LSI Computer Systems, Inc. 1235 Walt Whitman Road, Melville, NY 11747 (631) 271-0400 FAX (631) 271-0405

# LS7266R1

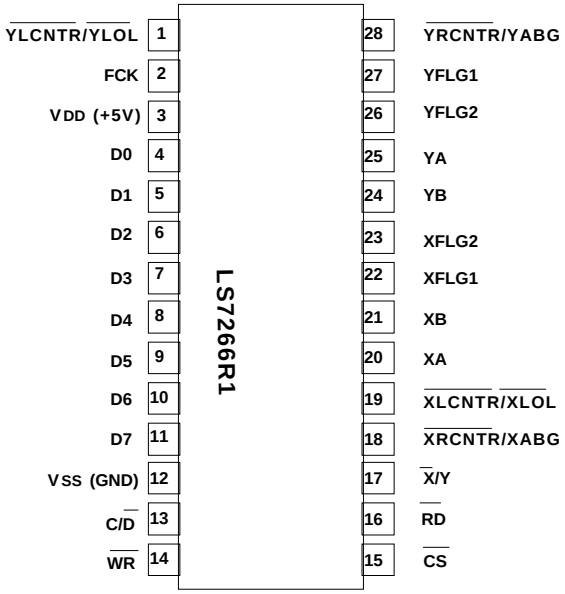
## 24-BIT DUAL-AXIS QUADRATURE COUNTER

*December 2002*

### FEATURES:

- 30 MHz count frequency in non-quadrature mode, 17MHz in x4 quadrature mode.
- Dual 24-bit counters to support X and Y axes in motion control applications.
- Dual 24-bit comparators.
- Digital filtering of the input quadrature clocks
- Programmable 8-bit separate filter clock prescalers for each axis.
- Error flags for noise exceeding filter band width.
- Programmable Index Input and other programmable I/Os.
- Independent mode programmability for each axis.
- Programmable count modes: Quadrature (x1, x2, x4) / Non-quadrature, Normal / Modulo-N / Range Limit / Non-Recycle, Binary / BCD.
- 8-bit 3-State data I/O bus.
- 5V operation ($V_{DD}$-$V_{SS}$).
- TTL/CMOS compatible I/Os.
- LS7266R1 (DIP); LS7266R1-SD (Skinny DIP); LS7266R1-S (SOIC); LS7266R1-TS (TSSOP)

**PIN ASSIGNMENT - TOP VIEW**
**28-Pin Package**



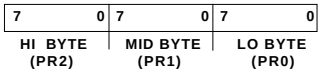

**LS7266R1 Registers:**

LS7266R1 has a set of registers associated with each X and Y axis. All X-axis registers have the name prefix X, whereas all Y-axis registers have the prefix Y. Selection of a specific register for Read/Write is made from the decode of the three most significant bits (D7-D5) of the data-bus. $\overline{CS}$ input enables the IC for Read/Write. C/$\overline{D}$ input selects between control and data information for Read/Write. Following is a complete list of LS7266R1 registers.

**Preset Registers: XPR and YPR**

Each of these PRs are 24-bit wide. 24-bit data can be written into a PR, one byte at a time, in a sequence of three data write cycles.

**PR**



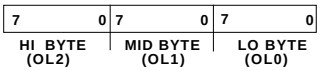

**Counters: XCNTR and YCNTR**

Each of these CNTRs are 24-bit synchronous Up/Down counters. The count clocks for each CNTR is derived from its associated A/B inputs. Each CNTR can be loaded with the content of its associated PR.

**Output Latches: XOL and YOL**

Each OL is 24-bits wide. In effect, the OLs are the output ports for the CNTRs. Data from each CNTR can be loaded into its associated OL and then read back on the data-bus, one byte at a time, in a sequence of three data Read cycles.

**OL**

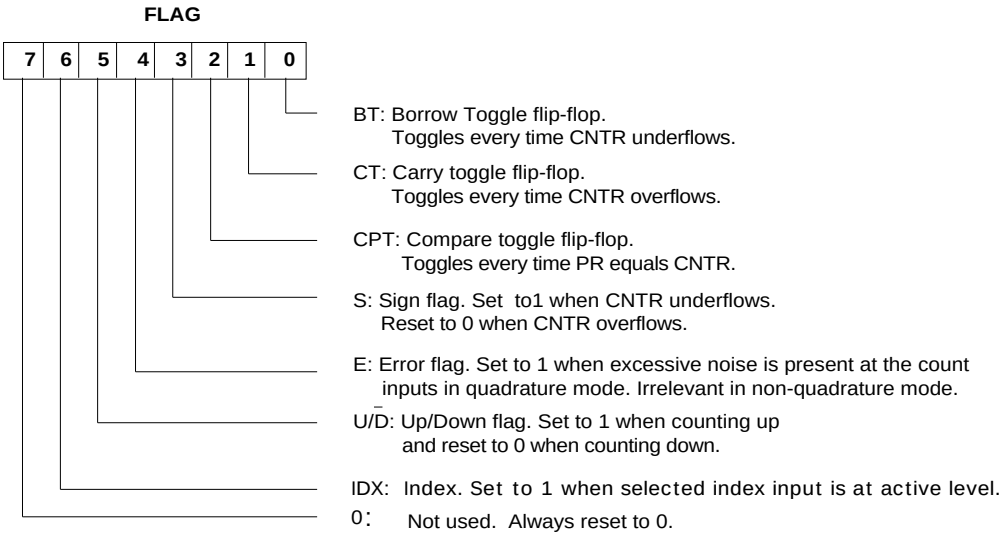



**Byte Pointers: XBP and YBP**

The Read and Write operations on an OL or a PR always accesses one byte at a time. The byte that is accessed is addressed by one of the BPs. At the end of every data Read or Write cycle on an OL or a PR, the associated BP is automatically incremented to address the next byte.

**Flag Register: XFLAG and YFLAG**

The FLAG registers hold the status information of the CNTRs and can be read out on the data bus. The E bit of a FLAG register is set to 1 when the noise pulses at the quadrature inputs are wide enough to be validated by the input filter circuits. E = 1 indicates excessive noise at the inputs but not a definite count error. Once set, E can only be reset via the RLD.



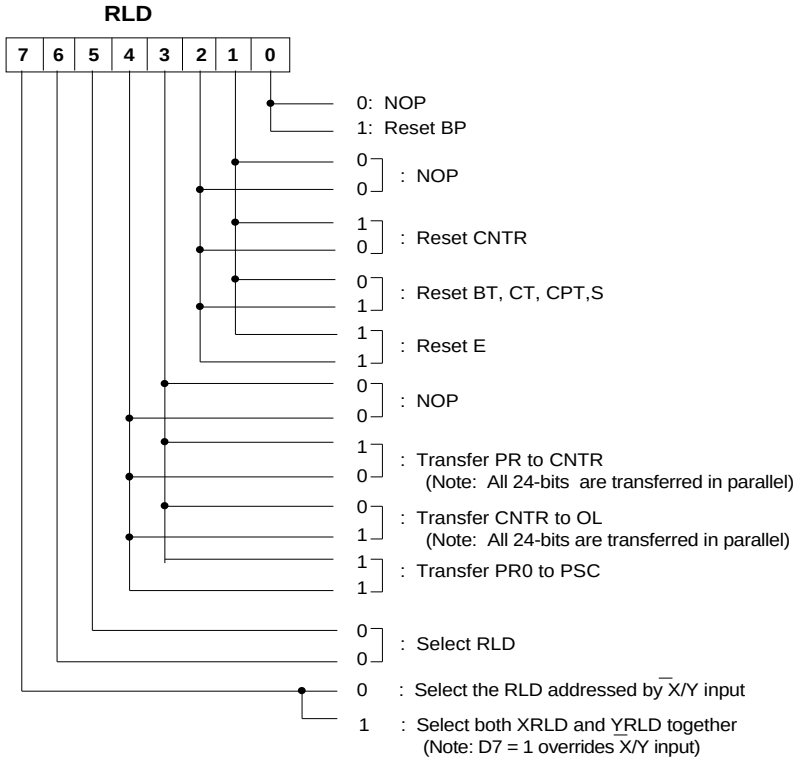

**Filter Clock Prescalers: XPSC and YPSC**

Each PSC is an 8-bit programmable modulo-N down counter, driven by the FCK clock. The factor N is down loaded into a PSC from the associated PR low byte register PR0. The PSCs provide the ability to generate independent filter clock frequencies for each channel. The PSCs generate the internal filter clock, FCKn used to validate inputs $X_A$, $X_B$, $Y_A$, $Y_B$ in the quadrature mode.

