LSI/CSI

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

LS7267

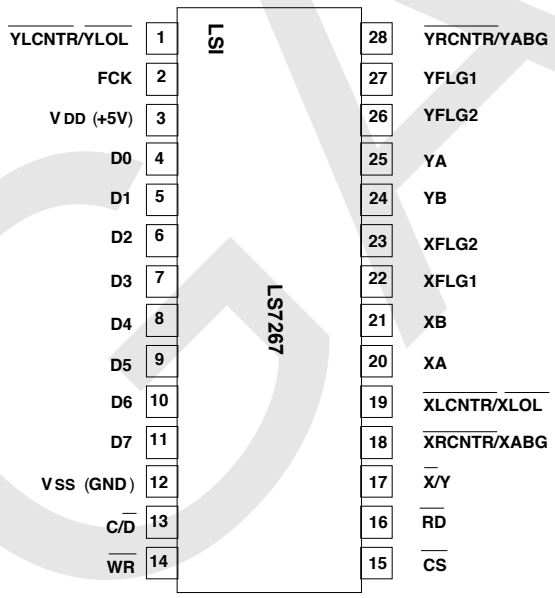
24-BIT DUAL-AXIS QUADRATURE COUNTER

NOV 2016

FEATURES:

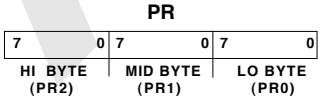
- Up to 50MHz count frequency in non-quadrature mode; Up to 5.6MHz clock frequency (22 x 10^6 counts/sec) in x4 quadrature mode.
- Dual 24-bit counters to support X and Y axes in motion control applications. • Dual 24-bit comparators.
- Digital filtering of the input quadrature clocks
- Programmable 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.
- 3V to 5.5V operation (V_{DD} - V_{SS}).
- TTL/CMOS compatible I/Os.
- **LS7267** (DIP); **LS7267-S** (SOIC); **LS7267-TS** (TSSOP)

PIN ASSIGNMENT
TOP VIEW



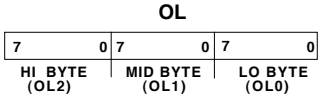
LS7267 Registers:
LS7267 has a set of registers associated with each X and Y axis. All X-axis registers have the name prefix X, whereas all Y-axis registers have the prefix Y. Selection of a specific register for Read/Write is made from the decode of the three most significant bits (D7 - D5) of the data-bus. $\overline{CS}$ 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 LS7267 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.



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.

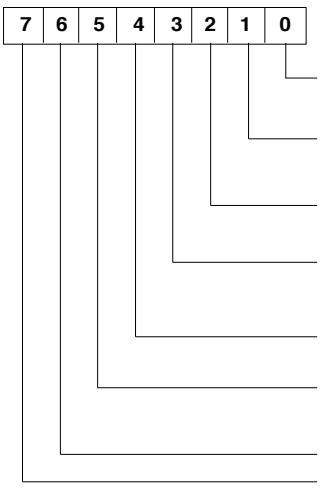


Byte Pointers: XBP and YBP
The Read and Write operations on an OL or a PR always accesses one byte at a time. The byte that is accessed is addressed by one of the BPs. At the end of every data Read or Write cycle on an OL or a PR, the associated BP is automatically incremented to address the next byte.

Flag Register: XFLAG and YFLAG

The FLAG registers hold the status information of the CNTRs and can be read on the data bus. All bits excepting the E and the IDX bits change dynamically to represent the instantaneous status of the CNTR's. In contrast the E and the IDX bits are latched. Once set they can only be reset via the RLD registers.

FLAG



BT: Borrow Toggle flip-flop.
Toggles every time CNTR underflows.

CT: Carry toggle flip-flop.
Toggles every time CNTR overflows.

CPT: Compare toggle flip-flop.
Toggles every time PR equals CNTR.

S: Sign flag. Set to 1 when CNTR underflows.
Reset to 0 when CNTR overflows.

E: Error latch. Set to 1 when parameter t4 or t5 is violated (see p.8 for t4 and t5)
Irrelevant in non-quadrature mode.

U/$\overline{D}$: Up/Down flag. Set to 1 when counting up
and reset to 0 when counting down.

IDX: Index. Set to 1 when designated index input transitions to active level.

0: Not used. Always reset to 0.

Filter Clock Prescalers: XPSC and YPSC

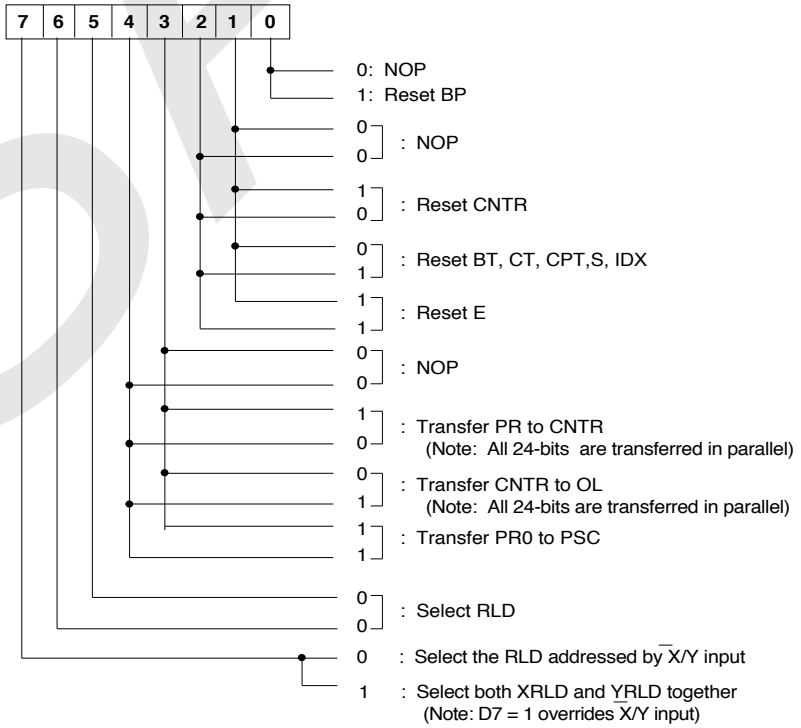
Each PSC is an 8-bit programmable modulo-n down counter, driven by the FCK input. The division factor n is stored into each PSC from its associated PR register low byte, PR0. The PSCs provide the ability to generate independent filter clock frequencies for each channel used for filtering the quadrature clocks applied at the A and B inputs in quadrature mode.The same filter clocks are also used for filtering the designated INDEX inputs.

