

July 2008

## 24-BIT x 4-AXES QUADRATURE COUNTER

### FEATURES:

- Read/write registers for count and I/O modes.  
Count modes include: non-quadrature (Up/Down), Quadrature (x1, x2, x4), Free-run, Non-recycle, Modulo-n and Range limit
- Separate mode-control registers for each axis
- Interrupt output and interrupt mask register
- 40 MHz count frequency, 5V  
20 MHz count frequency, 3V
- Sets of 24-bit counters, preset registers, comparators and output latches and 8-bit status registers for each axis
- Digital filtering of the input quadrature clocks for noise immunity.
- 3-state Octal I/O bus
- 3V to 5.5V operating voltage range
- LS7566R-TS (TSSOP) - See Figure 1 -

### GENERAL DESCRIPTION:

The LS7566R consists of four identical modules of 24-bit programmable counters with direct interface to incremental encoders. The modules can be configured to operate as quadrature-clock counters or non-quadrature up/down counters. In both quadrature and non-quadrature modes, the modules can be further configured into free-running, non-recycle, modulo-n and range-limit count modes. The mode configuration is made through two 8-bit read/write addressable control registers, MDR0 and MDR1. Data can be ported to a 24-bit preset register PR, organized in directly addressable (write-only) byte0 [PR0], byte1 [PR1] and byte2 [PR2] segments. PR can be transferred to the 24-bit counter CNTR, either by instruction to CMR or by hardware input control. A 24-bit digital comparator perpetually checks for the equality of the CNTR and the PR and can be used to set an output flag when the equality occurs. For reading the CNTR, its instantaneous value can be transferred to a 24-bit output latch OL, either by instruction to CMR or by hardware input control. The OL in turn can be read in directly addressable (read-only) byte0 [OL0], byte1 [OL1] and byte2 [OL2] segments. An addressable (read-only) Octal status register STR, stores the count related status information such as CNTR overflow, underflow, count direction, etc. Data communication for read/write is performed through an Octal 3-state parallel I/O bus.

### REGISTER DESCRIPTION:

Following is a list of the hardware registers. There are four sets of registers, with name prefixes x0 through x3 to refer to axes x0 through x3.

PIN ASSIGNMENT - Top View

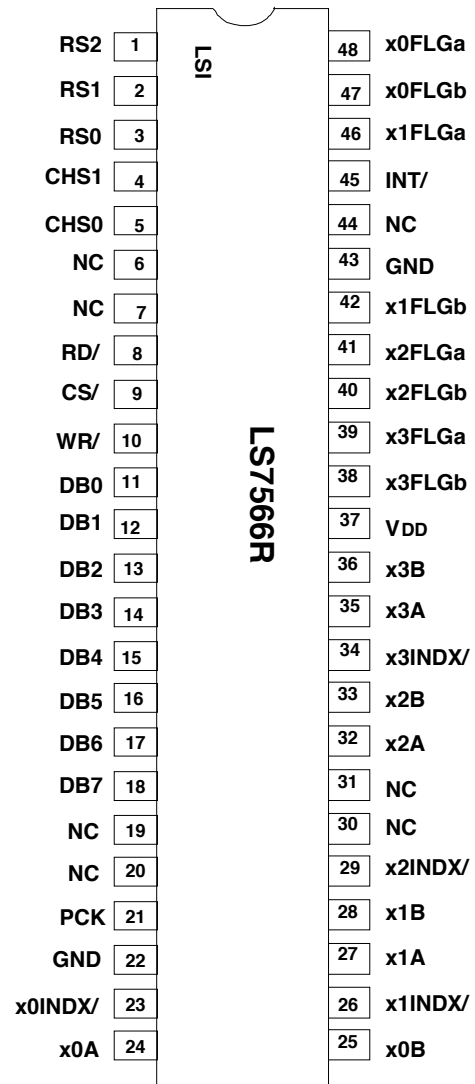
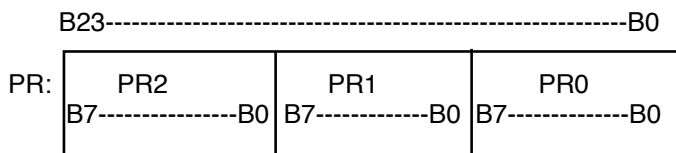


FIGURE 1

The information included herein is believed to be accurate and reliable. However, LSI Computer Systems, Inc. assumes no responsibilities for inaccuracies, nor for any infringements of patent rights of others which may result from its use.

### PR (x0PR, x1PR, x2PR, x3PR)

The PR is a 24-bit data register directly addressable for write in individual segments of byte0 [PR0], byte1 [PR1] and byte2 [PR2]. The PR serves as the input portal for the counter (CNTR), since the CNTR is not directly addressable for either read or write. In order to preset the CNTR to any desired value the data is first written into the PR and then transferred into the CNTR.



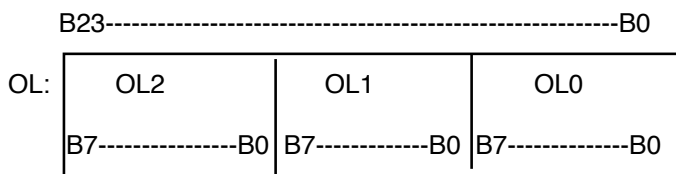
In **mod-n** and **range-limit** count modes the PR serves as the repository for the division factor n and the count range-limit, respectively. The PR can also be used to hold the compare data for the CNTR wherein the equality PR = CNTR sets an output flag.

### CNTR (x0CNTR, x1CNTR, x2CNTR, x3CNTR):

The CNTR is a 24-bit up/down counter which counts the up/down pulses resulting from the quadrature clocks applied at A and B inputs or alternatively, in non-quadrature mode, pulses applied at the A input. The CNTR is not directly accessible for read or write; instead it can be preloaded with data from the PR or it can port its own data out to the OL which in turn can be accessed by read operation. In both quadrature and non-quadrature modes, the CNTR can be further configured into either free-running or single-cycle or mod-n or range-limit mode.

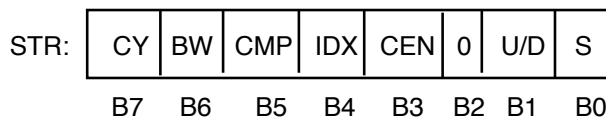
### OL (x0OL, x1OL, x2OL, x3OL):

The OL is a 24-bit register directly addressable for read in individual segments of byte0 [OL1], byte1 [OL1] and byte2 [OL2]. OL serves as the output portal for the CNTR. Snapshot of the CNTR data can be loaded in the OL without interfering with the counting process, which then can be accessed by read.



### STR (x0STR, x1STR, x2STR, x3STR):

The STR is an 8-bit status register indicating count related status.

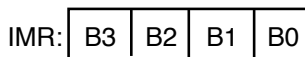


An individual STR bit is set to 1 when the bit related event has taken place. The STR is cleared to 0 at power-up. The STR can also be cleared through the control register CMR with the exception of bit1 (U/D) and bit3 (CEN). These two STR bits always indicate the instantaneous status of the count\_direction and count\_enable assertion/de-assertion. The STR bits are described below:

- B7 (CY): Carry; set by CNTR overflow
- B6 (BW): Borrow; set by CNTR underflow
- B5 (CMP): Set when CNTR = PR
- B4 (IDX): Set when INDX input is at active level
- B3 (CEN): Set when counting is enabled, reset when counting is disabled
- B2 (0): Always 0
- B1 (U/D): Set when counting up, reset when counting down
- B0 (S): Sign of count value; set when negative, reset when positive

### IMR:

The IMR is a trans-axis global register used for masking out the interrupt function of individual axes. It is a 4-bit read/write register with the following bit assignments.