Final filter clock frequency $f_{FCKn} = ( f_{FCK}/(n+1) )$, where n = PSC = 0 to $FF_H$. For proper counting in the quadrature mode, $f_{FCKn} \geq 8f_{QA}$ (or $8f_{QB}$), where $f_{QA}$ and $f_{QB}$ are the clock frequencies at inputs A and B. In non-quadrature mode filter clock is not needed and the FCK input (Pin 2), should be tied to $V_{DD}$.

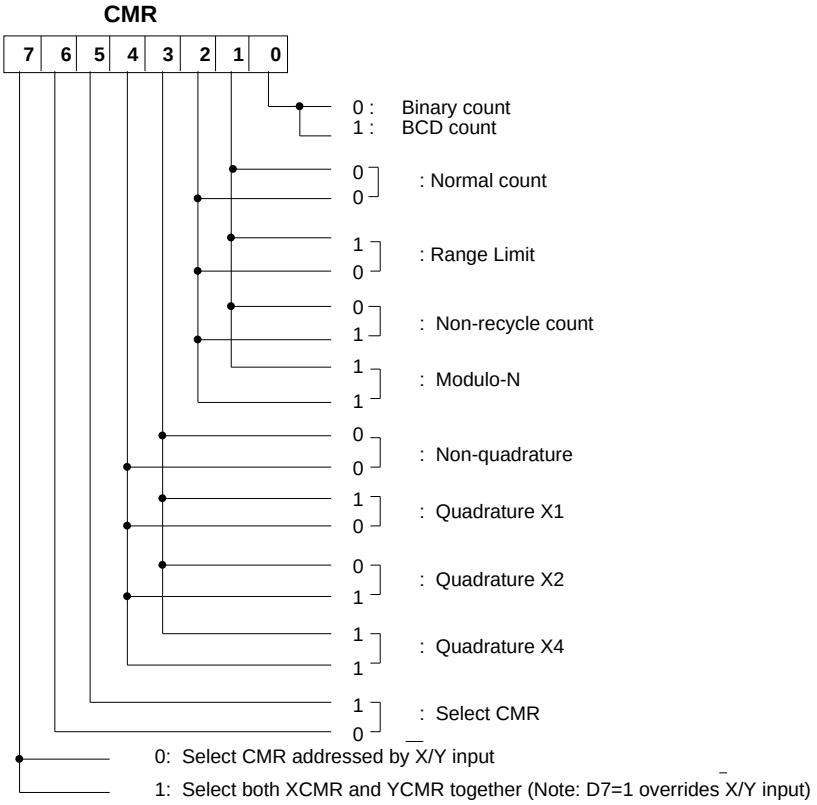
**Reset and Load Signal Decoders: XRLD and YRLD**

Following functions can be performed by writing a control byte into an RLD: Transfer PR to CNTR, Transfer CNTR to OL, reset CNTR, reset FLAG and reset BP.



**Counter Mode Registers: XCMR and YCMR**

The CNTR operational mode is programmed by writing into the CMRs.




**DEFINITIONS OF COUNT MODES:**

**Range Limit**. In range limit count mode, an upper and a lower limit is set, mimicking limit switches in the mechanical counterpart. The upper limit is set by the content of the PR and the lower limit is set to be 0. The CNTR freezes at CNTR = PR when counting up and at CNTR=0 when counting down. At either of these limits, the counting is resumed only when the count direction is reversed.

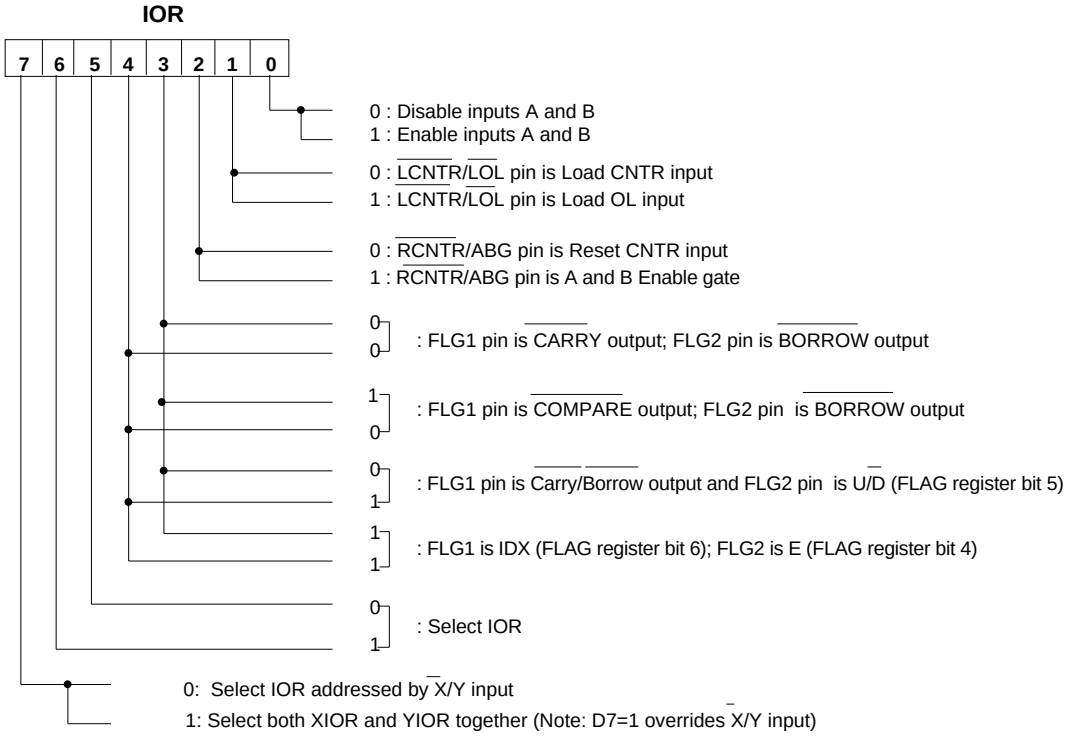
**Non-Recycle**. In non-recycle count mode, the CNTR is disabled, whenever a count overflow or underflow takes place. The end of cycle is marked by the generation of a Carry (in Up Count) or a Borrow (in Down Count). The CNTR is re-enabled when a reset or load operation is performed on the CNTR.

**Modulo-N.** In modulo-N count mode, a count boundary is set between 0 and the content of PR. When counting up, at CNTR=PR, the CNTR is reset to 0 and the up count is continued from that point. When counting down, at CNTR=0, the CNTR is loaded with the content of PR and down count is continued from that point.

The modulo-N is true bidirectional in that the divide-by-N output frequency is generated in both up and down direction of counting for same N and does not require the complement of N in the UP instance. In frequency divider application, the modulo-N output frequency can be obtained at either the $\overline{\text{Compare}}$ (FLG1) or the Borrow (FLG2) output. Modulo-N output frequency, $f_N = (f_i / (N+ 1))$ where $f_i$ = Input count frequency and N = PR.

**Input/Output Control Register: XIOR and YIOR**

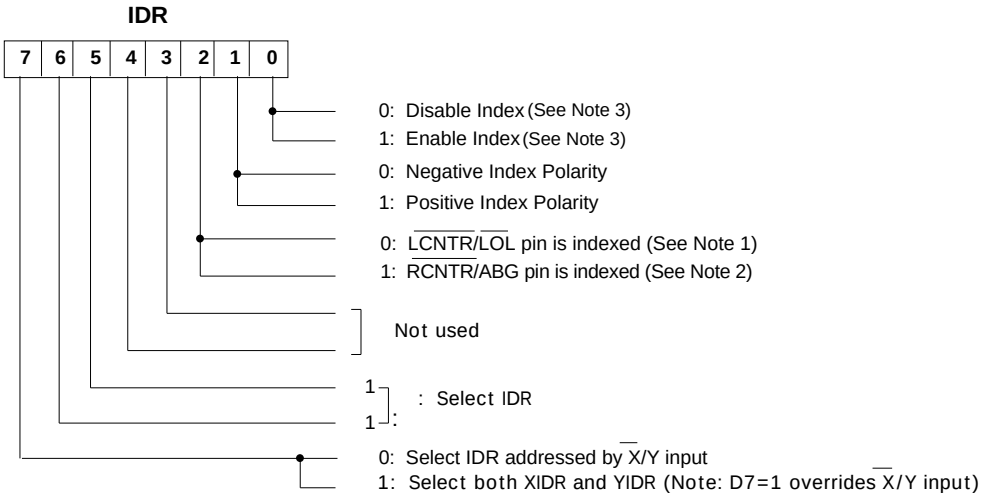
The functional modes of the programmable input and output pins are written into the IORs.