The effective internal filter clock frequency is: $f_{FCKn} = (f_{FCK}/(n+1))$, where n = PSC = 0 to h'FF and f_{FCK} is the clock frequency at the FCK input. For proper operation the required condition is: $f_{FCKn} > 8f_{QA}$ (or $8f_{QB}$), where f_{QA} (or f_{QB}) is the frequency at A (or B) input. The FCK pin is not used in non-quadrature mode and should be tied off to V_{DD}.

Reset and Load Signal Decoders: XRLD and YRLD

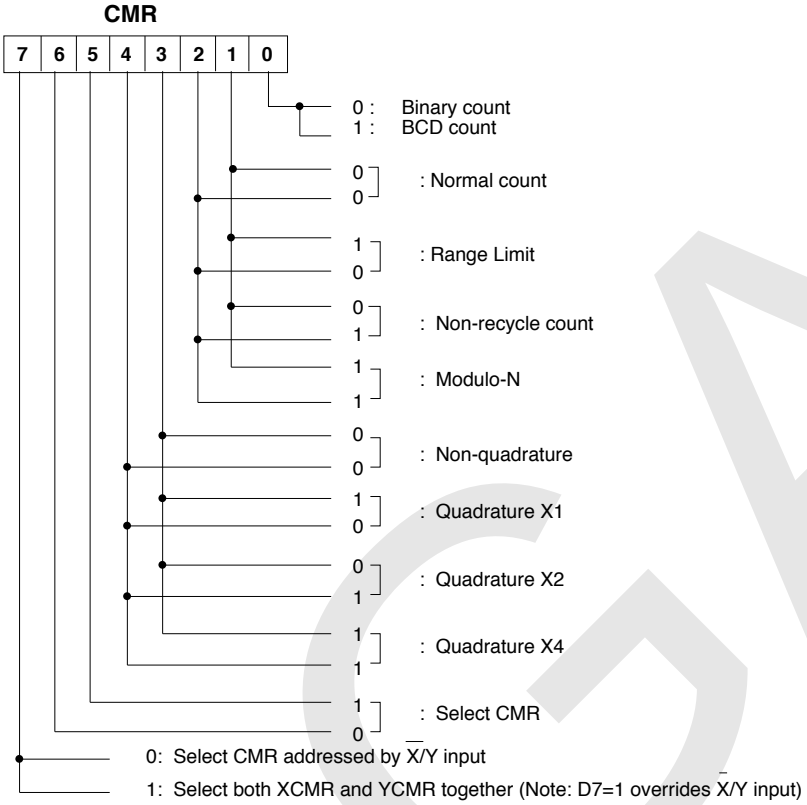
Following functions can be performed by writing a control byte into an RLD: Transfer PR to CNTR, Transfer CNTR to OL, reset CNTR, reset FLAG and reset BP.

RLD



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.

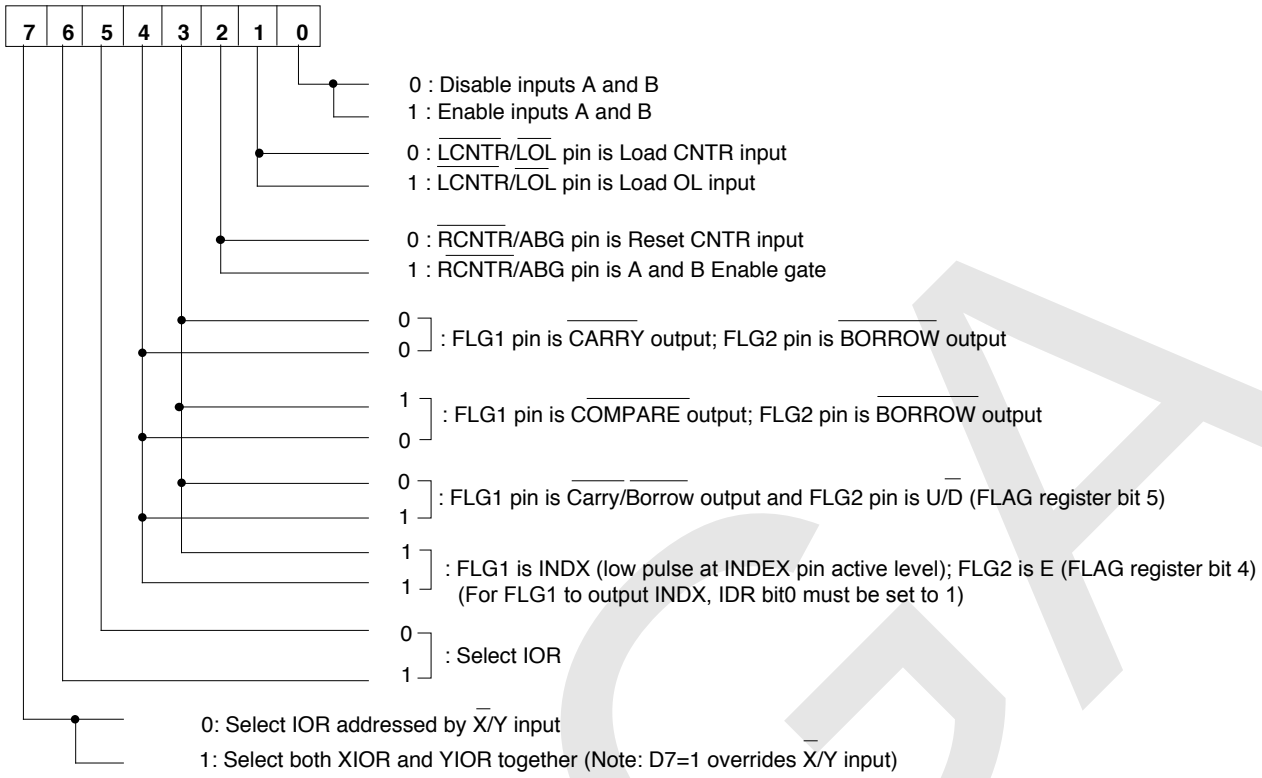
Modulo-N. In modulo-N count mode, a count boundary is set between 0 and the content of PR. When counting up, at CNTR = PR, the CNTR is reset to 0 and the up count is continued from that point. When counting down, at CNTR = 0, the CNTR is loaded with the content of PR and down count is continued from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