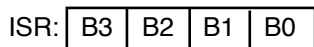


- B0 = 0: disable axis 0 interrupt  
= 1: enable axis 0 interrupt
- B1 = 0: disable axis 1 interrupt  
= 1: enable axis 1 interrupt
- B2 = 0: disable axis 2 interrupt  
= 1: enable axis 2 interrupt
- B3 = 0: disable axis 3 interrupt  
= 1: enable axis 3 interrupt

A write to IMR places the lower nibble of the databus into the IMR with identical bit map. A read of IMR produces a joint read of IMR and ISR (interrupt status register), with IMR occupying the lower nibble and ISR occupying the upper nibble of the databus.

**ISR:**

The ISR is a trans-axis global register used to hold the interrupt assertion status of all the axes. It is a 4-bit read-only register with the following bit assignment.



- B0 = 0: axis\_0 interrupt cleared  
= 1: axis\_0 interrupt asserted
- B1 = 0: axis\_1 interrupt cleared  
= 1: axis\_1 interrupt asserted
- B2 = 0: axis\_2 interrupt cleared  
= 1: axis\_2 interrupt asserted
- B3 = 0: axis\_3 interrupt cleared  
= 1: axis\_3 interrupt asserted

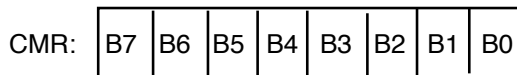
An ISR bit gets set when the FLG<sub>a</sub> output of the associated axis switches low. For this reason, in order for the interrupt to be enabled for any axis, its associated FLG<sub>a</sub> output must be enabled. In addition, the associated IMR bit must also be set for the interrupt to be enabled.

An individual ISR bit can be cleared through its axis relevant CMR register. The ISR is cleared upon power-up.

A read of ISR produces a joint read of ISR and IMR (interrupt mask register) with ISR occupying the upper nibble and IMR occupying the lower nibble of the databus.

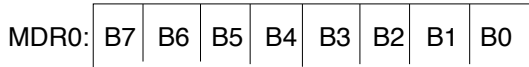
**CMR (x0CMR, x1CMR, x2CMR, x3CMR):**

The CMR is a write only register, which when written into, generates transient signals to perform load and reset operations as described below:



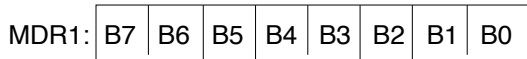
- B0 = 0: Nop  
= 1: Reset CNTR and sign to 0.  
(Should not be combined with load\_CNTR operation).
- B1 = 0: Nop  
= 1: Load CNTR from PR. Affects all 24 bits. (Should not be combined with reset\_CNTR operation)
- B2 = 0: Nop  
= 1: Load OL from CNTR. Affects all 24 bits.
- B3 = 0: Nop  
= 1: Reset STR. Affects status bits corresponding to carry, borrow, compare and index. Status bits corresponding to count\_enable, count direction and sign are not affected.
- B4 = 0: Nop.  
1: Master reset. Resets MDR0, MDR1, STR, CNTR, PR, OL, ISR and IMR
- B5 = 0: Nop  
1: Set sign bit
- B6 = 0: Nop  
1: Reset sign bit
- B7 = 0: Nop.  
1: Reset ISR bit for the selected axis

**MDR0 (x0MDR0, x1MDR0, x2MD0, x3MDR0):** The MDR0 is an 8-bit read/write register which configures the counting modes and the index input functionality. Upon power-up, the MDR0 is cleared to zero.



- B1B0 = 00: Non-quadrature count mode (A = clock, B = direction).
- = 01: x1 quadrature count mode (one count per quadrature cycle).
- = 10: x2 quadrature count mode (two counts per quadrature cycle).
- = 11: x4 quadrature count mode (four counts per quadrature cycle).
- B3B2 = 00: **Free-running** count mode.
- = 01: **Single-cycle** count mode (CNTR disabled with **carry** and **borrow**, re-enabled with **reset** or **load**)
- = 10: **Range-limit** count mode (up and down count ranges are limited between PR and zero, respectively. Counting freezes at these limits but resumes when the direction is reversed)
- = 11: **Modulo-n** count mode (input count clock frequency is divided by a factor of [n+1], where n = PR. In up direction, the CNTR is **cleared** to 0 at CNTR = PR and up count continues. In down direction, the CNTR is **preset** to the value of PR at CNTR = 0 and down count continues. A mod-n rollover marker pulse is generated at each limit at the FLG<sub>a</sub> output).
- B5B4 = 00: Disable INDX/ input.
- = 01: Configure INDX/ input as the load CNTR input (transfers PR to CNTR).
- = 10: Configure INDX/ as the reset \_CNTR input (clears CNTR to 0).
- = 11: Configure INDX/ as the load\_OL input (transfers CNTR to OL).
- B6 = 0: Asynchronous INDX/ input
- = 1: Synchronous INDX/ input (overridden in non-Quadrature Mode)
- B7 = 0: Input filter clock (PCK) division factor = 1. Filter clock frequency = fPCK.
- = 1: Input filter clock division factor = 2. Filter clock frequency = fPCK/2.

**MDR1 (x0MDR1, x1MDR1, x2MD1, x3MDR1):** The MDR1 is an 8-bit read/write register which configures the FLG<sub>a</sub> and FLG<sub>b</sub> output functionality. In addition, the MDR1 can be used to enable/disable counting. Upon power-up, the MDR1 is cleared to zero:



- B0 = 1: Enable Carry on FLG<sub>a</sub> (flags CNTR overflow; latched or unlatched logic low on carry).
- B1 = 1: Enable Borrow on FLG<sub>a</sub> (flags CNTR underflow, latched or unlatched logic low on borrow).
- B2 = 1: Enable Compare on FLG<sub>a</sub> (In free-running count mode a latched or unlatched logic low is generated in both up and down count directions at CNTR = PR. In contrast, in range-limit and mod-n count modes a latched or unlatched low is generated at CNTR = PR in the up-count direction only).
- B3 = 1: Enable index on FLG<sub>a</sub> (flags index, latched or unlatched logic low when INDX/ input is at active level)
- B5B4 = 00: FLG<sub>b</sub> disabled (fixed high)
- = 01: FLG<sub>b</sub> = **Sign**, high for negative signifying CNTR underflow, low for positive.
- = 10: FLG<sub>b</sub> = **Up/Down** count direction, high in count-up, low in count-down.
- B6 = 0: Enable counting.
- = 1: Disable counting.
- B7 = 0: FLG<sub>a</sub> is latched.
- = 1: FLG<sub>a</sub> is non-latched and instantaneous.

**NOTE:** Carry, Borrow, Compare and Index can all be simultaneously enabled on FLG<sub>a</sub>.

**I/O PINS:** The following is a description of the input/output pins.

**RS0**(Pin 3), **RS1** (Pin 2), **RS2** (Pin1).

Inputs. These three inputs select the hardware registers for read/write access according to Table 1.

**TABLE 1**

<b>CS/</b>	<b>RS2</b>	<b>RS1</b>	<b>RS0</b>	<b>RD/</b>	<b>WR/</b>	<b>SELECTED REGISTER</b>	<b>OPERATION</b>
1	x	x	x	x	x	none	none
x	x	x	x	0	0	none	none
x	x	x	x	1	1	none	none
0	0	0	0	0	1	[ISR:IMR]	READ (NOTE 2)
0	0	0	1	0	1	MDR0	READ
0	0	1	0	0	1	MDR1	READ
0	0	1	1	0	1	STR	READ
0	1	0	0	0	1	OL0	READ
0	1	0	1	0	1	OL1	READ
0	1	1	0	0	1	OL2	READ
0	1	1	1	0	1	none	none
0	0	0	0	1	0	IMR	WRITE
0	0	0	1	1	0	MDR0	WRITE
0	0	1	0	1	0	MDR1	WRITE
0	0	1	1	1	0	none	none
0	1	0	0	1	0	PR0	WRITE
0	1	0	1	1	0	PR1	WRITE
0	1	1	0	1	0	PR2	WRITE
0	1	1	1	1	0	CMR	WRITE

**Note 1.** x indicates don't care case.

**Note 2.** DB0 through DB3 contain IMR B0 through B3; DB4 through DB7 contain ISR B0 through B3.

**CHS0** (Pin 5), **CHS1** (Pin 4)

Inputs. These two inputs select one of four axes for read/write access according to the following table. The registers within the axis are selected according to Table 1.

