3, 6, 9, 10, EO1, and EO2 only   |                       | —   | 0.4 |       |
|                             |                   | I <sub>OL</sub> = 400 μA<br>Ports 0, 2, 3, 6, 9, 10, EO1, and EO2 only   |                       |     | 0.2 |       |
|                             | V <sub>OL2</sub>  | V <sub>DD</sub> = 4.5 V to 6.0 V<br>I <sub>OL</sub> = 100 μA<br>Port 11, 12, and 13 only   |                       |     | 1   |       |
|                             |                   | I <sub>OL</sub> = 50 μA  |                       |     | 1   |       |
| Input High Leakage Current  | I <sub>LIH1</sub> | V <sub>I</sub> = V <sub>DD</sub><br>All input pins except RESET and those specified below for I <sub>LIH2</sub> -I <sub>LIH3</sub> | —                     | —   | 3   | μA    |
|                             | I <sub>LIH2</sub> | V <sub>I</sub> = V <sub>DD</sub><br>X <sub>in</sub> , X <sub>out</sub> , XT <sub>in</sub> only                                     |                       |     | 25  |       |
|                             | I <sub>LIH3</sub> | V <sub>I</sub> = 9 V<br>Ports 4, 5, 7, and 8 are open-drain  |                       |     | 20  |       |
| Input Low Leakage Current   | I <sub>LIL1</sub> | V <sub>I</sub> = 0 V<br>All input pins except X <sub>in</sub> , X <sub>out</sub> , XT <sub>in</sub> and RESET                      | —                     | —   | -3  | μA    |
|                             | I <sub>LIL2</sub> | V <sub>I</sub> = 0 V<br>X <sub>in</sub> , X <sub>out</sub> , and XT <sub>in</sub> only   |                       |     | -20 |       |
| Output High Leakage Current | I <sub>LOH1</sub> | V <sub>O</sub> = V <sub>DD</sub><br>All output pins except for ports 4, 5, 7, and 8  | —                     | —   | 3   | μA    |
|                             | I <sub>LOH2</sub> | Ports 4, 5, 7 and 8 are open-drain<br>V <sub>O</sub> = 9 V   |                       |     | 20  |       |

**Table 39. D.C. Electrical Characteristics (Continued)**

( $T_A = -40\text{ }^\circ\text{C}$  to  $+85\text{ }^\circ\text{C}$ ,  $V_{DD} = 2.7\text{ V}$  to  $6.0\text{ V}$ )

| Parameter                     | Symbol  | Conditions   | Min  | Typ | Max      | Units            |
|-------------------------------|---|--|------|-----|----------|------------------|
| Output Low Leakage Current    | $I_{LOL}$   | $V_O = 0\text{ V}$   | —    | —   | -3       | $\mu\text{A}$    |
| Pull-Up Resistor              | $R_{L1}$  | $V_I = 0\text{ V}$ ; $V_{DD} = 5\text{ V} \pm 10\%$<br>Port 0-3, 6, 9, and 10<br>(except P1.3)   | 15   | 46  | 80       | $\text{k}\Omega$ |
|                               |   | $V_{DD} = 3\text{ V} \pm 10\%$   | 30   | 90  | 200      |                  |
|                               | $R_{L2}$  | $V_O = V_{DD} - 2\text{ V}$<br>$V_{DD} = 5\text{ V} \pm 10\%$<br>Ports 4, 5, 7, and 8 only   | 15   | 40  | 70       |                  |
|                               |   | $V_{DD} = 3\text{ V} \pm 10\%$   | 10   | —   | 60       |                  |
|                               | $R_{L3}$  | $V_I = 0\text{ V}$ ; $V_{DD} = 5\text{ V} \pm 10\%$<br>RESET   | 100  | 230 | 400      |                  |
|                               |   | $V_{DD} = 3\text{ V} \pm 10\%$   | 200  | 490 | 800      |                  |
| LCD Drive Voltage             | $V_{LCD}$   | —  | 2.5  | —   | $V_{DD}$ | V                |
| LCD Voltage Dividing Resistor | $R_{LCD}$   | —  | 50   | 100 | 140      | $\text{k}\Omega$ |
| COM Output Impedance          | $R_{COM}$   | $V_{DD} = 5\text{ V} \pm 10\%$   | —    | 3   | 6        | $\text{k}\Omega$ |
|                               |   | $V_{DD} = 3\text{ V} \pm 10\%$   | —    | 10  | 15       |                  |
| SEG Output Impedance          | $R_{SEG}$   | $V_{DD} = 5\text{ V} \pm 10\%$   | —    | 3   | 20       | $\text{k}\Omega$ |
|                               |   | $V_{DD} = 3\text{ V} \pm 10\%$   | —    | 10  | 60       |                  |
| Supply Current (1)            | $I_{DD1}^{(2)}$   | $V_{DD} = 5\text{ V} \pm 10\%$ (3)<br>4.5 MHz crystal oscillator<br>$C1 = C2 = 22\text{ pF}$<br>CE high; PLL operates                              | —    | 12  | 25       | mA               |
|                               |   | Idle mode; $V_{DD} = 5\text{ V} \pm 10\%$<br>4.5 MHz crystal oscillator<br>$C1 = C2 = 22\text{ pF}$<br>CPU clock = $f_{xx}/4$<br>CE low; PLL stops | —    | 1.4 | 1.8      |                  |
|                               | $V_{DD} = 3\text{ V} \pm 10\%$<br>CPU clock = $f_{xx}/64$ | —  | 0.23 | 1.0 |          |                  |
|                               | $I_{DD3}^{(5)}$   | $V_{DD} = 3\text{ V} \pm 10\%$<br>32 kHz crystal oscillator<br>CE low; PLL stops   | —    | 25  | 120      |                  |
|                               | $I_{DD4}^{(5)}$   | Idle mode; $V_{DD} = 3\text{ V} \pm 10\%$<br>32 kHz crystal oscillator   | —    | 20  | 30       | $\mu\text{A}$    |