IOR

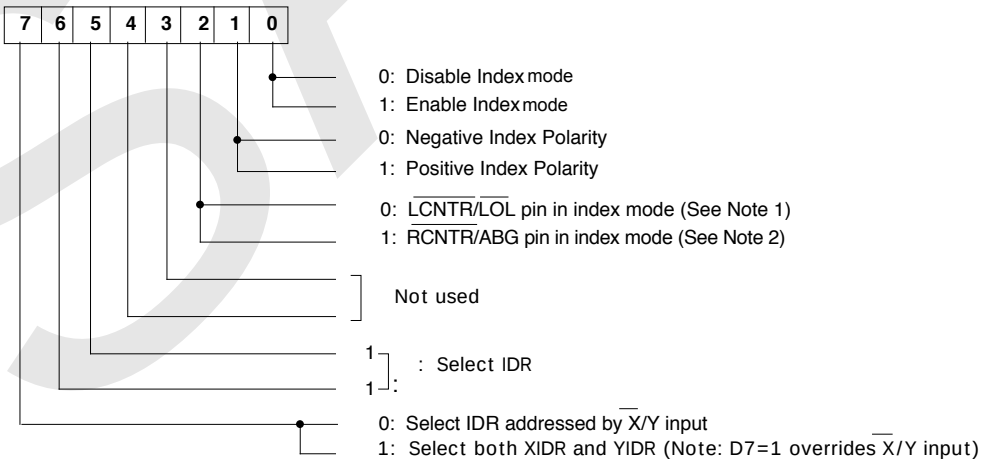


INDEX CONTROL REGISTERS: XIDR and YIDR

Either the LCNTR/LOL or the RCNTR/ABG inputs can be initialized to operate in the INDEX mode. In the INDEX mode the index signal from an encoder, applied to one of the two inputs performs either reset_CNTR, Load_CNTR or Load_OL operation **synchronously** with the quadrature clocks, the internal load or reset signal being generated at the same angular orientation of the encoder in both CW and CCW directions. In contrast in non-INDEX mode these signals are applied **asynchronously** with respect to A/B inputs and act instantaneously.

The index mode is unconditionally disabled in non-quadrature mode.

IDR



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 mode" causes the **synchronous** mode for the selected input (Pin 1, 18, 19 and 28) to be enabled. "Disable Index mode" causes the **asynchronous** mode for the selected pin to be enabled. The input, however, is not disabled in either configuration.

I/O pins:

Following is a description of all the input/output pins.

YLCNTR/YLOL/ (pin 1). Input. Programmable input in the y-axis which can be configured with the YIOR register to function as either LD_YCNTR or LD_YOL input. LD_YCNTR causes the YPR to be loaded into YCNTR when the input is at active level. LD_YOL causes YCNTR to be loaded into YOL when the input is at active level.

In quadrature mode the input can further be initialized with the YIDR register to function in either INDEX mode or non-INDEX mode. The INDEX mode allows for the direct interface with the INDEX output of an incremental encoder. In this mode the YLCNTR/YLOL/ input is sampled with the filter clock derived from the FCK input clock which is used for validating the quadrature clocks as well

In the INDEX mode the YCLNTR/YLOL/ input can be configured for either high or low active logic levels with the YIDR. In the non-INDEX mode the active level is not programmable and is unconditionally set to low active.

In the non-INDEX mode YCLNTR/YLOL/ input is not sampled with the filter clock and can be applied asynchronously with respect to YA and YB inputs. In non-quadrature mode the YCLNTR/YLOL/ input is unconditionally set to the non-INDEX mode.

FCK. (pin 2). Input. A clock applied at the FCK input is used for generating filter clocks for both X and Y axes. The filter clocks are used for validating the quadrature clocks (at XA, XB, YA and YB inputs), and the INDEX signal (at XLCNTR/XLOL/ or XRCNTR/ or YLCNTR/YLOL/ or YRCNTR in INDEX mode). The clock at the FCK input is divided down separately by XPSC and YPSC prescalers to generate the filter clock for each axis. The prescaler output frequency is given by: $f_{FCKn} = f_{FCK}/(n+1)$, where f_{FCK} is the frequency at the FCK input and n = [XPSC] or [YPSC].

For proper operation in the quadrature mode the following condition of frequencies must be satisfied: $f_{FCKn} \geq 8f_{QAB}$, where f_{QAB} is the clock frequency at XA or XB or YA or YB input..

In non-quadrature mode the filter clock is not used and the FCK input must be tied off to either VDD or VSS.

VDD (pin 3). Supply voltage positive rail. +3V to +5V.

D0 through **D7** (pins 4 through 11). Inputs/Outputs. The 8-bit databus D0 through D7 is a 3-state portal for the READ/WRITE operation in and of the device. The databus is common to both axes. During a read operation when both CS/ and RD/ inputs are low, the content of either the OL or the FLAG register of the selected axis is placed on databus. During a write operation the content of the databus is written into the selected register at the trailing edge of the WR/ pulse. When CS/ is high the databus is disabled and placed in the high impedance state.

VSS (pin 12). Supply voltage negative rail or GND.