**INDEX CONTROL REGISTERS: XIDR and YIDR**

Either the LCNTR/LOL or the RCNTR/ABG inputs can be initialized to operate as an index input. When initialized as such, the index signal from the encoder, applied to one of these inputs performs either the Reset CNTR or the Load CNTR or the Load OL operation synchronously with the quadrature clocks. Note that only one of these inputs can be selected as the Index input at a time and hence only one type of indexing function can be performed in any given set-up.

**The index function must be disabled in non-quadrature count mode.**




**Note 1:** Function selected for this pin via IOR, becomes the operating INDEX function.

**Note 2:** RCNTR/ABG input must also be initialized as the reset CNTR input via IOR

**Note 3:** "Enable Index" causes the synchronous mode for the selected index input (as described in Pin 18 and Pin 19 sections of the I/O Description) to be enabled. "Disable Index" causes the non-synchronous mode to be enabled. The input, however, is not disabled in either selection.

## REGISTER ADDRESSING MODES

| D7 | D6 | D5 | C/$\overline{D}$ | $\overline{RD}$ | $\overline{WR}$ | X/$\overline{Y}$ | $\overline{CS}$ | FUNCTION |
|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | 1 | Disable both axes for Read/Write |
| X | X | X | 0 | 1 | ⌐⌐ | 0 | 0 | Write to XPR byte segment addressed by XBP (Note 3) |
| X | X | X | 0 | 1 | ⌐⌐ | 1 | 0 | Write to YPR byte segment addressed by YBP (Note 3) |
| 0 | 0 | 0 | 1 | 1 | ⌐⌐ | 0 | 0 | Write to XRLD |
| 0 | 0 | 0 | 1 | 1 | ⌐⌐ | 1 | 0 | Write to YRLD |
| 1 | 0 | 0 | 1 | 1 | ⌐⌐ | X | 0 | Write to both XRLD and YRLD |
| 0 | 0 | 1 | 1 | 1 | ⌐⌐ | 0 | 0 | Write to XCMR |
| 0 | 0 | 1 | 1 | 1 | ⌐⌐ | 1 | 0 | Write to YCMR |
| 1 | 0 | 1 | 1 | 1 | ⌐⌐ | X | 0 | Write to both XCMR and YCMR |
| 0 | 1 | 0 | 1 | 1 | ⌐⌐ | 0 | 0 | Write to XIOR |
| 0 | 1 | 0 | 1 | 1 | ⌐⌐ | 1 | 0 | Write to YIOR |
| 1 | 1 | 0 | 1 | 1 | ⌐⌐ | X | 0 | Write to both XIOR and YIOR |
| 0 | 1 | 1 | 1 | 1 | ⌐⌐ | 0 | 0 | Write to XIDR |
| 0 | 1 | 1 | 1 | 1 | ⌐⌐ | 1 | 0 | Write to YIDR |
| 1 | 1 | 1 | 1 | 1 | ⌐⌐ | X | 0 | Write to both XIDR and YIDR |
| X | X | X | 0 | 0 | 1 | 0 | 0 | Read XOL byte segment addressed by XBP (Note 3) |
| X | X | X | 0 | 0 | 1 | 1 | 0 | Read YOL byte segment addressed by YBP (Note 3) |
| X | X | X | 1 | 0 | 1 | 0 | 0 | Read XFLAG |
| X | X | X | 1 | 0 | 1 | 1 | 0 | Read YFLAG |

**X = Don't Care**

**Note 3**: Relevant BP is automatically incremented at the trailing edge of $\overline{RD}$ or $\overline{WR}$ pulse

**Absolute Maximum Ratings:**

| Parameter | Symbol | Values | Unit |
|---|---|---|---|
| Voltage at any input | $V_{IN}$ | $V_{SS}$ - 0.3 to $V_{DD}$ + 0.3 | V |
| Supply Voltage | $V_{DD}$ | +7.0 | V |
| Operating Temperature | $T_A$ | -25 to +80 | $^oC$ |
| Storage Temperature | $T_{STG}$ | -65 to +150 | $^oC$ |

**DC Electrical Characteristics**. ($T_A$ = -25°C to +80°C, $V_{DD}$ = 4.5V to 5.5V)

| Parameter | Symbol | Min. Value | Max.Value | Unit | Remarks |
|---|---|---|---|---|---|
| Supply Voltage | $V_{DD}$ | 4.5 | 5.5 | V | - |
| Supply Current | $I_{DD}$ | - | 800 | µA | All clocks off |
| Input Logic Low | $V_{IL}$ | - | 0.8 | V | - |
| Input Logic High | $V_{IH}$ | 2.0 | - | V | - |
| Output Low Voltage | $V_{OL}$ | - | 0.5 | V | $I_{OSNK}$ = 5mA |
| Output High Voltage | $V_{OH}$ | $V_{DD}$ - 0.5 | - | V | $I_{OSRC}$ = 1mA |
| Input Leakage Current | $I_{ILK}$ | - | 30 | nA | - |
| Data Bus Leakage Current | $I_{DLK}$ | - | 60 | nA | Data bus off |
| Output Source Current | $I_{OSRC}$ | 1.0 | - | mA | $V_O$ = $V_{DD}$ - 0.5V |
| Output Sink Current | $I_{OSNK}$ | 5.0 | - | mA | $V_O$ = 0.5V |