**Table 39. D.C. Electrical Characteristics (Concluded)**(T<sub>A</sub> = -40 °C to +85 °C, V<sub>DD</sub> = 2.7 V to 6.0 V)

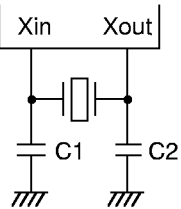
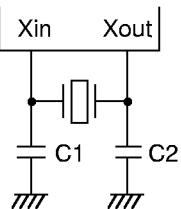
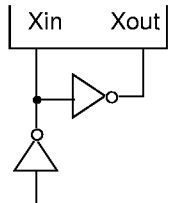
| Parameter              | Symbol               | Conditions  | Min | Typ  | Max | Units |
|------------------------|----------------------|---|-----|------|-----|-------|
| Supply Current (cont.) | I <sub>DD5</sub>     | Stop 1 mode; XT <sub>in</sub> = 0 V<br>V <sub>DD</sub> = 5 V ± 10%<br>CPU clock = f <sub>xx</sub> /4<br>CE low; PLL stops | —   | 0.6  | 5   | μA    |
|                        |                      | V <sub>DD</sub> = 3 V ± 10%<br>CPU clock = f <sub>xx</sub> /64  |     | 0.2  | 3   |       |
|                        | I <sub>DD6</sub>     | V <sub>DD</sub> = 5 V ± 10%<br>4.5 MHz crystal oscillator<br>CPU clock = f <sub>xx</sub> /4<br>CE low; PLL stops          |     | 4.2  | 8   | mA    |
|                        |                      | V <sub>DD</sub> = 3 V ± 10%<br>CPU clock = f <sub>xx</sub> /64  |     | 0.7  | 1.2 |       |
|                        | I <sub>DD7</sub> (2) | Stop 2 mode; XT <sub>in</sub> = 0 V<br>V <sub>DD</sub> = 5 V ± 10%<br>CPU clock = f <sub>xx</sub> /4<br>CE low; PLL stops |     | 0.12 | 2.0 | μA    |

**NOTES:**

1. 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.
2. I<sub>DD1</sub> and I<sub>DD7</sub> are guaranteed in T<sub>A</sub> = -20 °C to +85 °C
3. Data includes power consumption for subsystem clock oscillation.
4. For high-speed controller operation, the power control register (PCON) must be set to 0011B.
5. For low-speed controller operation, the power control register (PCON) must be set to 0000B.
6. When the system clock control register, SCMOD, is set to 1001B, main system clock oscillation stops and the subsystem clock is used.

**Table 40. Main System Oscillator Characteristics**

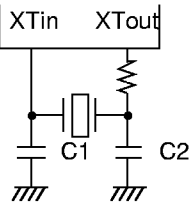
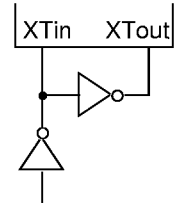
( $T_A = -40\text{ }^\circ\text{C} + 85\text{ }^\circ\text{C}$ ,  $V_{DD} = 2.7\text{ V to } 6.0\text{ V}$ )

| Oscillator         | Clock Configuration   | Parameter   | Test Condition   | Min | Typ | Max | Units |
|--------------------|---|---|--|-----|-----|-----|-------|
| Ceramic Oscillator |    | Oscillation frequency <sup>(1)</sup>                            | —  | 0.4 | —   | 5.0 | MHz   |
|                    |   | Stabilization time <sup>(2)</sup>                               | Stabilization occurs when $V_{DD}$ is equal to the minimum oscillator voltage range. | —   | —   | 4   | ms    |
| Crystal Oscillator |   | Oscillation frequency <sup>(1)</sup>                            | —  | 0.4 | 4.5 | 5.0 | MHz   |
|                    |   | Stabilization time <sup>(2)</sup>                               | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$   | —   | —   | 10  | ms    |
|                    |   |   | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$   | —   | —   | 30  |       |
| External Clock     |  | $X_{in}$ input frequency <sup>(1)</sup>                         | —  | 0.4 | —   | 4.5 | MHz   |
|                    |   | $X_{in}$ input high and low level width ( $t_{XH}$ , $t_{XL}$ ) | —  | —   | 111 | —   | 1250  |