D/C (pin 13). Input. This input selects between a control register or a data register for read/write operation according to Table 1. When low, a write operation causes the content of the databus to be written into the selected PR register. When high, a write causes the databus to be written into the selected control register. During a read operation, a low at the D/C input causes the content of the selected OL to be output on the databus while a high causes the content of the selected FLAG register to be output on the databus.

WR/ (pin 14). Input. A low pulse at the WR/ input causes the content of the databus to be written into the selected register according to Table 1. The write operation is completed at the trailing edge of the WR/ pulse.

CS/ (pin 15). Input. A low at the CS/ input enables the device for read or write operations. When the CS/ input is high the read and write operations are disabled and the databus, D0-D7 is placed in the high impedance state.

RD/ (pin 16). Input. A low at the RD/ input causes the content of the selected register to be output on the databus according to Table 1.

X/Y (pin 17). Input. Selects between X and Y axis for read and write. A low at this input selects the X axis while a high selects the Y axis.

XRCNTR/XABG (pin 18). Input. Programmable input in the x-axis which can be configured with the XIOR register to function as either R_XCNTR or E_XAB input. R_XCNTR causes the XCNTR to be reset to 0 when the input is in the active level. E_XAB causes the input to function as the XA and XB enable/disable gate. In this mode XA and XB become enabled when the XRCNTR/XABG input is high and disabled when the input is low,

In quadrature mode the input can further be initialized with the XIDR register to function in either INDEX mode or non-INDEX mode. In the INDEX mode the XRCNTR/XABG is sampled with the filter clock derived from the FCK input clock which is used for validating the quadrature clocks as well.

When configured in the INDEX mode the XRCNTR/XABG input must also be configured as R_XCNTR to function correctly. In the INDEX mode the XRCNTR/XABG input can be configured for either high or low active logic levels for the R_XCNTR function via the XIDR register. In the non-INDEX mode the active level is not programmable and is unconditionally set to low active for the R_XCNTR function.

In the non-INDEX mode XRCNTR/XABG input is not sampled with the filter clock and can be applied

asynchronously with respect to XA and XB inputs. In non-quadrature mode the XCNTR/XABG input is unconditionally set to the asynchronous mode

XLCNTR/XLOL/ (pin 19). Input. This is the x-axis functional equivalent of the YCLNTR/YLOL input. The associated x-axis reference registers are: XCNTR, XOL, XIOR and XIDR

XA (pin 20), **XB** (pin 21). Inputs. These are the A and B count inputs in the x-axis. These inputs can be configured to function either in quadrature mode or in non-quadrature mode. The configuration is made with the XCMR register. In quadrature mode, XA and XB clocks are normally supplied from incremental encoders with the two clocks being 90º out of phase. When XA leads XB in phase, the counter (XCNTR) counts up; when XA lags XB, the XCNTR counts down.

In non-quadrature mode XA functions as the count input and XB as the count direction control input. When XB is high positive transitions at the XA input causes the XCNTR to count up. Conversely, when XB is low positive transitions at the XA input causes the XCNTR to count down.

In quadrature mode XA and XB inputs are sampled for logic level validation and noise discrimination with an internal filter clock derived from the FCK input clock. In non-quadrature mode XA and XB inputs are not sampled and the count clocks are applied directly to the XCNTR count input bypassing the filter circuit.

XFLG1 (pin 22). Output. Programmable output in the x-axis to function as CARRY or BORROW or COMPARE or INDX. The configuration is made with the XIOR register. The output is generated as a low pulse when the selected event takes place CARRY or BORROW output is generated when the XCNTR overflows or underflows respectively. INDX output is generated when the designated INDEX input goes into active level. COMPARE output is generated at XCNTR = XPR. An exception is made in the **modulo-N** and **range-limit** modes In these modes the COMPARE output is generated at XCNTR = XPR only in the **up** count direction.

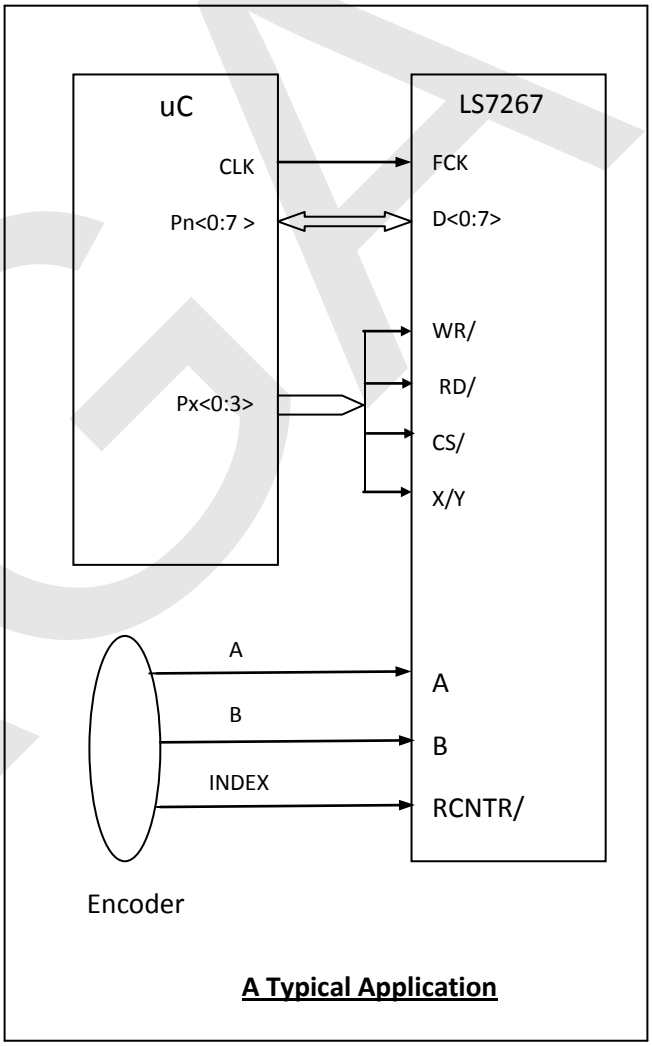
XFLG2 (pin 23). Output. Programmable output in the x-axis to function as BORROW or UP/DN or E. The configuration is made with the XIOR register. The BORROW output is generated as a low pulse when the XCNTR underflows. When configured as UP/DN the output dynamically goes high when the XCNTR is counting up and low when it is counting down. When configured as E, XFLG2 switches low whenever the following condition exists: $t_4 \leq t_3$ or $t_5 \leq 2t_3$ (see page 8). It remains low until the FLAG register bit-4 is reset whereupon XFLG2 output switches high.