**Transient Characteristics**. ($T_A$ = -25°C to +80°C, $V_{DD}$ = 4.5V to 5.5V)

| Parameter | Symbol | Min. Value | Max.Value | Unit | Remarks |
|---|---|---|---|---|---|
| **Read Cycle** (See Fig. 1) | | | | | |
| $\overline{RD}$ Pulse Width | $t_{r1}$ | 50 | - | ns | - |
| $\overline{CS}$ Set-up Time | $t_{r2}$ | 50 | - | ns | - |
| $\overline{CS}$ Hold Time | $t_{r3}$ | 0 | - | ns | - |
| C/$\overline{D}$ Set-up Time | $t_{r4}$ | 50 | - | ns | - |
| C/$\overline{D}$ Hold Time | $t_{r5}$ | 10 | - | ns | - |
| $\overline{X}$/Y Set-up Time | $t_{r6}$ | 50 | - | ns | - |
| $\overline{X}$/Y Hold Time | $t_{r7}$ | 10 | - | ns | - |
| Data Bus Access Time | $t_{r8}$ | 50 | - | ns | Access starts when both $\overline{RD}$ and $\overline{CS}$ are low. |
| Data Bus Release Time | $t_{r9}$ | - | 25 | ns | Release starts when either $\overline{RD}$ or $\overline{CS}$ is terminated. |
| Back to Back Read delay | $t_{r10}$ | 60 | - | ns | - |
| **Write Cycle** (See Fig. 2) | | | | | |
| $\overline{WR}$ Pulse Width | $t_{W1}$ | 30 | - | ns | - |
| $\overline{CS}$ Set-up Time | $t_{W2}$ | 30 | - | ns | - |
| $\overline{CS}$ Hold Time | $t_{W3}$ | 0 | - | ns | - |
| C/$\overline{D}$ Set-up Time | $t_{W4}$ | 30 | - | ns | - |
| C/$\overline{D}$ Hold Time | $t_{W5}$ | 10 | - | ns | - |
| $\overline{X}$/Y Set-up Time | $t_{W6}$ | 30 | - | ns | - |
| $\overline{X}$/Y Hold Time | $t_{W7}$ | 10 | - | ns | - |
| Data Bus Set-up Time | $t_{W8}$ | 30 | - | ns | - |
| Data Bus Hold Time | $t_{W9}$ | 10 | - | ns | - |
| Back to Back Write Delay | $t_{W10}$ | 60 | - | ns | - |
| **Quadrature Mode** (See Fig. 3-5) | | | | | |
| FCK High Pulse Width | $t_1$ | 14 | - | ns | - |
| FCK Low Pulse Width | $t_2$ | 14 | - | ns | - |
| FCK Frequency | $f_{FCK}$ | - | 35 | MHz | - |
| Mod-n Filter Clock(FCKn)Period | $t_3$ | 28 | - | ns | $t_3 = (n+1)(t_1+t_2)$, where n = PSC = 0 to FF$_H$ |
| FCKn frequency | $f_{FCKn}$ | - | 35 | MHz | - |
| Quadrature Separation | $t_4$ | 57 | - | ns | $t_4 \geq 2t_3$ |
| Quadrature Clock Pulse Width | $t_5$ | 115 | - | ns | $t_5 \geq 4t_3$ |
| Quadrature Clock frequency | $f_{QA}$, $f_{QB}$ | - | 4.3 | MHz | $f_{QA} = f_{QB} = 1/8t_3$ |
| Quadrature Clock to Count Delay | $t_{Q1}$ | $5t_3$ | $6t_3$ | - | - |
| X1/X2/X4 Count Clock Pulse Width | $t_{Q2}$ | 28 | - | ns | $t_{Q2} = t_3$ |
| Index Input Pulse Width | $t_{idx}$ | 85 | - | ns | $t_{idx} \geq 3t_3$ |
| Index Skew from A | $t_{Ai}$ | - | 28 | ns | $t_{Ai} \leq t_3$ |
| Carry/Borrow/Compare Output Width | $t_{Q3}$ | 28 | - | ns | $t_{Q3} = t_3$ |
| **Non-Quadrature Mode** (See Fig. 6-7) | | | | | |
| Clock A - High Pulse Width | $t_6$ | 16 | - | ns | - |
| Clock A - Low Pulse Width | $t_7$ | 16 | - | ns | - |
| Direction Input B Set-up Time | $t_{8S}$ | 20 | - | ns | - |
| Direction Input B Hold Time | $t_{8H}$ | 20 | - | ns | - |
| Gate Input (ABG) Set-up Time | $t_{GS}$ | 20 | - | ns | - |
| Gate Input (ABG) Hold Time | $t_{GH}$ | 20 | - | ns | - |
| Clock Frequency (non-Mod-N) | $f_A$ | - | 30 | MHz | $f_A = (1/(t_6 + t_7))$ |
| Clock Frequency (Mod-N) | $f_{AN}$ | - | 25 | MHz | - |
| Clock to Carry or Borrow Out Delay | $t_9$ | - | 30 | ns | - |
| Carry or Borrow Out Pulse Width | $t_{10}$ | 16 | - | ns | $t_{10} = t_7$ |
| Load CNTR, Reset CNTR and Load OL Pulse Width | $t_{11}$ | 20 | - | ns | - |
| Clock to Compare Out Delay | $t_{12}$ | 50 | - | ns | - |

## INPUTS/OUTPUTS

**X-AXIS I/Os:**

**XA** (Pin 20) X-axis count input A
**XB** (Pin 21) X-axis count input B

Either quadrature encoded clocks or non-quadrature clocks can be applied to XA and XB. In quadrature mode XA and XB are digitally filtered and decoded for UP/DN clock. In non-quadrature mode, the filter and the decoder circuits are by-passed. Also, in non-quadrature mode XA serves as the count input and XB as the UP/DOWN direction control input, with XB = 1 selecting Up Count mode and XB = 0, selecting Down Count mode.

**$\overline{\text{XLCNTR}}$/$\overline{\text{XLOL}}$** (Pin 19) X-axis programmable input, to operate as either direct load XCNTR or direct load XOL or synchronous load XCNTR or synchronous load XOL. The synchronous load mode is intended for interfacing with the encoder Index output in quadrature clock mode. In direct load mode, a logic low level is the active level at this input. In synchronous load mode the active level can be programmed to be either logic low or logic high. Both quarter-cycle and half-cycle Index signals are supported by this input in the indexed Load mode. The synchronous function must be disabled in non-quadrature count mode (See description of IDR on P. 4)

**$\overline{\text{XRCNTR}}$/XABG** (Pin 18) X-axis programmable input to operate either as direct reset XCNTR or count enable/disable gate or synchronous reset XCNTR. The synchronous reset XCNTR mode is intended for interfacing with the encoder Index output in quadrature clock mode. In direct reset XCNTR mode, a logic low level is the active level at this input whereas in synchronous reset XCNTR mode the active level can be programmed to be either a logic low or a logic high. Both quarter-cycle and half-cycle index signals are supported by this input in the indexed reset CNTR mode. The synchronous function must be disabled in non-quadrature count mode (See description of IDR on P. 4). In count enable/disable mode, a logic high at this input enables the counter and a logic low level disables the counter.

**XFLG1** (Pin 22) X-axis programmable output to operate either as XCARRY (Active low), or XCOMPARE (generated when XPR=XCNTR; Active low), or XIDX (XFLAG bit 6) or XCARRY/XBORROW (Active low).

**XFLG2** (Pin 23) X-axis programmable output to operate as either XBORROW (Active low) or XU/$\overline{\text{D}}$ (XFLAG bit 5) or XE (XFLAG bit 4).

**Y-AXIS I/Os:**

All the X-axis inputs/outputs are duplicated for the Y-axis with similar functionalities.

**YA** (Pin 25)
**YB** (Pin 24)
**$\overline{\text{YLCNTR}}$/$\overline{\text{YLOL}}$** (Pin 1)
**$\overline{\text{YRCNTR}}$/YABG** (Pin 28)
**YFLG1** (Pin 27)
**YFLG2** (Pin 26)

**COMMON I/Os:**

**$\overline{\text{WR}}$** (Pin 14) Write input. Control/data bytes are written at the trailing edge of low level pulse applied to this input.

**$\overline{\text{RD}}$** (Pin 16) Read input. A low level applied to this input enables the FLAGs and OLs to be read on the data bus.

**$\overline{\text{CS}}$** (Pin 15) Chip select input. A low level applied to this input enables the chip for Read and Write.

**C/$\overline{\text{D}}$** (Pin 13) Control/Data input. This input selects between a control register or a data register for Read/Write. When low, a data register is selected. When high, a control register is selected.

**D0-D7** (Pins 4-11) Data Bus input/output. The 8-bit three-state data bus is the I/O port through which all data transfers take place between the LS7266R1 and the host processor.

**FCK** (Pin 2) Filter clock input in quadrature mode. The FCK is divided down internally by two 8-bit programmable prescalers, one for each channel.

**$\overline{\text{X}}$/Y** (Pin 17) Selects between X and Y axes for Read or Write. $\overline{\text{X}}$/Y = 0 selects X-axis and $\overline{\text{X}}$/Y = 1 selects Y-axis. $\overline{\text{X}}$/Y is overridden by D7 =1 in Control Write Mode (C/$\overline{\text{D}}$ = 1).