**NOTES:**

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

**Table 41. Subsystem Clock Oscillator Characteristics** $(T_A = -40\text{ }^\circ\text{C} + 85\text{ }^\circ\text{C}, V_{DD} = 2.7\text{ V to } 6.0\text{ V})$ 

| Oscillator         | Clock Configuration   | Parameter  | Test Condition                           | Min | Typ    | Max | Units         |
|--------------------|---|--|--|-----|--------|-----|---------------|
| Crystal Oscillator |  | Oscillation frequency (1)  | —  | 32  | 32.768 | 35  | kHz           |
|                    |   | Stabilization time (2)   | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$ | —   | 1.0    | 2   | s             |
|                    |   |  | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$ | —   | —      | 10  |               |
| External Clock     |  | $XT_{in}$ input frequency (1)                                    | —  | 32  | —      | 100 | kHz           |
|                    |   | $XT_{in}$ input high and low level width ( $t_{XH}$ , $t_{XL}$ ) | —  | 5   | —      | 15  | $\mu\text{s}$ |

**NOTES:**

- Oscillation frequency and  $XT_{in}$  input frequency data are for oscillator characteristics only.
- Stabilization time is the interval required for oscillator stabilization after a power-on occurs.

**Table 42. Input/Output Capacitance**

( $T_A = 25\text{ }^\circ\text{C}$ ,  $V_{DD} = 0\text{ V}$ )

| 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 43. A/D Converter Electrical Characteristics**

( $T_A = -10\text{ }^\circ\text{C}$  to  $+70\text{ }^\circ\text{C}$ ,  $V_{DD} = 3.5\text{ V}$  to  $6.0\text{ V}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ )

| Parameter              | Symbol    | Condition                          | Min       | Typ  | Max           | Units         |
|------------------------|-----------|------------------------------------|-----------|------|---------------|---------------|
| Resolution             |           | —                                  | 8         | 8    | 8             | bit           |
| Absolute accuracy (1)  |           | $2.5\text{ V} < AV_{REF} < V_{DD}$ | —         | —    | $\pm 1.5$     | LSB           |
| Conversion time (2)    | $t_{CON}$ | —                                  | —         | —    | $100/f_x$ (3) | $\mu\text{s}$ |
| Analog input voltage   | $V_{IAN}$ | —                                  | $AV_{SS}$ | —    | $AV_{REF}$    | V             |
| Analog input impedance | $R_{AN}$  | —                                  | —         | 1000 | —             | $M\Omega$     |

**NOTES:**

1. Absolute accuracy does not include the quantization error ( $\pm 1/2$  LSB).
2. Conversion time is the time required from the moment a conversion operation starts until it ends (EOC = 0).
3. 'fx' is the abbreviation for main system clock.

**Table 43. A.C. Electrical Characteristics** $(T_A = -40\text{ }^\circ\text{C to } +85\text{ }^\circ\text{C, } V_{DD} = 2.7\text{ V to } 6.0\text{ V})$ 

| Parameter                        | Symbol                                       | Conditions  | Min               | Typ | Max  | Units         |
|----------------------------------|--|---|-------------------|-----|------|---------------|
| Instruction Cycle Time (Note 1)  | $t_{CY}$                                     | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$                        | 0.89              | —   | 64   | $\mu\text{s}$ |
|                                  |  | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$                        | 3.8               |     | 64   |               |
|                                  |  | With subsystem clock (fxt)                                      | 114               | 122 | 125  |               |
| TCL0, TCL1 Input Frequency       | $f_{TI0}, f_{TI1}$                           | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$                        | 0                 | —   | 1    | MHz           |
|                                  |  | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$                        |                   |     | 275  | kHz           |
| TCL0, TCL1 Input High, Low Width | $t_{TIH0}, t_{TIL0}$<br>$t_{TIH1}, t_{TIL1}$ | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$                        | 0.48              | —   | —    | $\mu\text{s}$ |
|                                  |  | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$                        | 1.8               |     |      |               |
| SCK Cycle Time                   | $t_{KCY}$                                    | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$<br>External SCK source | 800               | —   | —    | ns            |
|                                  |  | Internal SCK source   | 950               |     |      |               |
|                                  |  | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$<br>External SCK source | 3200              |     |      |               |
|                                  |  | Internal SCK source   | 3800              |     |      |               |
| SCK High, Low Width              | $t_{KH}, t_{KL}$                             | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$<br>External SCK source | 400               | —   | —    | ns            |
|                                  |  | Internal SCK source   | $t_{KCY}/2 - 50$  |     |      |               |
|                                  |  | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$<br>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       | $t_{KSO}$                                    | $V_{DD} = 4.5\text{ V to } 6.0\text{ V}$<br>External SCK source | —                 | —   | 300  | ns            |
|                                  |  | Internal SCK source   |                   |     | 250  |               |
|                                  |  | $V_{DD} = 2.7\text{ V to } 4.5\text{ V}$<br>External SCK source |                   |     | 1000 |               |
|                                  |  | Internal SCK source   |                   |     | 1000 |               |