YA (pin25), **YB** (pin 24). Inputs. These are the y-axis functional equivalents of the XA and XB inputs. They operate on the YCNTR and are configured with the YCMR register.

YFLG2 (pin26), Output. This is the y-axis functional equivalent of XFLG2 .

YFLG1 (pin27), Output. This is the y-axis functional equivalent of XFLG1

YRCNTR/YABG (pin 28). Input. This is the y-axis functional equivalent of the XRCNTR/XABG input. The associated y-axis reference registers are: YCNTR, YOL, YIOR and YIDR .



A Typical Application

REGISTER ADDRESSING MODES (Table 1)

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

Absolute Maximum Ratings:

Parameter	Symbol	Values	Unit
Supply Voltage	V_{DD}	+7.0	V
Voltage at any input	V_{IN}	V_{SS} - 0.3 to V_{DD} + 0.3	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} = 3V to 5.5V)

Parameter	Symbol	Min. Value	Max.Value	Unit	Remarks
Supply Voltage	V_{DD}	3.0	5.5	V	-
Supply Current	I_{DD}	-	800	μA	All clocks off
Input Logic Low	V_{IL}	-	0.15V_{DD}	V	-
Input Logic High	V_{IH}	0.5V_{DD}	-	V	-
Output Low Voltage	V_{OL}	-	0.5	V	I_{OSNK} = 5mA , V_{DD} = 5V
Output High Voltage	V_{OH}	V_{DD} - 0.5	-	V	I_{OSRC} = 1mA , V_{DD} = 5V
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, V_{DD} = 5V
Output Sink Current	I_{OSNK}	5.0	-	mA	V_O = 0.5V, V_{DD} = 5V

Transient Characteristics. (T_A = -25˚C to +80˚C)

Parameter	Symbol	Min. Value	Max.Value	Unit	Remarks
For V_{DD} = 3V to 5.5V:					
Read Cycle (See Fig. 1)					
$\overline{RD}$ Pulse Width	t_{r1}	80	-	ns	-
$\overline{CS}$ Set-up Time	t_{r2}	80	-	ns	-
$\overline{CS}$ Hold Time	t_{r3}	0	-	ns	-
C/$\overline{D}$ Set-up Time	t_{r4}	80	-	ns	-
C/$\overline{D}$ Hold Time	t_{r5}	10	-	ns	-
$\overline{X}$/Y Set-up Time	t_{r6}	80	-	ns	-
$\overline{X}$/Y Hold Time	t_{r7}	10	-	ns	-
Data Bus Access Time	t_{r8}	80	-	ns	Access starts when both $\overline{RD}$ and $\overline{CS}$ are low.
Data Bus Release Time	t_{r9}	-	35	ns	Release starts when either $\overline{RD}$ or $\overline{CS}$ is terminated.
Back to Back Read delay	t_{r10}	90	-	ns	-
Write Cycle (See Fig. 2)					
$\overline{WR}$ Pulse Width	t_{W1}	45	-	ns	-
$\overline{CS}$ Set-up Time	t_{W2}	45	-	ns	-
$\overline{CS}$ Hold Time	t_{W3}	0	-	ns	-
C/$\overline{D}$ Set-up Time	t_{W4}	45	-	ns	-
C/$\overline{D}$ Hold Time	t_{W5}	10	-	ns	-
$\overline{X}$/Y Set-up Time	t_{W6}	45	-	ns	-
$\overline{X}$/Y Hold Time	t_{W7}	10	-	ns	-
Data Bus Set-up Time	t_{W8}	45	-	ns	-
Data Bus Hold Time	t_{W9}	10	-	ns	-
Back to Back Write Delay	t_{W10}	90	-	ns	-
Load CNTR, Reset CNTR and Load OL Pulse Width	t_{11}	35	-	ns	-
For V_{DD} = 3V:					
Quadrature Mode (See Fig. 3-5)					
FCK High Pulse Width	t_1	18	-	ns	-
FCK Low Pulse Width	t_2	18	-	ns	-
FCK Frequency	f_{FCK}	-	27	MHz	-
Mod-n Filter Clock(FCKn)Period	t_3	36	-	ns	$t_3 = (n+1)(t_1+t_2)$, where n = PSC = 0 to FF_H
FCKn frequency	f_{FCKn}	-	27	MHz	-
Quadrature Separation	t_4	40	-	ns	$t_4 > t_3$
Quadrature Clock Pulse Width	t_5	160	-	ns	$t_5 > 4t_3$
Quadrature Clock frequency	f_{QA}, f_{QB}	-	3.1	MHz	$f_{QA} = f_{QB} = 1/(2t_5)$
Quadrature Clock to Count Delay	t_{Q1}	$4t_3$	5t3	-	-
x1/x2/x4 Count Clock Pulse Width	t_{Q2}	36	-	ns	$t_{Q2} = t_3$
Index Input Pulse Width	t_{idx}	76	-	ns	$t_{idx} > 2t3$
Index setup/hold time	tsi/thi	5	-	ns	-
INDX Output Width	tQ3	18	-	ns	tQ3 = (t3 - t2)
Carry/Borrow/Compare Output Width	t_{Q3}	18	-	ns	$t_{Q3} = (t_3 - t_2)$
Non-Quadrature Mode (See Fig. 6-7)					
Clock A - High Pulse Width	t_6	18	-	ns	-
Clock A - Low Pulse Width	t_7	18	-	ns	-
Direction Input B Set-up Time	t_{bS}	20	-	ns	-
Direction Input B Hold Time	tbh	10	-	ns	-
Gate Input (ABG) Set-up Time	t_{GS}	20	-	ns	-
Gate Input (ABG) Hold Time	t_{GH}	10	-	ns	-
Clock Frequency	f_A	-	27	MHz	$f_A = (1/(t_6 + t_7))$
Clock to Carry or Borrow Out Delay	t_9	-	20	ns	-
Carry or Borrow Out Pulse Width	t_{10}	18	-	ns	$t_{10} = t_7$
Clock to Compare Out Delay	t_{12}	-	30	ns	-