**$V_{DD}$** (Pin 3) +5VDC

**$V_{SS}$** (Pin 12) GND

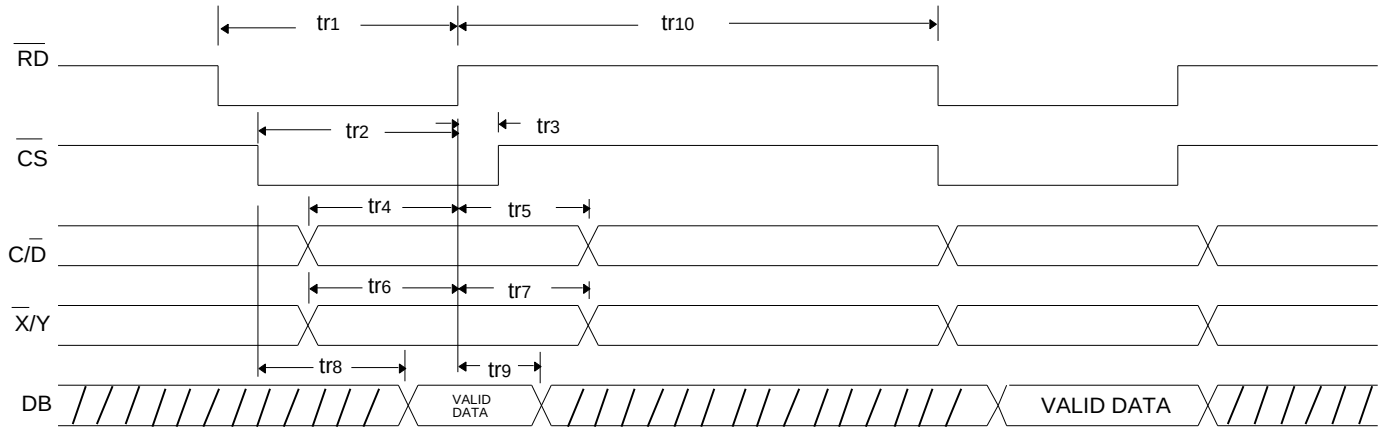



**FIGURE 1. READ CYCLE**

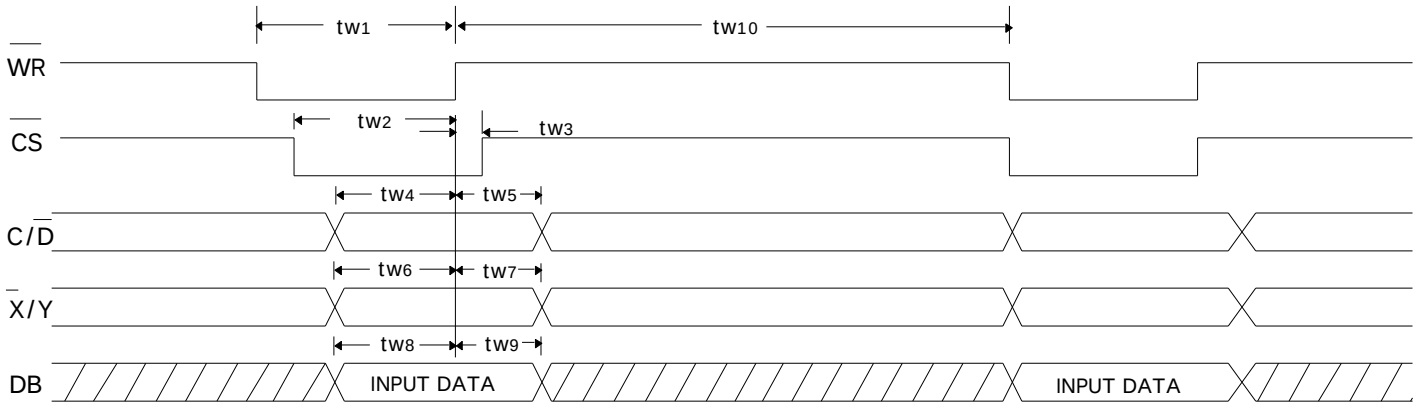



**FIGURE 2. WRITE CYCLE**

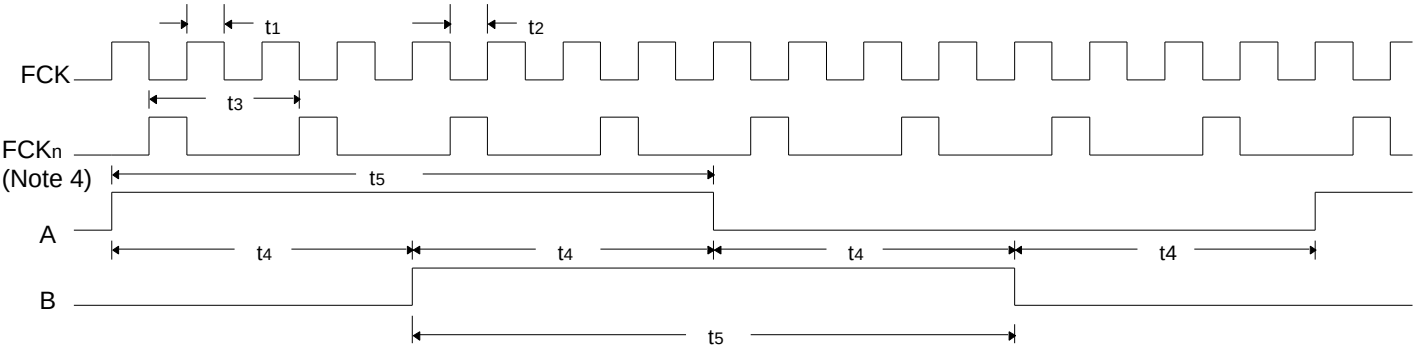



**FIGURE 3. FILTER CLOCK FCK AND QUADRATURE CLOCKS A AND B**

**Note 4**: FCKn is the final modulo-n internal filter clock, arbitrarily shown here as modulo-1.

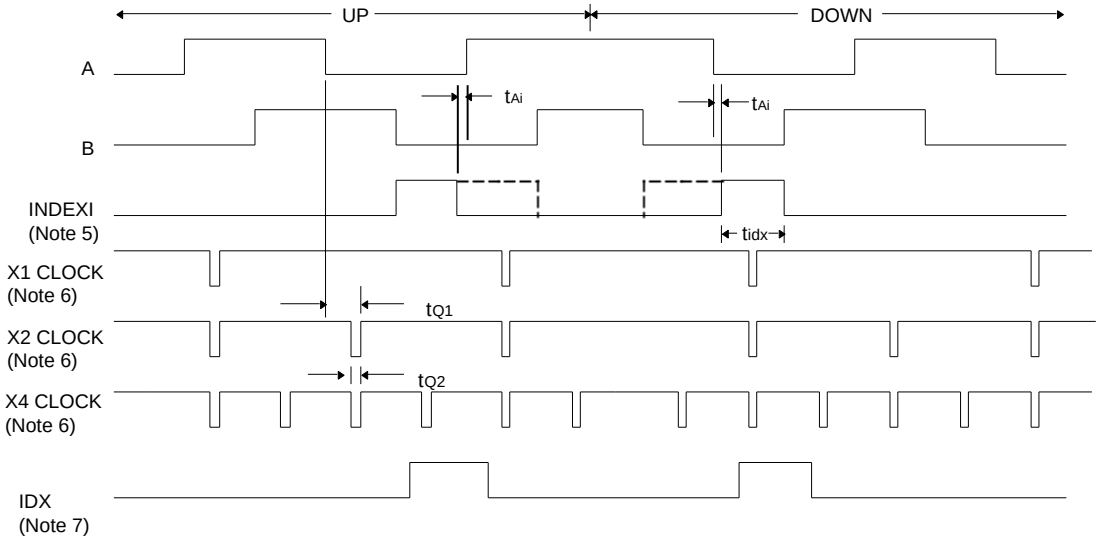



**FIGURE 4. QUADRATURE CLOCK A, B AND INDEX INPUT**

**Note 5**: Shown here is positive index with solid line depicting 1/4 cycle index and dotted line depicting 1/2 cycle index. Either LCNTR/LOL or RCNTR/ABG input can be used as the INDEX input.

**Note 6**: X1, X2 and X4 clocks are the final internal Up/Down count clocks derived from filtered and decoded Quadrature Clock inputs, A and B.

**Note 7**: IDX is the synchronized internal "load OL" or "load CNTR" or "reset CNTR" signal based on LCNTR/LOL or RCNTR/ABG input being selected as the INDEX input, respectively. This signal is identical with FLAG register bit 6.