**TABLE 2**

<b>CHS1</b>	<b>CHS0</b>	<b>AXIS</b>
0	0	x0
0	1	x1
1	0	x2
1	1	x3

**RD/** (Pin 8) Input. A low on RD/ input accesses an addressed register for read and places the data on the octal databus, DB<7:0>. The register selection is made according to Table 1.

**CS/** (Pin 9) Input. A low on the CS/ input enables the chip for read or write operation. When the CS/ input is high, read and write operations are disabled and the databus, DB<7:0> is placed in a high impedance state.

**WR/** (Pin 10) Input. A low pulse on the WR/ input writes the data on the databus, DB<7:0> into the addressed register according to Table 1. The write operation is completed at the trailing edge of the WR/ pulse.

**DB<7:0>** (Pin 18 thru Pin 11) Input/Output.

The octal databus DB<7:0> is the input/output portal for write and read data transfers between LS7566R and the outside world. During a read operation, when both CS/ and the RD/ inputs are low, DB<7:0> are outputs. During a write operation, when both CS/ and WR/ are low, DB<7:0> are inputs. When CS/ is high, DB<7:0> are in high impedance state independent of the states of RD/ and WR/.

**PCK** (Pin21) Input. A clock applied at PCK input is used for validating the logic states of the A and B quadrature clocks and the INDX/ input. The PCK input frequency,  $f_{PCK}$  is divided down by a factor of 1 or 2 according to bit7 of MDR0. The resultant clock is used to sample the logic levels of the

A, the B and the INDX/ inputs. If a logic level at any of these inputs remains stable for a minimum of two filter clock periods, it is validated as a correct logic state. The PCK input is common to all four axes, but the filter clock frequency for any axis is set by its associated MDR0 register.

In non-quadrature mode no filter clock is used and the PCK input should be tied to either VDD or GND.

**x0A** (pin 24), **x0B** (Pin 25) Inputs. These are the A and B count inputs in axis x0. These inputs can be configured to function either in quadrature mode or in non-quadrature mode. The configuration is made through MDR0. In quadrature mode, A and B clocks are 90 degrees out of phase. When A leads B in phase, the CNTR counts up and when B leads A in phase, the CNTR counts down.

In non-quadrature mode, A is the count input and B is the count direction control input. When B is **high**, positive transitions at the A input causes the CNTR to count **up**. Conversely, when B is **low**, the positive transition at the A input causes the CNTR to count **down**.

In quadrature mode, A and B inputs are sampled by an internal filter clock generated from the PCK input. In non-quadrature mode A and B inputs are not sampled and the count clocks are applied to the CNTR bypassing the filter circuit.

**x1A** (Pin 27), **x1B** (Pin 28), **x2A** (Pin 32), **x2B** (Pin33), **x3A** (Pin 35), **x3B** (Pin36)

These are the A and B inputs corresponding to axes x1, x2 and x3. Functionally, they are identical with the A and B inputs of axis x0.

**x0INDX/** (Pin 23) Input. The INDX/ input in axis x0. The INDX/ input can be configured by MDR0 to function as load\_CNTR or reset\_CNTR or load\_OL input. In quadrature mode, the INDX/ input can be configured to function in either synchronous or asynchronous mode. In synchronous mode, the INDX/ input is sampled with the same filter clock used for sampling the A and the B inputs and must satisfy the phase relationship with A and B in which INDX/ is at the active level during a minimum of quarter cycle of both A and B high or both A and B low.

In asynchronous mode the INDX/ input is not sampled and can be applied in any phase relationship with respect to the A and B inputs.

The INDX/ input can be either enabled or disabled in both synchronous and asynchronous modes.

**x1INDX/** (Pin 26), **x2INDX/** (Pin 29), **x3INDX/** (Pin 34)

These are the INDX inputs corresponding to axes x1, x2 and x3. Functionally, they are identical with the INDX input of axis x0.

### **INT/** (Pin 45) **Output**

The INT/ output is the common interrupt output for all the axes. When any of the ISR bits gets set, INT/ switches low indicating an asserted interrupt. The axis generating the interrupt can then be identified by reading the ISR register.

**x0FLGa** (Pin 48) Output. The FLGa output in axis x0. The FLGa output is configured by MDR1 register to function as either Carry or Borrow or Compare or Index flag. A Carry flag is generated when the CNTR overflows, a Borrow flag is generated when the CNTR underflows and a Compare flag is generated by the condition, CNTR=PR. When configured as Index, the FLGb output switches low when the INDX input is enabled and at active level. The FLGa can be configured to produce outputs in either latched mode or instantaneous mode. In the latched mode when the selected event of Carry or Borrow or Compare or Index has taken place, the FLGa switches low and remains low until the status register, STR is cleared. In the instantaneous mode a negative pulse is generated instantaneously when the event takes place. The FLGa output can be disabled to remain at a fixed logic high.

**x1FLGa** (Pin 46), **x2FLGa** (Pin 41), **x3FLGa** (Pin 39)

These outputs are the FLGa outputs corresponding to axes x1, x2 and x3 respectively. Functionally, they are identical with the FLGa output of axis x0.

**x0FLGb** (Pin 47) Output. The FLGb output in axis x0. The FLGb output is configured by MDR1 to function as either Sign or Up/Down status indicator.

When configured as Sign, the FLGb output remains high when CNTR is in an underflow state (caused by down counts at or below zero), indicating a negative number. When the CNTR counts up past zero, FLGb switches low, indicating a positive number.

When configured as Up/Down indicator, a high at the FLGb indicates that the current count direction is up (incremental) whereas a low indicates that the direction is down (decremental).

The FLGb output can be disabled to remain at a fixed logic high.

**x1FLGb** (Pin42), **x2FLGb** (Pin 40) **x3FLGb** (Pin 38)

These are the FLGb outputs corresponding to axes x1, x2 and x3 respectively. Functionally, they are identical with the FLGb output of x0.

**Absolute Maximum Ratings:**

Parameter	Symbol	Values	Unit
Voltage at any input	V <sub>IN</sub>	V <sub>SS</sub> - 0.3 to V <sub>DD</sub> + 0.3	V
Supply Voltage	V <sub>DD</sub>	+7.0	V
Operating Temperature	T <sub>A</sub>	-25 to +85	°C
Storage Temperature	T <sub>STG</sub>	-65 to +150	°C