Parameter	Symbol	Min. Value	Max.Value	Unit	Remarks
For V_{DD} = 5V:					
Quadrature Mode (See Fig. 3-5)					
FCK High Pulse Width	t_1	10	-	ns	-
FCK Low Pulse Width	t_2	10	-	ns	-
FCK Frequency	f_{FCK}	-	50	MHz	-
Mod-n Filter Clock(FCKn)Period	t_3	20	-	ns	$t_3 = (n+1)(t_1+t_2)$, where n = PSC = 0 to FF$_H$
FCKn frequency	f_{FCKn}	-	50	MHz	-
Quadrature Separation	t_4	22	-	ns	$t_4 > t_3$
Quadrature Clock Pulse Width	t_5	88	-	ns	$t_5 > 4t_3$
Quadrature Clock frequency	f_{QA}, f_{QB}	-	5.6	MHz	$f_{QA} = f_{QB} = 1/(2t_5)$
Quadrature Clock to Count Delay	t_{Q1}	$4t_3$	$5t_3$	-	-
x1/x2/x4 Count Clock Pulse Width	t_{Q2}	20	-	ns	$t_{Q2} = t_3$
Index Input Pulse Width	t_{idx}	42	-	ns	$t_{idx} > 2t3$
Index setup/hold time	tsi/thi	5	-	ns	-
INDX output width	tQ3	10	-	ns	tQ3 = (t3 - t2)
Carry/Borrow/Compare Output Width	t_{Q3}	10	-	ns	$t_{Q3} = (t_3 - t_2)$
Non-Quadrature Mode (See Fig. 6-7)					
Clock A - High Pulse Width	t_6	10	-	ns	-
Clock A - Low Pulse Width	t_7	10	-	ns	-
Direction Input B Set-up Time	tbs	12	-	ns	-
Direction Input B Hold Time	tbh	5	-	ns	-
Gate Input (ABG) Set-up Time	t_{GS}	12	-	ns	-
Gate Input (ABG) Hold Time	t_{GH}	5	-	ns	-
Clock Frequency	f_A	-	50	MHz	$f_A = (1/(t_6 + t_7))$
Clock to Carry or Borrow Out Delay	t_9	-	10	ns	-
Carry or Borrow Out Pulse Width	t_{10}	10	-	ns	$t_{10} = t_7$
Clock to Compare Out Delay	t_{12}	-	15	ns	-