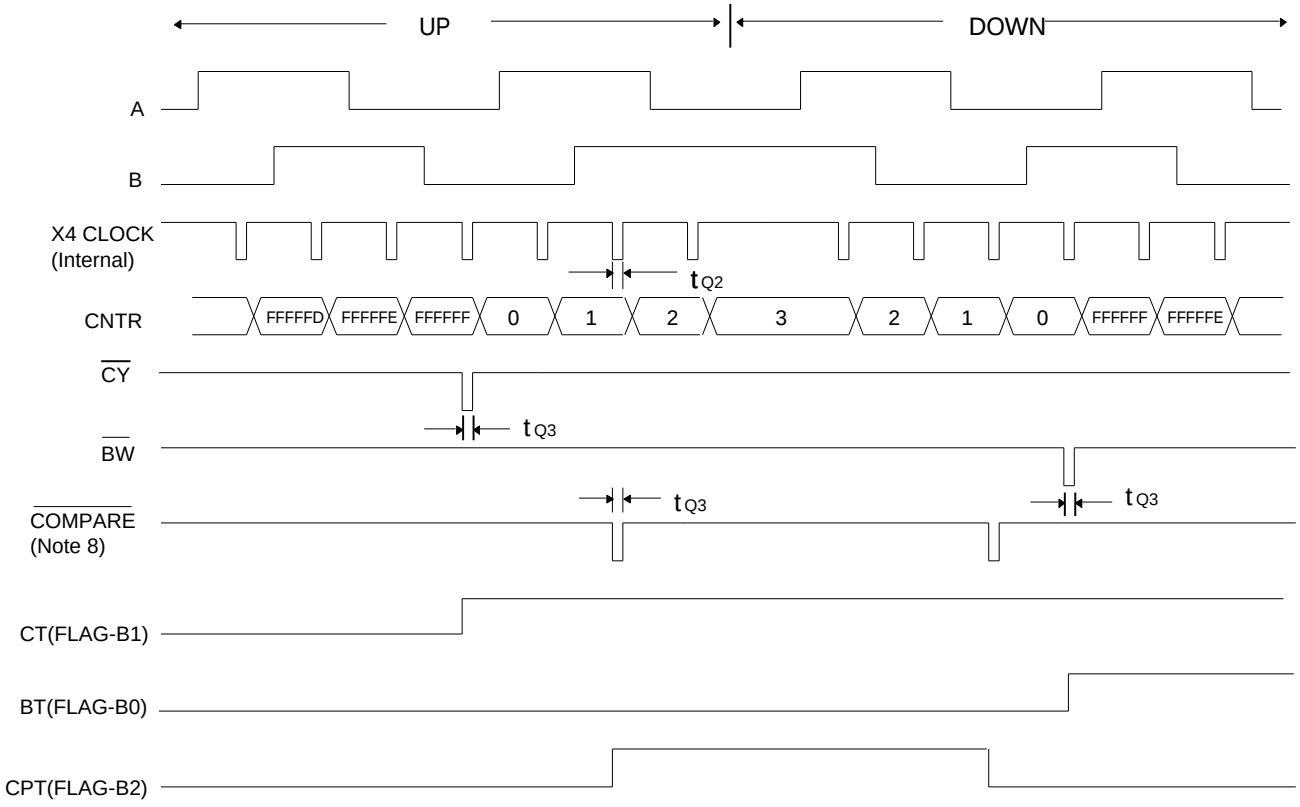



**FIGURE 5. CARRY, BORROW, COMPARE, CARRY TOGGLE, BORROW TOGGLE AND COMPARE TOGGLE IN X4 QUADRATURE, NORMAL, BINARY COUNT MODE.**

**Note 8**: $\overline{\text{COMPARE}}$ is generated when PR = CNTR. In this timing diagram it is arbitrarily assumed that PR = 1.