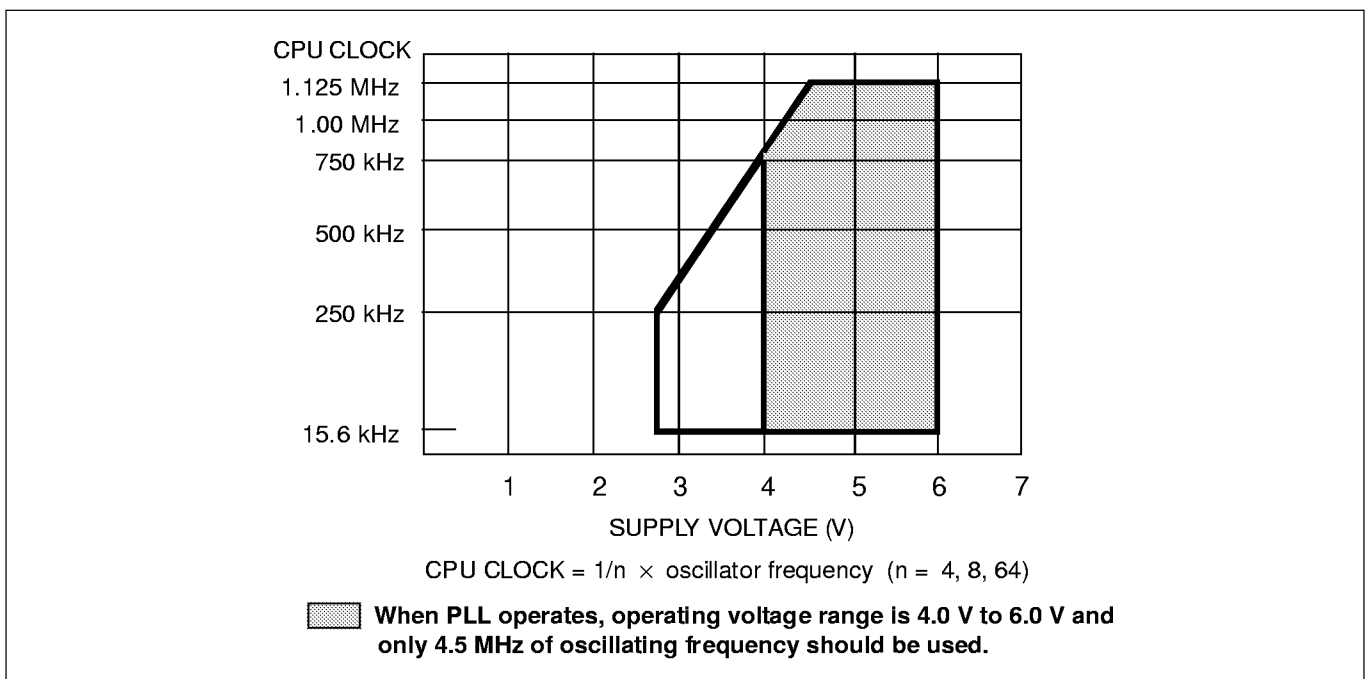
**Table 43. A.C. Electrical Characteristics (Continued)**

( $T_A = -40\text{ }^\circ\text{C}$  to  $+85\text{ }^\circ\text{C}$ ,  $V_{DD} = 2.7\text{ V}$  to  $6.0\text{ V}$ )