7267-091615-9

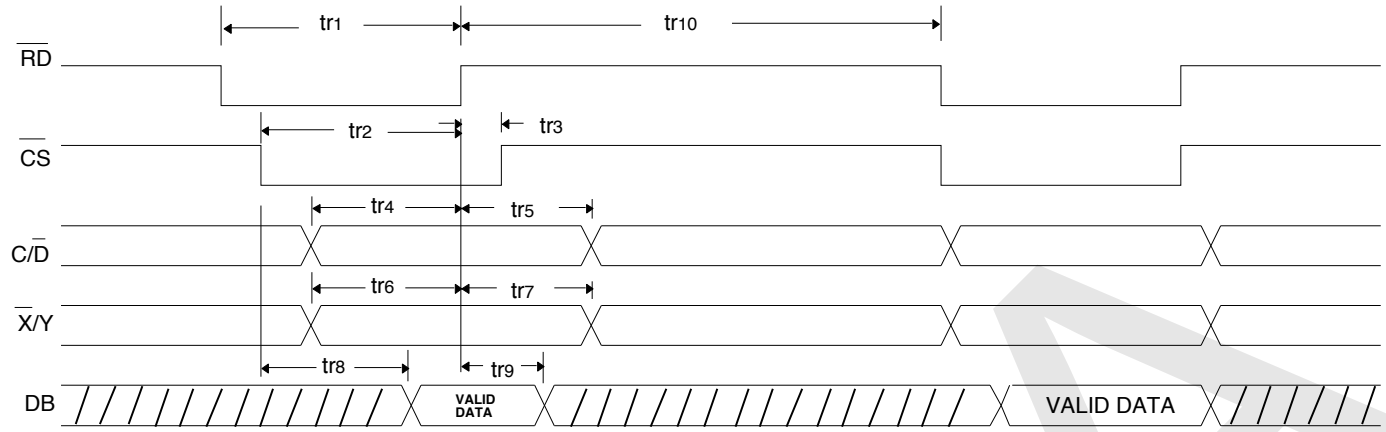


FIGURE 1. READ CYCLE

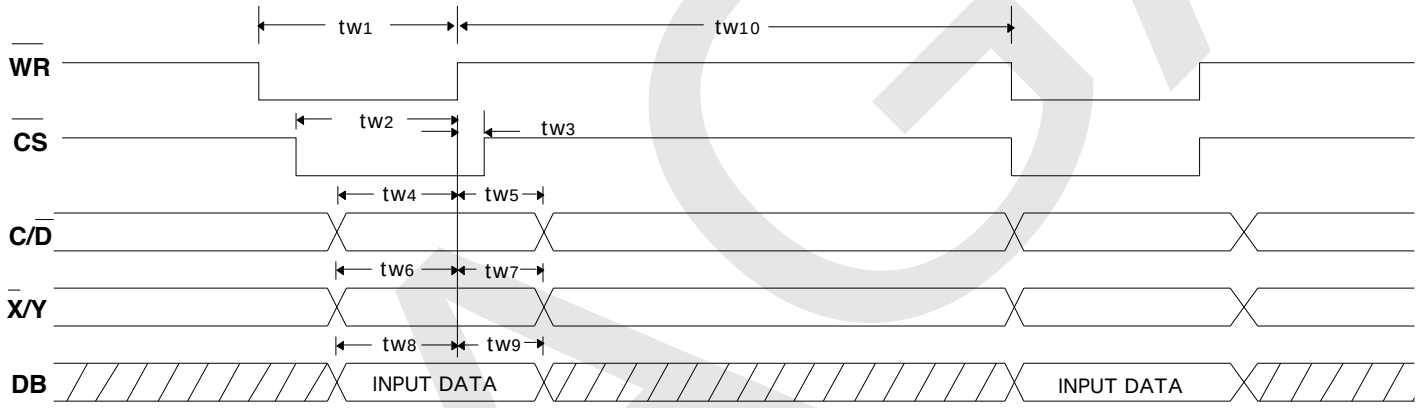


FIGURE 2. WRITE CYCLE

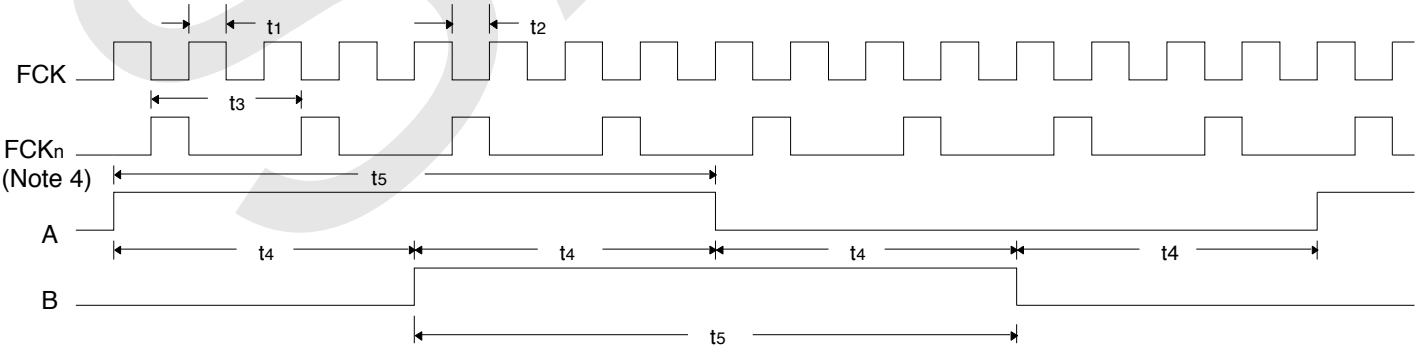


FIGURE 3. FILTER CLOCK FCK AND QUADRATURE CLOCKS A AND B

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

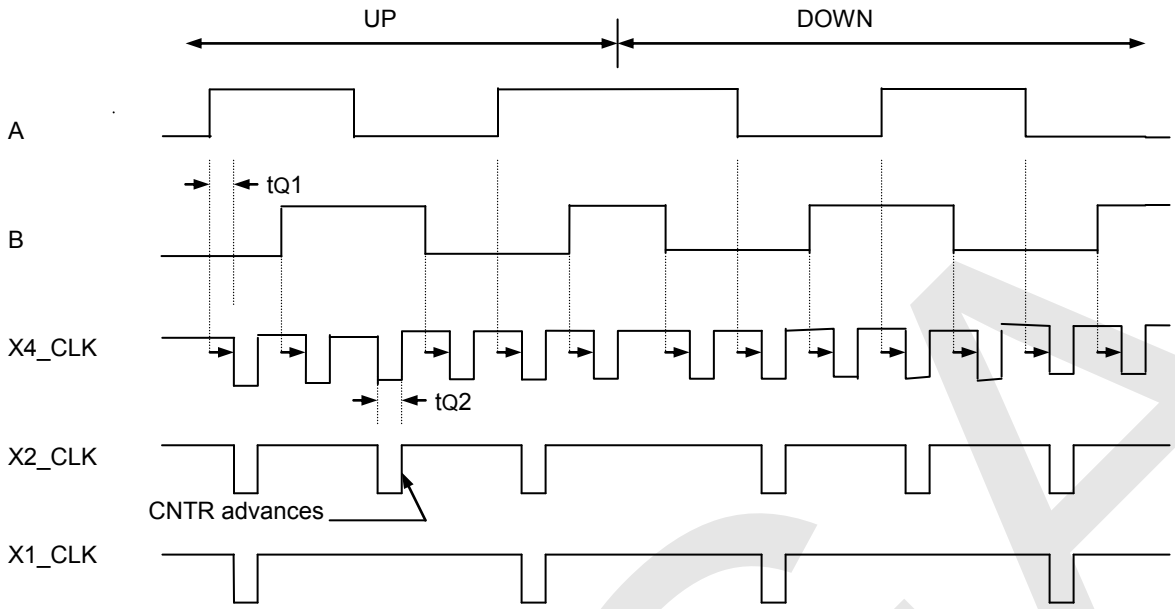
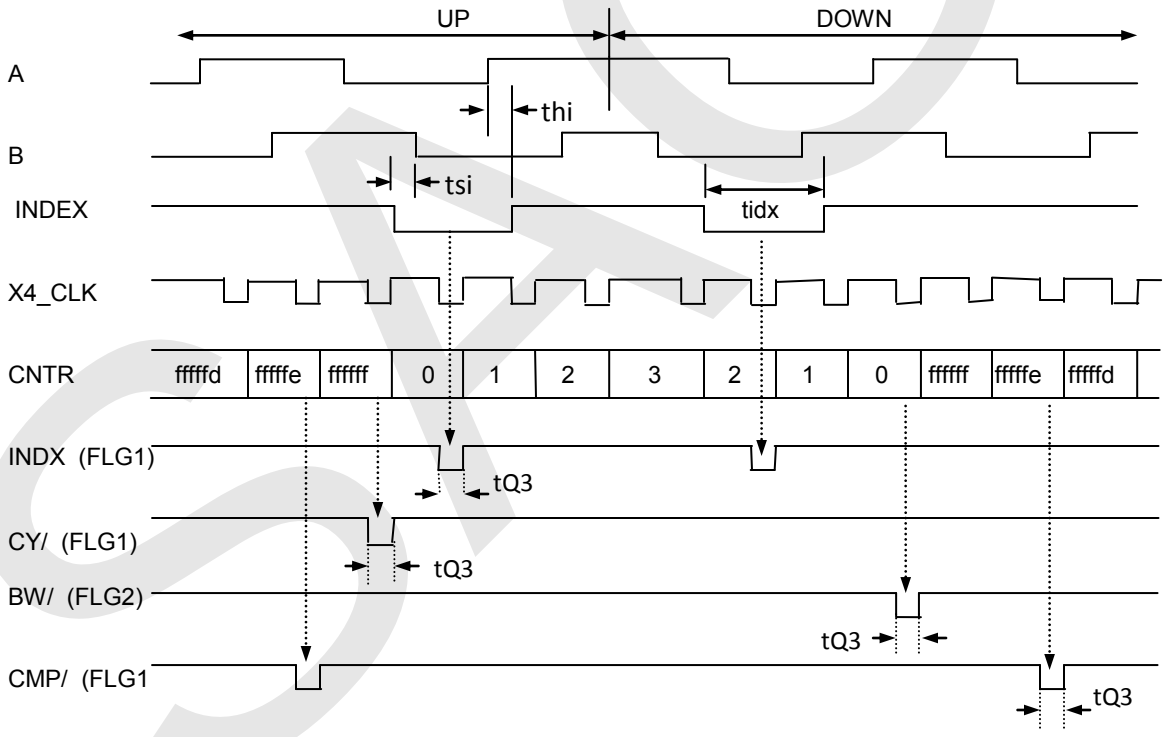