**DC Electrical Characteristics.** (T<sub>A</sub> = -25°C to +85°C, V<sub>DD</sub> = 3V to 5.5V)

Parameter	Symbol	Min. Value	Max. Value	Unit	Remarks
Supply Voltage	V <sub>DD</sub>	3.0	5.5	V	-
Supply Current	I <sub>DD</sub>	-	800	μA	All clocks off
Input Logic Low	V <sub>IL</sub>	-	0.15V <sub>DD</sub>	V	-
Input Logic High	V <sub>IH</sub>	0.5V <sub>DD</sub>	-	V	-
Output Low Voltage	V <sub>OL</sub>	-	0.5	V	I <sub>OSNK</sub> = 5mA, V <sub>DD</sub> = 5V
Output High Voltage	V <sub>OH</sub>	V <sub>DD</sub> - 0.5	-	V	I <sub>OSRC</sub> = 1mA, V <sub>DD</sub> = 5V
Input Leakage Current	I <sub>ILK</sub>	-	30	nA	-
Data Bus Leakage Current	I <sub>DLK</sub>	-	60	nA	Data bus off
Data Bus Source Current	I <sub>OSRC</sub>	3.0	-	mA	V <sub>O</sub> = V <sub>DD</sub> - 0.5V, V <sub>DD</sub> = 5V
Data Bus Sink Current	I <sub>OSNK</sub>	8.0	-	mA	V <sub>O</sub> = 0.5V, V <sub>DD</sub> = 5V
FLGa, FLGb, INT/ Source	I <sub>OSRC</sub>	1.0	-	mA	V <sub>O</sub> = V <sub>DD</sub> - 0.5V, V <sub>DD</sub> = 5V
FLGa, FLGb, INT/ Sink	I <sub>OSNK</sub>	6.0	-	mA	V <sub>O</sub> = 0.5V, V <sub>DD</sub> = 5V

**Transient Characteristics.** (T<sub>A</sub> = -25° to +85°C)**For V<sub>DD</sub> = 3V to 5.5V**

Parameter	Symbol	Min. Value	Max. Value	Unit	Remarks
<b>Read Cycle</b> (See Fig. 2)					
RD/ Pulse Width	tr1	80	-	ns	-
CS/ Set-up Time	tr2	80	-	ns	-
CS/ Hold Time	tr3	0	-	ns	-
RS<2:0> Set-up Time	tr4	80	-	ns	-
RS<2:0> Hold Time	tr5	10	-	ns	-
CHS<1:0> Set-up Time	tr6	80	-	ns	-
CHS<1:0> Hold Time	tr7	10	-	ns	-
DB<7:0> Access Time	tr8	80	-	ns	Access starts when both RD/ and CS/ are low.
DB<7:0> Release Time	tr9	-	35	ns	Release starts when either RD/ or CS/ is terminated.
Back to Back Read delay	tr10	10	-	ns	-
<b>Write Cycle</b> (See Fig. 3)					
WR/ Pulse Width	tw1	45	-	ns	-
CS/ Set-up Time	tw2	45	-	ns	-
CS/ Hold Time	tw3	0	-	ns	-
RS<2:0> Set-up Time	tw4	45	-	ns	-
RS<2:0> Hold Time	tw5	10	-	ns	-
CHS<1:0> Set-up Time	tw6	45	-	ns	-
CHS<1:0> Hold Time	tw7	10	-	ns	-
DB<7:0> Set-up Time	tw8	45	-	ns	-
DB<7:0> Hold Time	tw9	10	-	ns	-
Back to Back Write Delay	tw10	90	-	ns	-

**For VDD = 3.3V ±10%**

Parameter	Symbol	Min. Value	Max. Value	Unit	Remarks
<b>Quadrature Mode</b> (See Fig. 4-6)					
PCK High Pulse Width	t1	24	-	ns	-
PCK Low Pulse Width	t2	24	-	ns	-
PCK Frequency	fPCK	-	20	MHz	-
Filter Clock(ff)Period	t3	50	-	ns	t3 = t1+ t2, MDR0 <7> =0
	t3	100	-	ns	t3 = t1+ t2, MDR0 <7> =1
Filter clock frequency	fr	-	20	MHz	fr = 1/ t3
Quadrature Separation	t4	52	-	ns	t4 > t3
Quadrature Clock Pulse Width	t5	105	-	ns	t5 > 2t3
Quadrature Clock frequency	fQA, fQB	-	4.5	MHz	fQA = fQB < 1/4t3
Quadrature Clock to Count Delay	tQ1	4t3	5t3	-	-
X1/X2/X4 Count Clock Pulse Width	tQ2	25	-	ns	tQ2 = t3/2
Quadrature Clock to FLGa delay	trda	4.5t3	5.5t3	ns	-
Quadrature Clock to FLGb delay	trdb	3t3	4t3	ns	-
FLGa to INT/ delay	trnt	0	-	ns	-
INDX/ Input Pulse Width (Synchronous)	tid	60	-	ns	tid > t4
INDX/ set-up time (Synchronous)	tis	10	-	ns	-
INDX/ hold time (Synchronous)	tih	10	-	ns	-
FLGa Output Width	trfw	50	-	ns	trfw t4

**Non-Quadrature Mode** (See Fig. 7-8)

Clock A - High Pulse Width	t6	24	-	ns	-
Clock A - Low Pulse Width	t7	24	-	ns	-
Direction Input B Set-up Time	t8s	24	-	ns	-
Direction Input B Hold Time	t8	20	-	ns	-
Clock Frequency	fA	-	20	MHz	fA = ( 1/ (t6 + t7) )
Clock to FLGa Out Delay	t9	-	40	ns	-
FLGa Out Pulse Width	t10	24	-	ns	t10 = t7
INDX/ Pulse Width (Asynchronous)	t11	30	-	ns	-

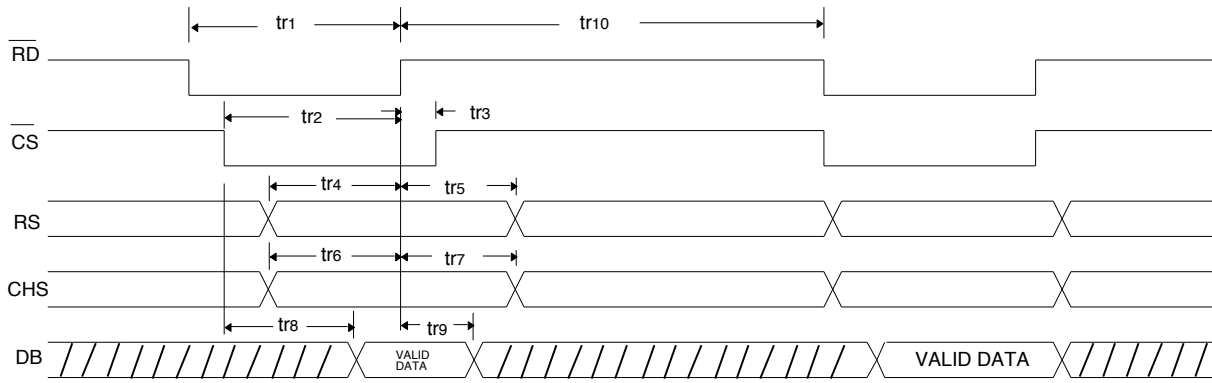
**For VDD = 5V ±10%**

Parameter	Symbol	Min. Value	Max. Value	Unit	Remarks
<b>Quadrature Mode</b> (See Fig. 4-6)					
PCK High Pulse Width	t1	12	-	ns	-
PCK Low Pulse Width	t2	12	-	ns	-
PCK Frequency	fPCK	-	40	MHz	-
Filter Clock(fr)Period	t3	25	-	ns	t3 = t1+ t2, MDR0 <7> =0
	t3	50	-	ns	t3 = t1+ t2, MDR0 <7> =1
Filter clock frequency	fr	-	40	MHz	-
Quadrature Separation	t4	26	-	ns	t4 > t3
Quadrature Clock Pulse Width	t5	52	-	ns	t5 > 2t3
Quadrature Clock frequency	fQA, fQB	-	9.6	MHz	fQA = fQB < 1/4t3
Quadrature Clock to Count Delay	tQ1	4t3	5t3	-	-
x1 / x2 / x4 Count Clock Pulse Width	tQ2	12	-	ns	tQ2 = t3/2
Quadrature Clock to FLGa delay	trda	4.5t3	5.5t3	ns	-
Quadrature Clock to FLGb delay	trdb	3t3	4t3	ns	-
FLGa to INT/ delay	trnt	0	-	ns	-
INDX/ Input Pulse Width (Synchronous)	tid	32	-	ns	tid > t4
INDX/ set-up time (Synchronous)	tis	5	-	ns	-
INDX/ hold time (Synchronous)	tih	5	-	ns	-
FLGa Output Width	trfw	24	-	ns	trfw t4