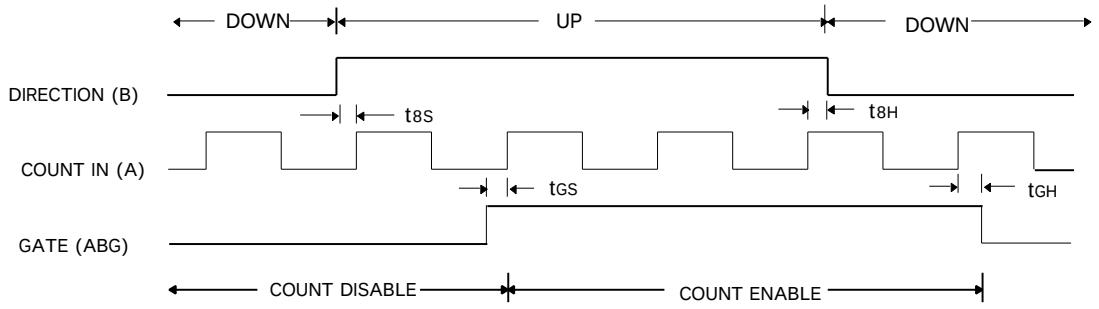



**FIGURE 6. COUNT (A), DIRECTION (B) AND GATE (ABG) INPUTS IN NON-QUADRATURE MODE**

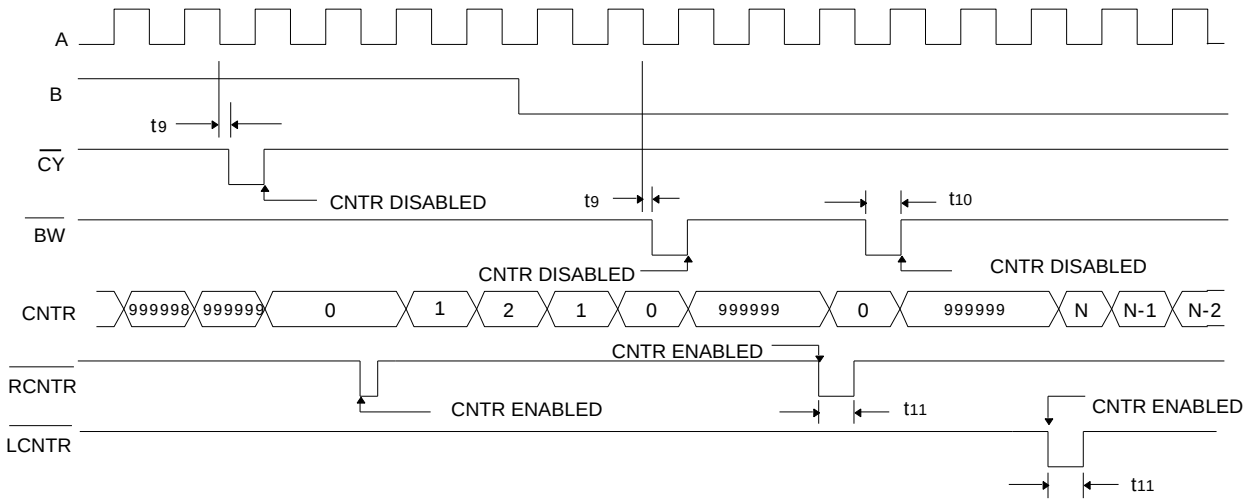



**FIGURE 7. NON-RECYCLE, NON-QUADRATURE, BCD MODE**

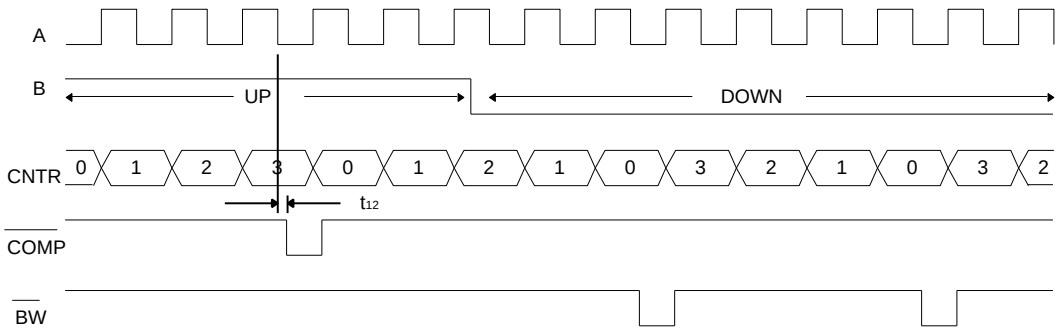



**FIGURE 8. MODULO - N, NON-QUADRATURE** (Shown with N = 3)

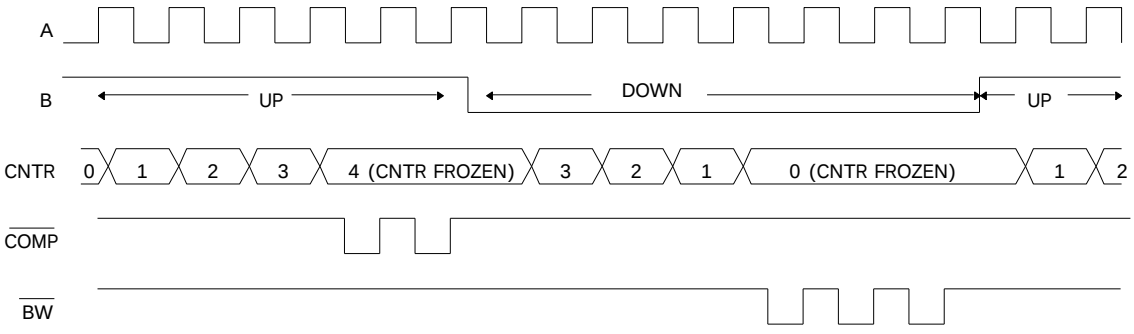



**FIGURE 9. RANGE LIMIT, NON-QUADRATURE** (Shown with PR = 4)

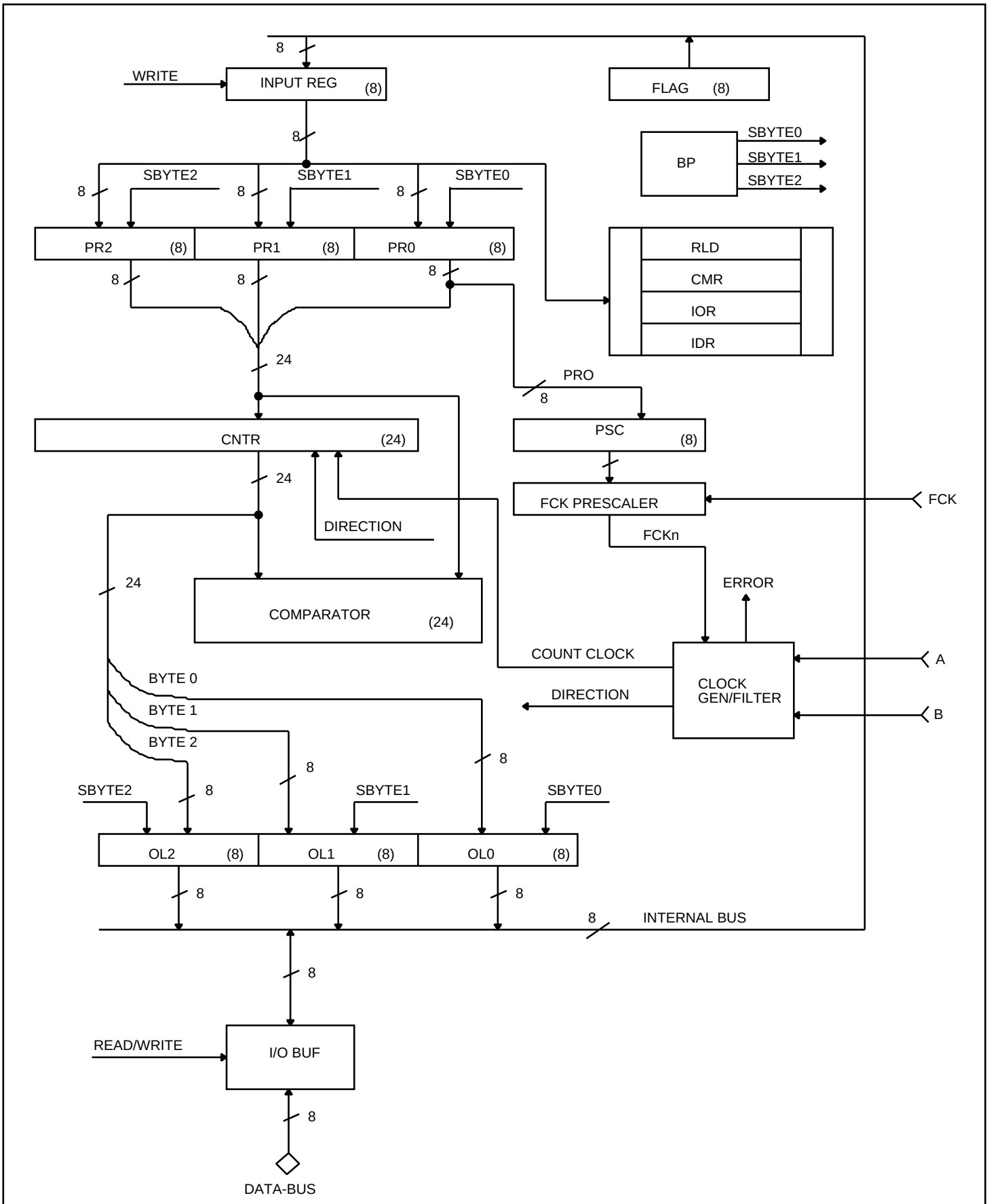



**FIGURE 10. SINGLE-AXIS BLOCK DIAGRAM SHOWING MAJOR DATA PATHS**

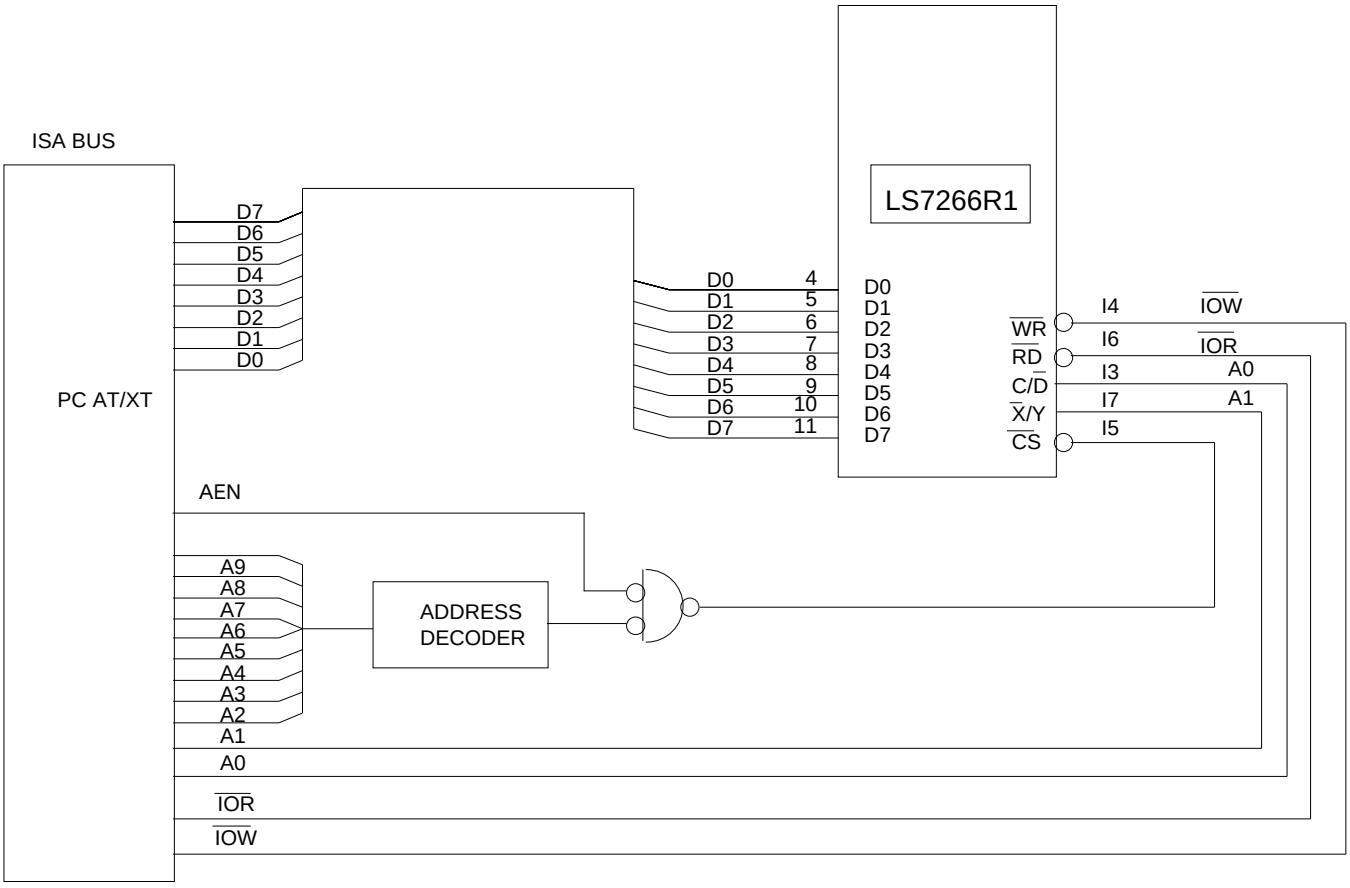



**FIGURE 11A. LS7266R1 INTERFACE EXAMPLES**

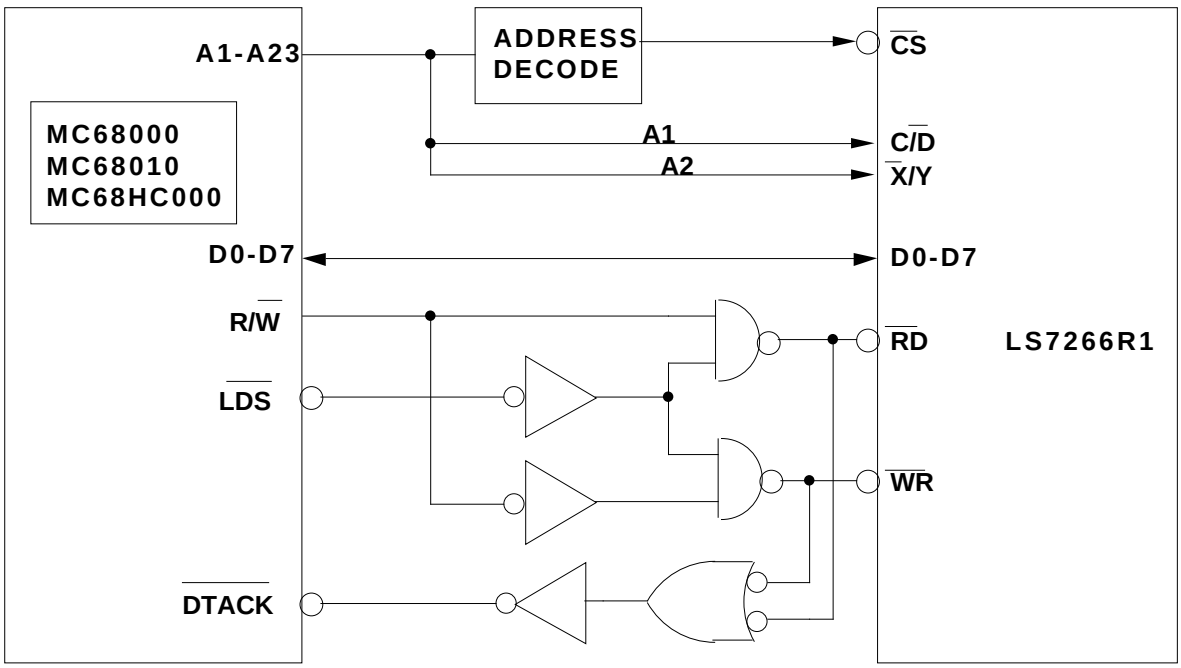



**FIGURE 11B. LS7266R1 INTERFACE EXAMPLES**

# C Sample Routines for Interfacing with LS7266R1

```
#include<stdlib.h>
#include <stdio.h>
#include <conio.h>

#define XDATA(arg) (arg +0)
#define XCMD (arg) (arg + 1)
#define YDATA (arg) (arg +2)
#define YCMD (arg) (arg +3)

// RLD Reg.
#define    RLD (arg) (arg | 0x80)
#define XRLD (arg) (arg | 0)
#define YRLD (arg) XRLD(arg)
#define Rst_BP 0x01
#define Rst_CNTR 0x02
#define Rst_FLAGS 0x04
#define Rst_E 0x06
#define Trf_PR_CNTR 0x08
#define Trf_CNTR_OL 0x10
#define Trf_PS0_PSC 0x18

//CMR Reg.
#define    CMR(arg) (arg | 0xA0)
#define XCMR(arg) (arg | 0x20)
#define YCMR(arg)  XCMR(arg)
#define BINCnt      0x00
#define BCDCnt     0x01
#define NrmCnt     0x00
#define RngLmt    0x02
#define NRcyc      0x04
#define ModN       0x06
#define NQDX       0x00
#define QDX1       0x08
#define QDX2       0x10
#define QDX4       0x18

//IOR Reg.
#define  IOR(arg) (arg | 0xC0)
#define XIOR(arg) (arg | 0x40)
#define YIOR(arg) XIOR(arg)
#define DisAB      0x00
#define EnAB       0x01

#define LCNTR         0x00
#define LOL           0x02
#define RCNTR        0x00
#define ABGate        0x04
#define CYBW          0x00
#define CPBW          0x08
#define CB_UPDN     0x10
#define IDX_ERR      0x18

// IDR
#define  IDR(arg) (arg | 0xE0)
#define XIDR(arg) (arg | 0x60)
#define YIDR(arg) XIDR(arg)
#define DisIDX     0x00
#define EnIDX      0x01
#define NIDX       0x00
#define  PIDX      0x02
#define LIDX       0x00
#define RIDX       0x04
```

```
void Init_7266(int Addr);
/*   Initialize 7266 as follows
     Modulo N count mode for N = 0x123456
     Binary Counting
     Index on LCNTR/LOL Input
     CY and BW outputs
     RCNTR/ABG controls Counters
     A and B Enabled
*/
void Init_7266(int Addr)
{
     /Setup IOR Reg.