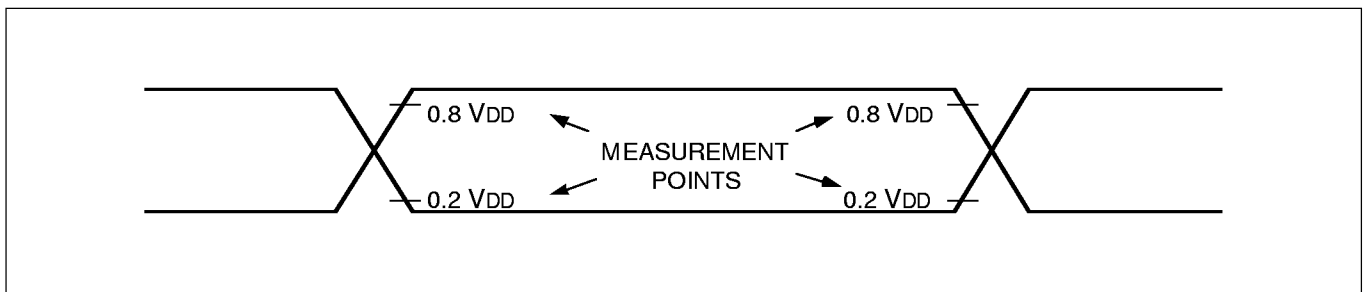
| Parameter                       | Symbol                  | Conditions                | Min      | Typ | Max | Units         |
|---------------------------------|-------------------------|---------------------------|----------|-----|-----|---------------|
| Interrupt Input High, Low Width | $t_{INTH}$ , $t_{INTL}$ | INT0                      | (Note 2) | —   | —   | $\mu\text{s}$ |
|                                 |                         | INT1, INT2, INT4, KS0–KS3 | 10       | —   | —   | $\mu\text{s}$ |
| RESET Input Low Width           | $t_{RSL}$               | Input                     | 10       | —   | —   | $\mu\text{s}$ |

**NOTES:**

1. Unless otherwise specified, Instruction Cycle Time condition values assume a main system clock (fx) source.
2. Minimum value for INT0 is based on a clock of  $2t_{CY}$  or  $128 / f_x$  as assigned by the IMOD0 register setting.



**Figure 70. Standard Operating Voltage Range**



**Figure 71. A.C. Timing Measurement Points (Except for  $X_{In}$  and  $XT_{In}$ )**

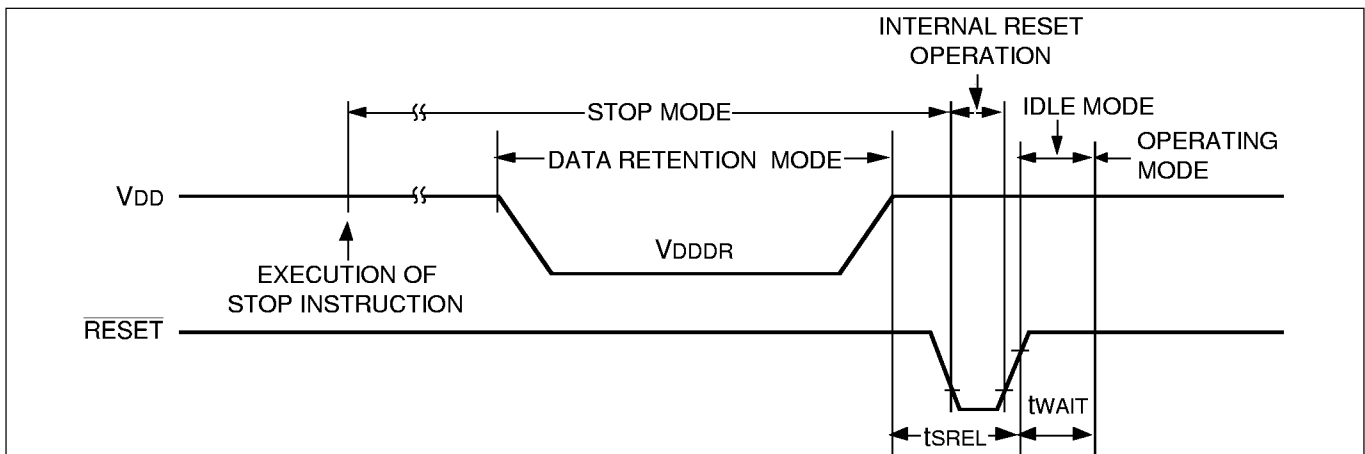
**Table 44. RAM Data Retention Supply Voltage in Stop Mode**

( $T_A = -40\text{ }^\circ\text{C}$  to  $+85\text{ }^\circ\text{C}$ )