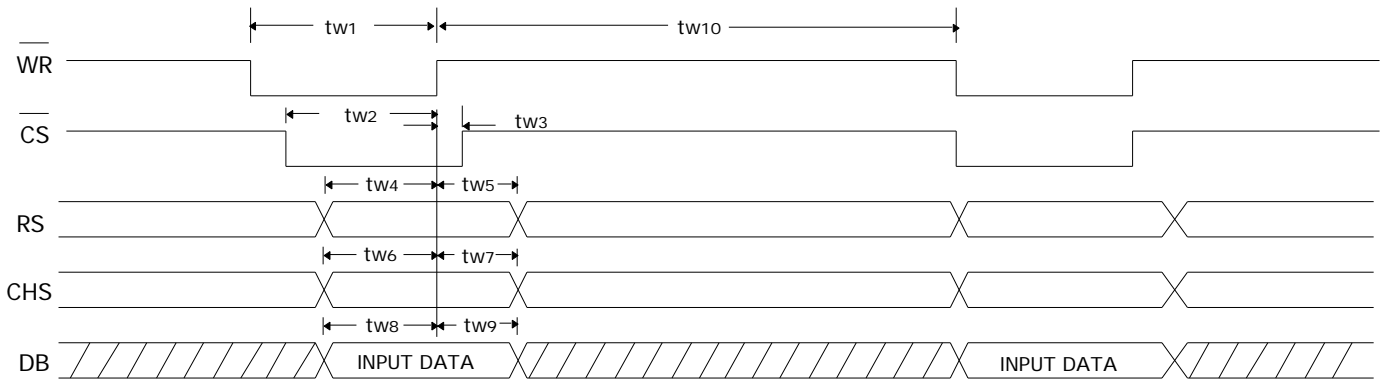
**Non-Quadrature Mode** (See Fig. 7-8)

Clock A - High Pulse Width	t6	12	-	ns	-
Clock A - Low Pulse Width	t7	12	-	ns	-
Direction Input B Set-up Time	t8	12	-	ns	-
Direction Input B Hold Time	t8	10	-	ns	-
Clock Frequency	fA	-	40	MHz	fA = ( 1/ (t6 + t7) )
Clock to FLGa Out Delay	t9	-	20	ns	-
FLGa Out Pulse Width	t10	12	-	ns	t10 = t7
INDX/ Pulse Width (Asynchronous)	t11	15	-	ns	-