     outp(XCMD(Addr),IOR(DisAB + LOL + ABGate + CYBW));    //Disable Counters and Set CY BW Mode

     //Setup RLD Reg.
     outp(XCMD(Addr),RLD(Rst_BP + Rst_FLAGS));      //Reset Byte Pointer(BP) And Flags
     outp(XDATA(Addr),0x06);           //Load 6 to PR0 to setup Transfer to PS0
     outp(XCMD(Addr),RLD(Rst_E + Trf_PS0_PSC));     //Reset E Flag and Transfer PR0 to PSC
     outp(XCMD(Addr),RLD(Rst_BP + Rst_CNTR));       //Reset BP and Reset Counter

     //Setup IDR Reg.
     outp(XCMD(Addr),IDR(EnIDX + NIDX + LIDX));    //Enable Negative Index on LCNTR/LOL Input

    //Setup CMR Reg.
    outp(XCMD(Addr),CMR(BINCnt + ModN + QDX4)); //Set Binary Mondulo N Quadrature X4
```

```
    //Setup PR Reg. for Modulo N Counter to 0x123456
    outp(XDATA(Addr),0x56);    //Least significant Byte first
    outp(XDATA(Addr),0x34);    //then middle byte
    outp(XDATA(Addr),0x12);    //then most significant byte

    //Enable Counters
    outp(XCMD(Addr),IOR(EnAB));

}

/* Write_7266_PR
Input: Addr has Address of 7266 counter.
Data: has 24 bit data to be written to PR register
 */
void Write_7266_PR(int Addr,unsigned long Data);
void Write_7266_PR(int Addr,unsigned long Data)
{
    outp(XCMD(Addr),RLD(Rst_BP));          //Reset Byte Pointer to Synchronize Byte Writing
    outp(XDATA(Addr),(unsigned char)Data);
    Data >>= 8;
    outp (XDATA(Addr),(unsigned char)Data);
    Data >>= 8;
    outp(XDATA(Addr),(unsigned char)Data);
}

/* Read_7266_OL
    Input: Addr has Address of 7266 counter.
    Output: Data returns 24 bit OL register value.
*/
unsigned long Read_7266_OL(int Addr);
unsigned long Read_7266_OL(int Addr)
{   unsigned long Data=0;
    outp(XCMD(Addr),(RLD(Rst_BP + Trf_Cntr_OL));  //Reset Byte Pointer to Synchronize Byte reading and
                                                    Transferring of data from counters to OL.
     Data |=(unsigned long)inp(XDATA(Addr));          //read byte 0 from OL
     lrotr(Data,8);                                   //Rotate for next Byte
     Data |=(unsigned long)inp(XDATA(Addr));          //read byte 1 from OL
     lrotr(Data,8);                                   //Rotate for next Byte
     Data |=(unsigned long)inp(XDATA(Addr));  //read byte 2 from OL
     lrotr(Data,16);                                  //Rotate for last Byte
     return(Data);
}
/*   Get_7266_Flags
      Input: Addr has Address of 7266 counter.
      returns Flags of counter
*/
unsigned char Get_7266_Flags(int Addr);
unsigned char Get_7266_Flags(int Addr)
{
     return(inp(XCMD(Addr)));
}
```