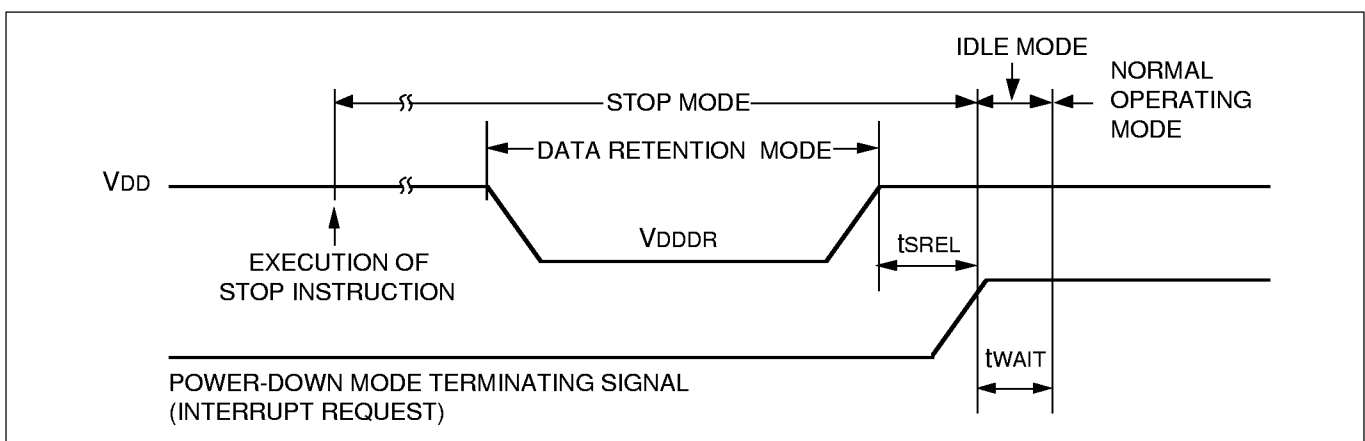
| Parameter                              | Symbol     | Conditions                            | Min | Typ            | Max | Unit          |
|--|------------|---------------------------------------|-----|----------------|-----|---------------|
| Data retention supply voltage          | $V_{DDDR}$ | —                                     | 2.0 | —              | 6.0 | V             |
| Data retention supply current          | $I_{DDDR}$ | $V_{DDDR} = 2.0\text{ V}$             | —   | 0.1            | 10  | $\mu\text{A}$ |
| Release signal set time                | $t_{SREL}$ | —                                     | 0   | —              | —   | $\mu\text{s}$ |
| Oscillator stabilization wait time (1) | $t_{WAIT}$ | Released by $\overline{\text{RESET}}$ | —   | $2^{17} / f_x$ | —   | 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.