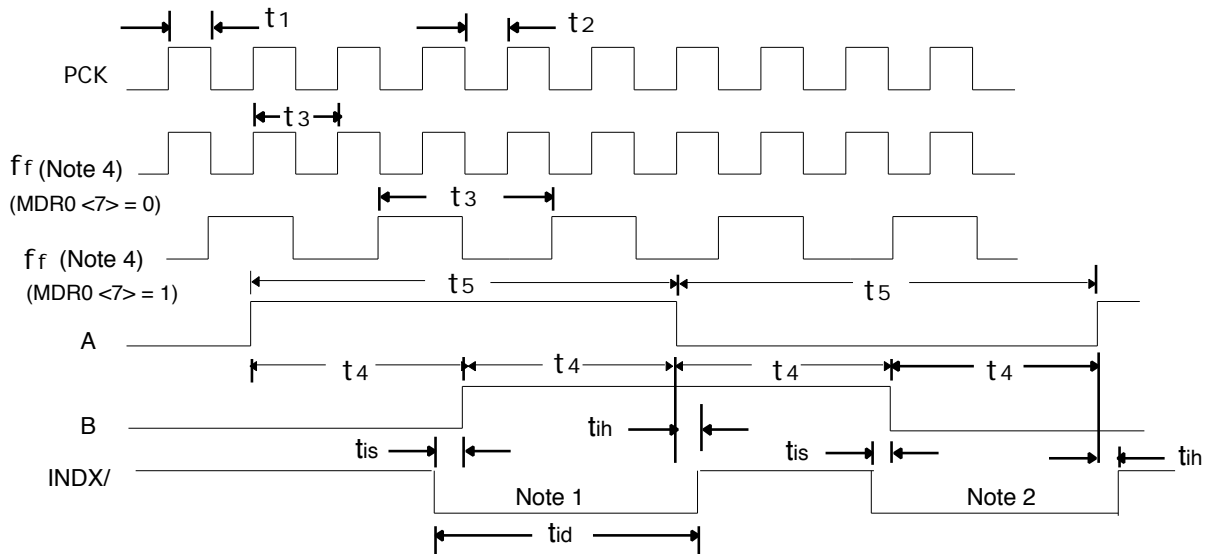




**FIGURE 2. READ CYCLE**

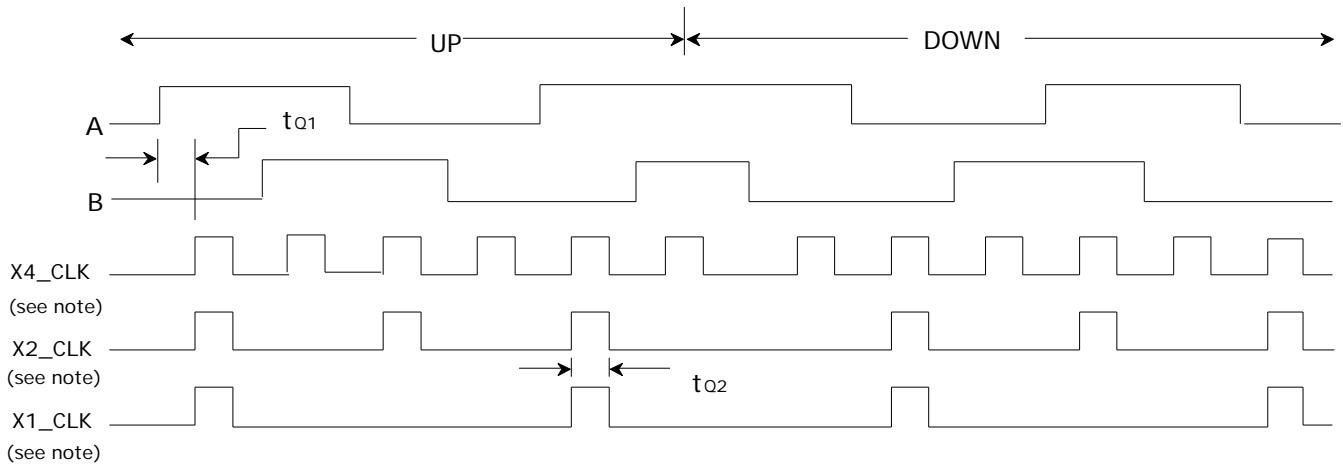


**FIGURE 3. WRITE CYCLE**



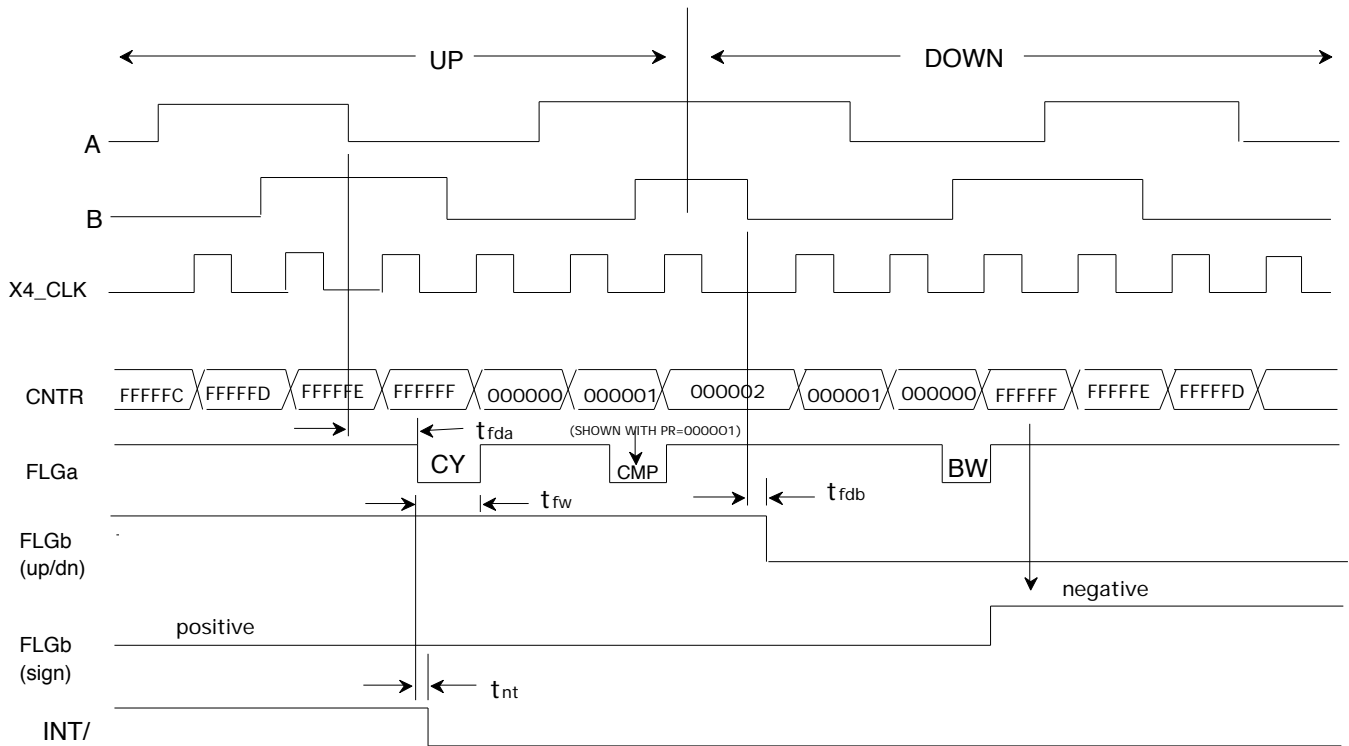
- Note 1.** Synchronous index coincident with both A and B high.
- Note 2.** Synchronous index coincident with both A and B low.
- Note 3.**  $f_f$  is the internal effective filter clock.

**FIGURE 4. PCK, A, B and INDX**



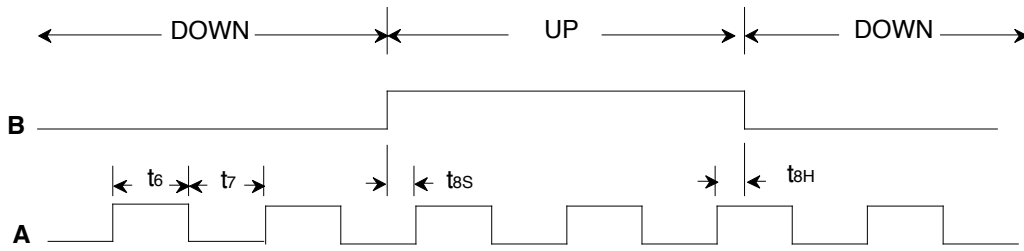
**NOTE.** x1, x2 and x4 CLKs are internal Up/Down clocks derived from filtered and decoded quadrature clocks.

**FIGURE 5. A/B QUADRATURE CLOCKS vs INTERNAL COUNT CLOCKS**

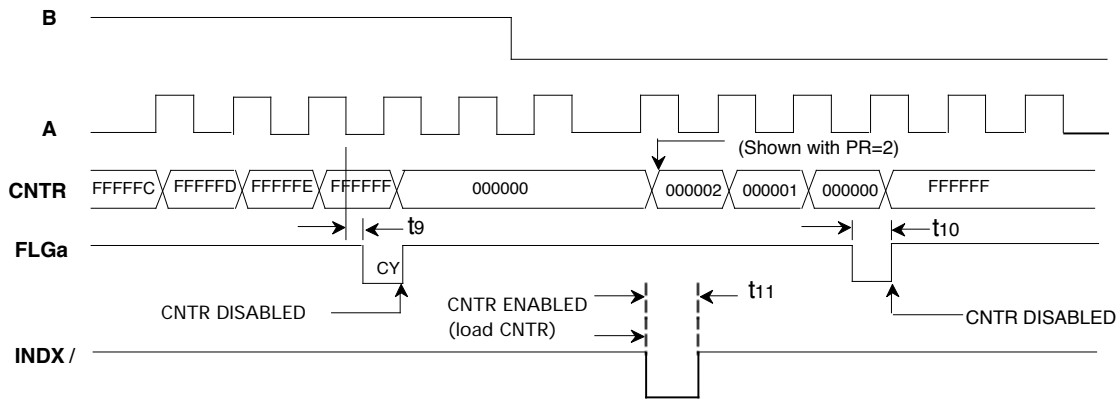


**NOTE.** FLGa is in non-latched mode.

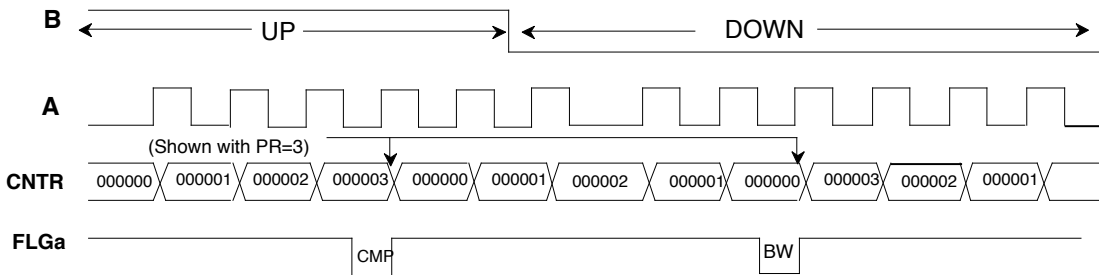
**FIGURE 6. QUADRATURE CLOCKS vs FLGa, FLGb and INT/ OUTPUTS**



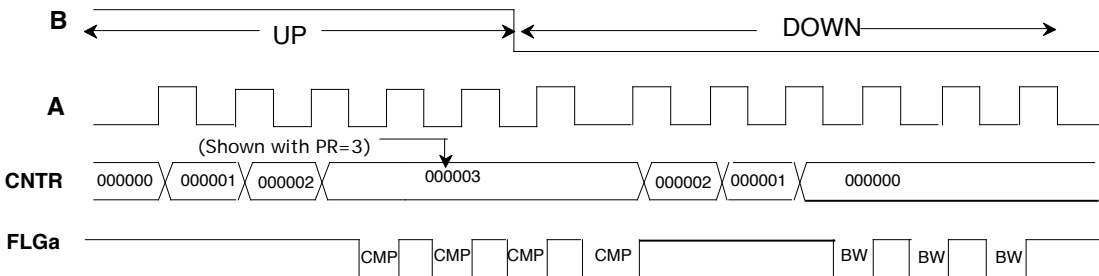
**FIGURE 7. COUNT (A) and DIRECTION (B) INPUTS IN NON-QUADRATURE MODE**



**FIGURE 8. SINGLE-CYCLE, NON-QUADRATURE**



**FIGURE 9. MODULO-N, NON-QUADRATURE**



**FIGURE 10. RANGE-LIMIT, NON-QUADRATURE**

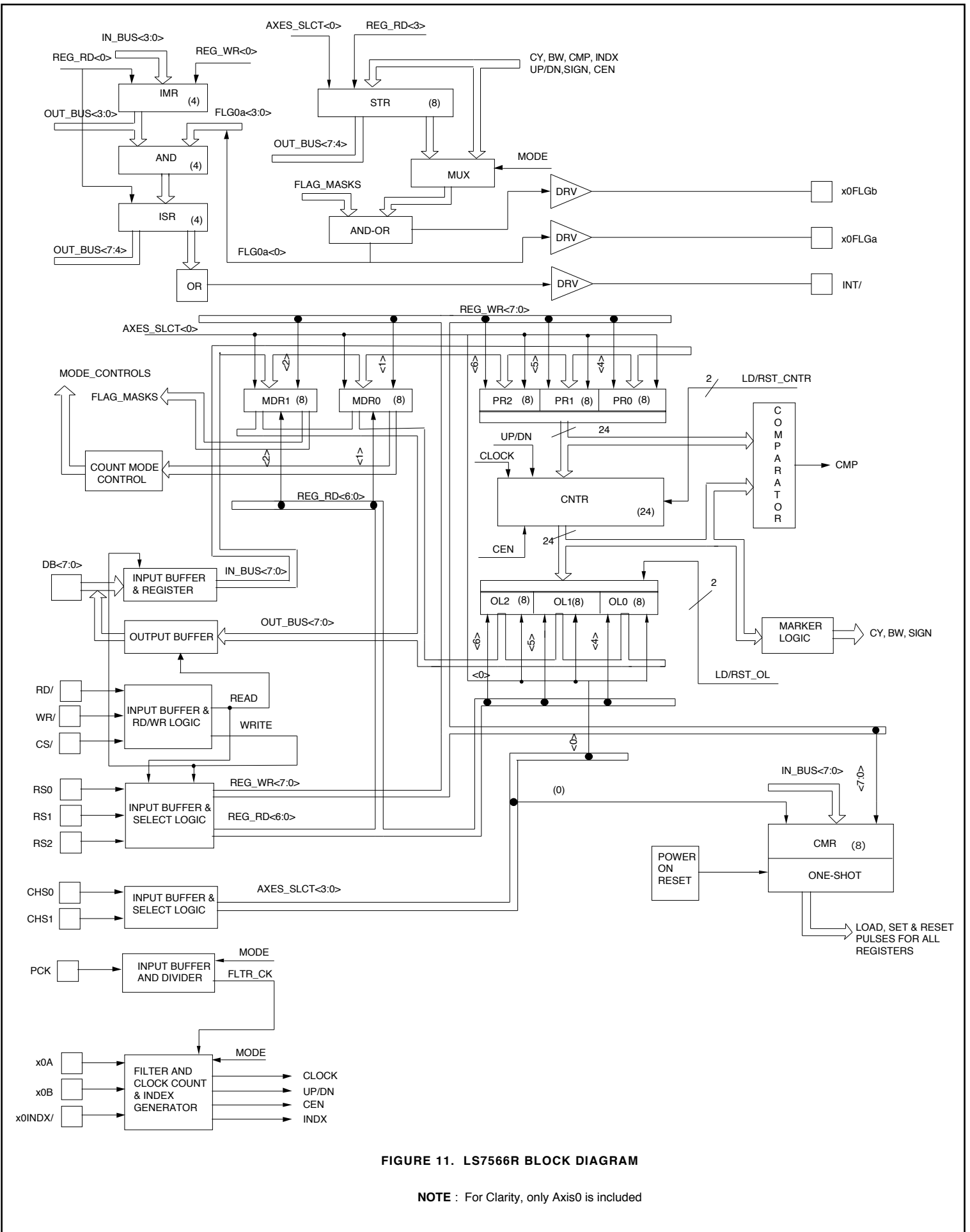


FIGURE 11. LS7566R BLOCK DIAGRAM

NOTE : For Clarity, only Axis0 is included

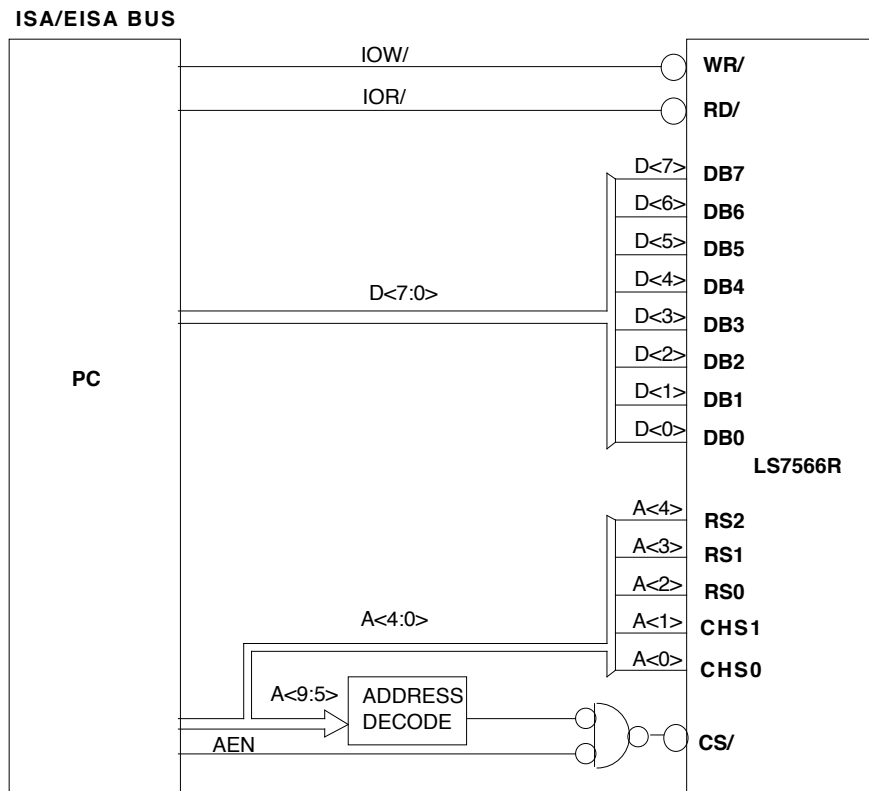


FIGURE 12. LS7566R TO ISA/EISA INTERFACE

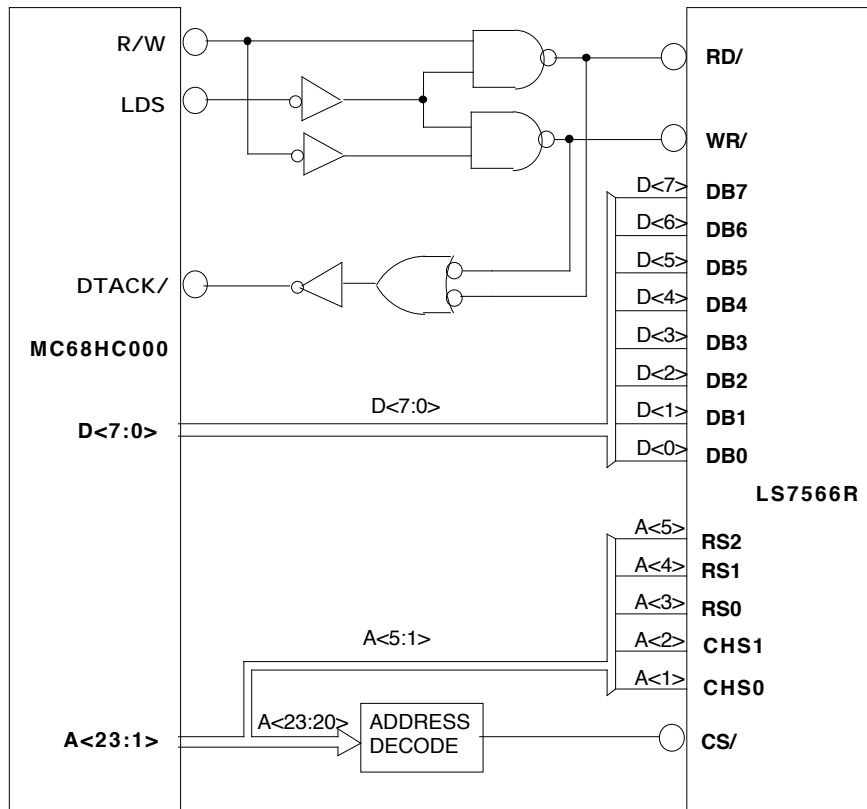


FIGURE 13. LS7566R TO MC68HC000 INTERFACE

## C Sample Routines for Interfacing with LS7566R

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

/*****
  ADDRESSES
  *****/

#define axis_0    0x00
#define axis_1    0x01
#define axis_2    0x02
#define axis_3    0x03

/*****
  READ ONLY ADDRESSES
  *****/

#define ISR_IMR(arg)    (arg + 0)
#define STR(arg)        (arg + 0x0C)
#define OL0(arg)        (arg + 0x10)
#define OL1(arg)        (arg + 0x14)
#define OL2(arg)        (arg + 0x18)

/*****
  WRITE ONLY ADDRESSES
  *****/

#define IMR(arg)        (arg + 0)
#define PR0(arg)        (arg + 0x10)
#define PR1(arg)        (arg + 0x14)
#define PR2(arg)        (arg + 0x18)
#define CMR(arg)        (arg)

/*****
  READ / WRITE ADDRESSES
  *****/

#define MDR0(arg)       (arg + 0x04)
#define MDR1(arg)       (arg + 0x08)

/*****
  OP CODES
  *****/

/* IMR (Interrupt Mask Register) */
#define Dis_Int_x0    0x00
#define En_Int_x0     0x01

#define Dis_Int_x1    0x00
#define En_Int_x1     0x02

#define Dis_Int_x2    0x00
#define En_Int_x2     0x04

#define Dis_Int_x3    0x00
#define En_Int_x3     0x08
```

```
/* CMR (Load / Reset Register) */
#define Rst_CNTR      0x01
#define Ld_CNTR      0x02
#define Ld_OL        0x04
#define Rst_STR      0x08    /* CEN, U/D, S not affected */
#define Master_Rst   0x10
#define Set_Sign     0x20
#define Rst_Sign     0x40
#define Rst_ISR      0x80
```

```
/* MDR0 (Counting Modes / Index Input Functionality) */
```

```
#define NQDX          0x00
#define QDX1         0x01
```

```
#define QDX2         0x02
#define QDX4         0x03
```

```
#define FreeRun      0x00
#define SngleCyc    0x04
#define RngLmt      0x08
#define ModN        0x0C
```

```
#define DisIDX       0x00
#define LCNTR       0x10
#define RCNTR       0x20
#define LOL         0x30
```

```
#define ASYN_IDX    0x00
#define SYN_IDX     0x40
```

```
#define FDF1        0x00
#define FDF2        0x80
```

```
/* MDR1 (FLGa / FLGb Output Functionality) */
```

```
#define DisFLGa     0x00
#define CY          0x01
#define BW          0x02
#define CMP         0x04
#define IDX         0x08
#define DisFLGb     0x00
#define SIGN        0x10
#define UPDN        0x20
#define EnCNTR      0x00
#define DisCNTR     0x40
#define Ltch_FLGa   0x00