**Figure 72. Stop Mode Release Timing When Initiated By  $\overline{\text{RESET}}$**



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

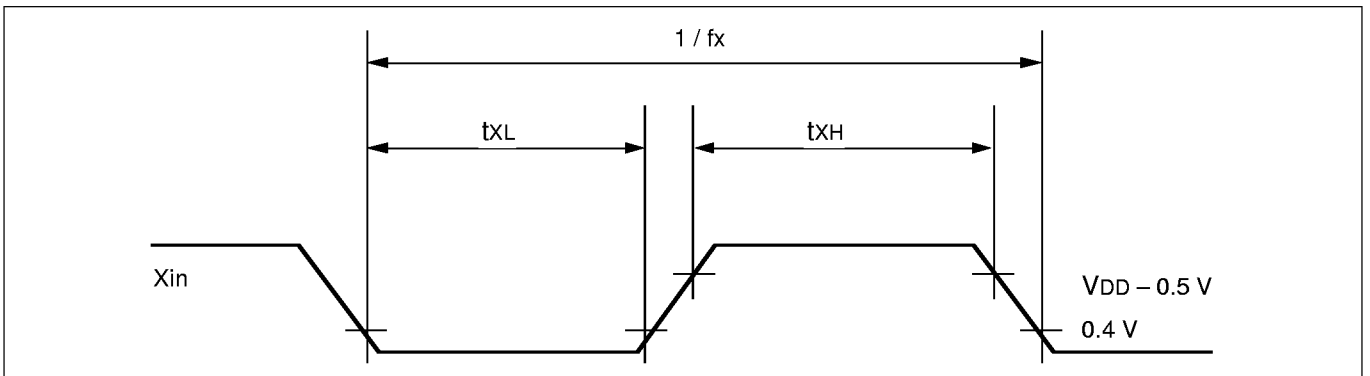


Figure 74. Clock Timing Measurement at  $X_{in}$

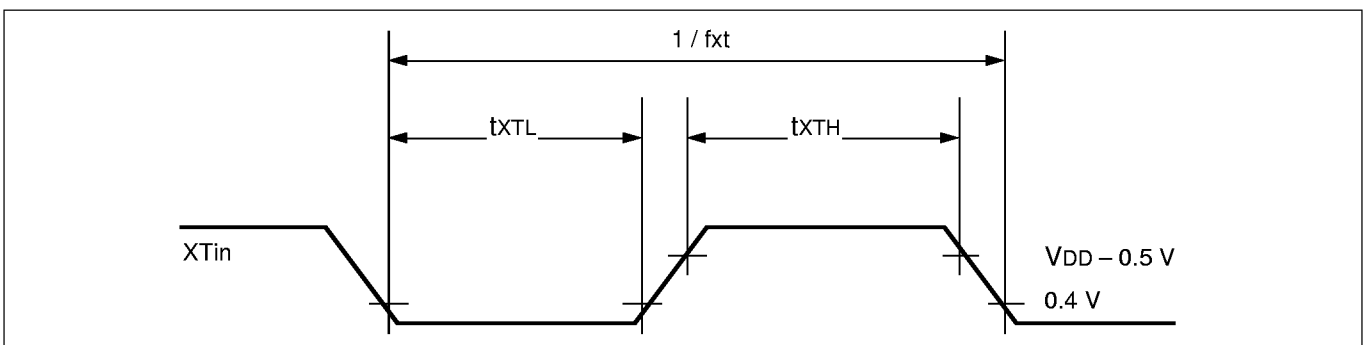


Figure 75. Clock Timing Measurement at  $XT_{in}$

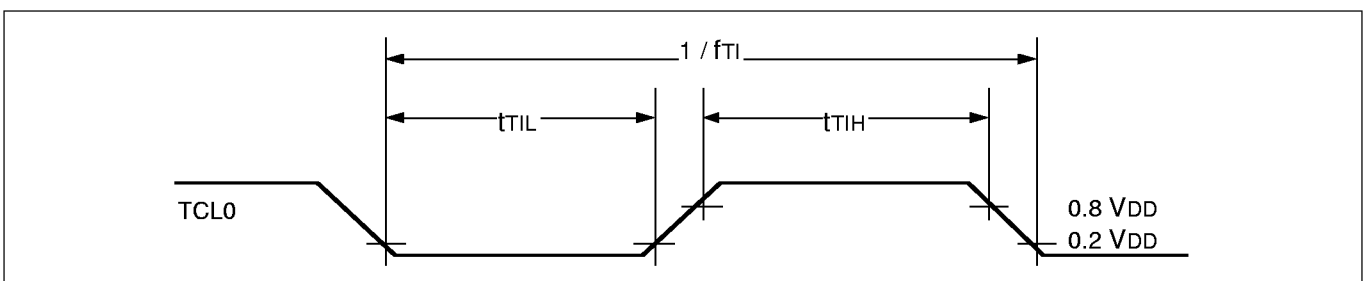


Figure 76.  $TCl_0$  Timing