#define Instnt_FLGa 0x80
```

```

void Init_7566(int Addr);
/*****
Initialize 7566

Set x0A, x0B for Non-Quadrature Mode
Free-running Mode
A = clk, B = up / dn

Configure            as the Reset Counter pin
FLGa - Carry / Borrow / Compare
FLGb disabled
Enable counting
FLGa - instantaneous

Set x1A, x1B for x1-Quadrature Mode
Free-running Mode

Configure x1-        INDX as the load_OL pin

Set for Synchronous        INDX input
Input Filter Clock Division Factor = 1
FLGa - Carry / Borrow / Compare / Index
FLGb disabled
Enable Counting
FLGa – latched

Set x2A, x2B for x4-Quadrature Mode
Modulo N Count Mode for N = 0x123456

Disable        INDX input
Input Filter Clock Division Factor = 2
FLGa - Carry / Borrow / Compare
FLGb disabled
Enable Counting
FLGa – latched
Disable x3A, x3B counting
*****/

```



```

void Init_7566(int Addr){

/* Set up for x0-axis */
outp(MDR0(Addr + axis_0), (NQDX + FreeRun + RCNTR + ASYN_IDX));
outp(MDR1(Addr + axis_0), (CY + BW + CMP + DisFLGb + EnCNTR + Instnt_FLGa));

/* Set up for x1-axis */
outp(MDR0(Addr + axis_1), (QDX1 + FreeRun + RCNTR + SYN_IDX + FDF1));
outp(MDR1(Addr + axis_1), (CY + BW + CMP + IDX + DisFLGb + EnCNTR + Ltch_FLGa));

/* Set up for x2-axis */
outp(MDR0(Addr + axis_2), (QDX4 + ModN + DisIDX + FDF2));
outp(MDR1(Addr + axis_2), DisCNTR);

outp(PR0(Addr + axis_2), 0x56);
outp(PR1(Addr + axis_2), 0x34);
outp(PR2(Addr + axis_2), 0x12);

outp(CMR(Addr + axis_2), Ld_CNTR);

outp(MDR1(Addr + axis_2), (CY + BW + CMP + DisFLGb + EnCNTR + Ltch_FLGa));

/* Set up for x3-axis */
outp(MDR1(Addr + axis_3), DisCNTR);
}

/* Write_7566_PR
Input: Addr has address of 7566 counter
Data has 24 bit data to be written to PR Register of axis-x0 */

void Write_7566_PR(int Addr, unsigned long Data);

void Write_7566_PR(int Addr, unsigned long Data){
outp(PR0(Addr + axis_0), Data);
Data >>= 8;
outp(PR1(Addr + axis_0), Data);
Data >>= 8;
outp(PR2(Addr + axis_0), Data);
}

```

```
/* Read_7566_OL
Input: Addr has address of 7566 counter
Output: Data returns 24 bit OL register value of axis-x0 */
```

```
unsigned long Read_7566_OL(int Addr);
```

```
unsigned long Read_7566_OL(int Addr){
unsigned long Data = 0;
outp(CMR(Addr + axis_0), Ld_OL);
Data |= (unsigned long) inp(OL0(Addr + axis_0));
lrotr(Data,8);
Data |= (unsigned long) inp(OL1(Addr + axis_0));
lrotr(Data,8);
Data |= (unsigned long) inp(OL2(Addr + axis_0));
lrotr(Data,16);
return(Data);
}
```

```
/* Read x0-axis Status Register */
```

```
unsigned char Get_7566_Status(int Addr);
```

```
unsigned char Get_7566_Status(int Addr){
return(inp(STR(Addr + axis_0)));
}
```

```
/* Read IMR and ISR */
```

```
unsigned char Get_7566_Interrupt_Status(int Addr);
```

```
unsigned char Get_7566_Interrupt_Status(int Addr){
return(inp(ISR_IMR(Addr)));
}
```