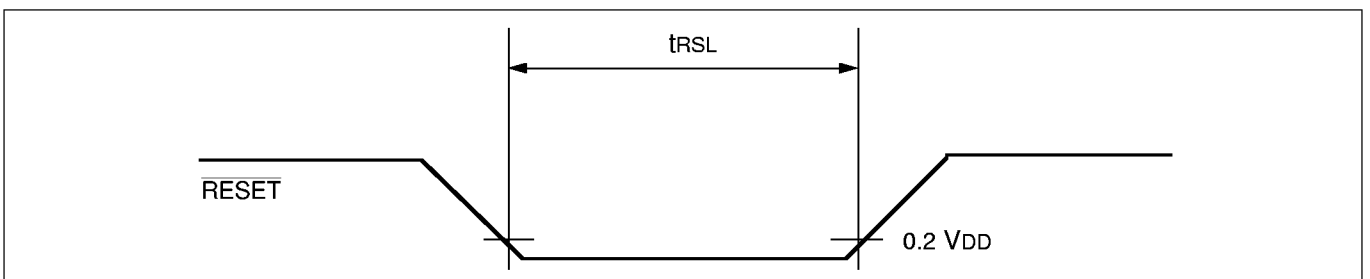


Figure 77. Input Timing for  $RESET$  Signal

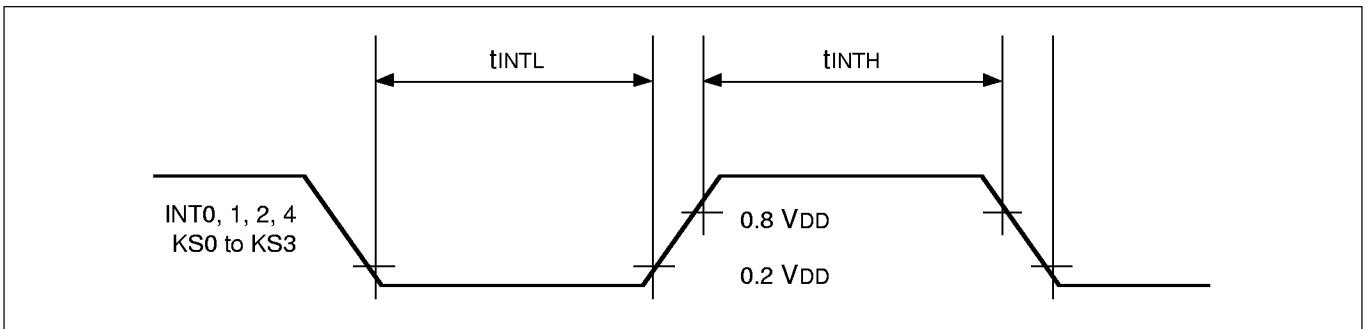


Figure 78. Input Timing for External Interrupts and Quasi-Interrupts

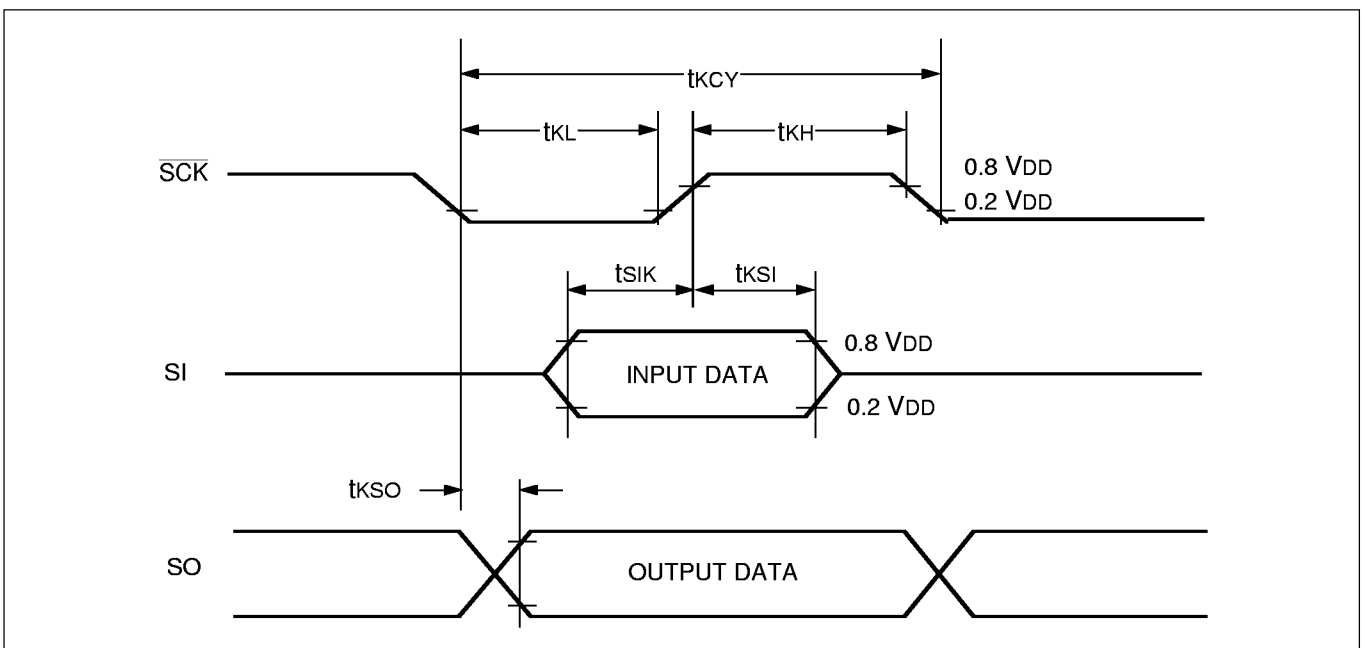
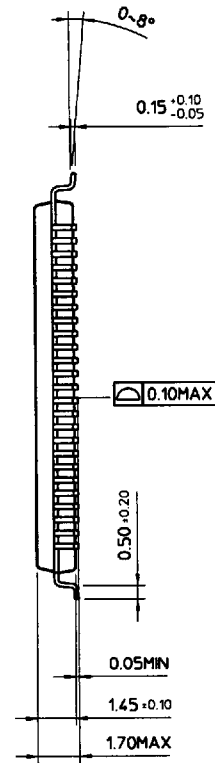
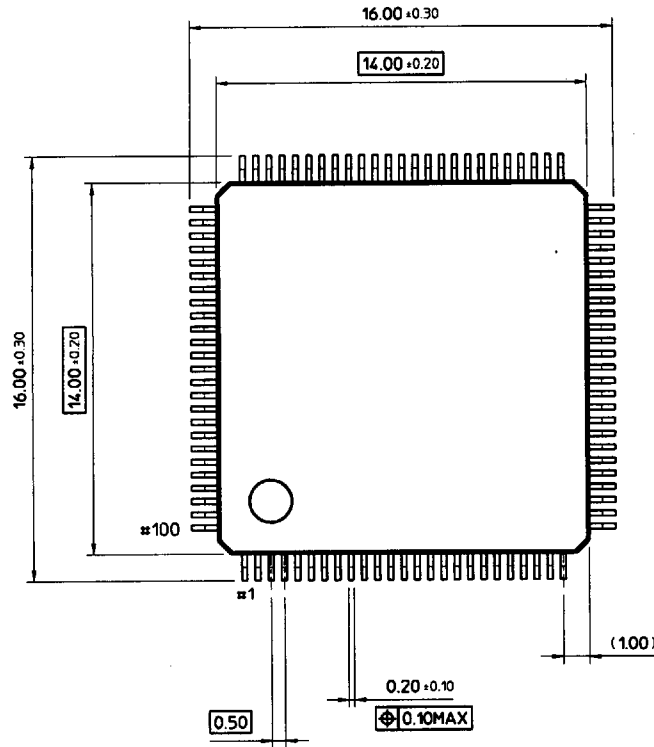


Figure 79. Serial Data Transfer Timing

100-QFP-1414



100-QFP-1420B