Fig 4. A/B quadrature clocks and internal count clocks in X1, X2 and X4 modes



Note. CMP/ signal is arbitrarily shown to be generated at CNTR = fffffe
Note. In modulo-N and range-limit modes the CMP/ output is generated in up count only
Note. INDEX signal must overlap quarter cycle of both A and B high or both A and B low. Shown here both low.

Fig 5. FLG1 and FLG2 outputs in quadrature count mode

7267-052815-11

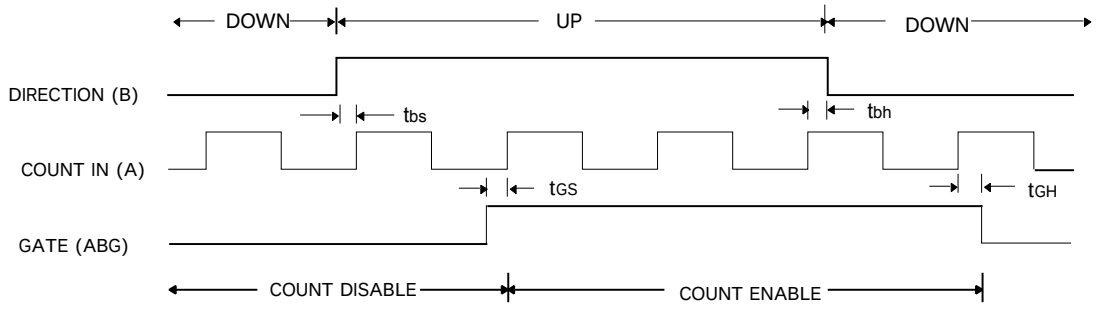


FIGURE 6. COUNT (A), DIRECTION (B) AND GATE (ABG) INPUTS IN NON-QUADRATURE MODE

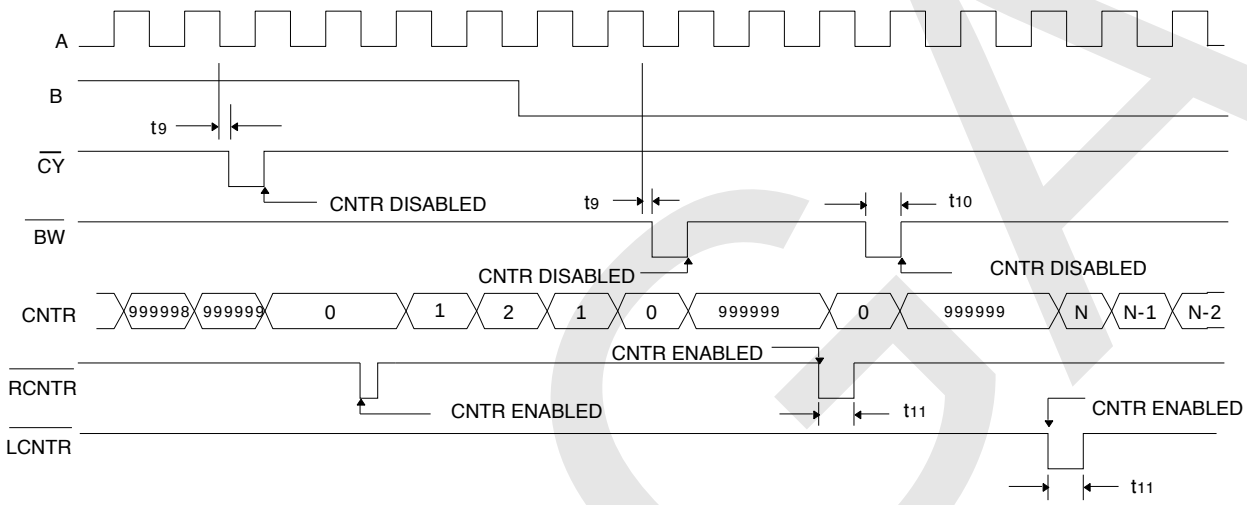


FIGURE 7. NON-RECYCLE, NON-QUADRATURE, BCD MODE

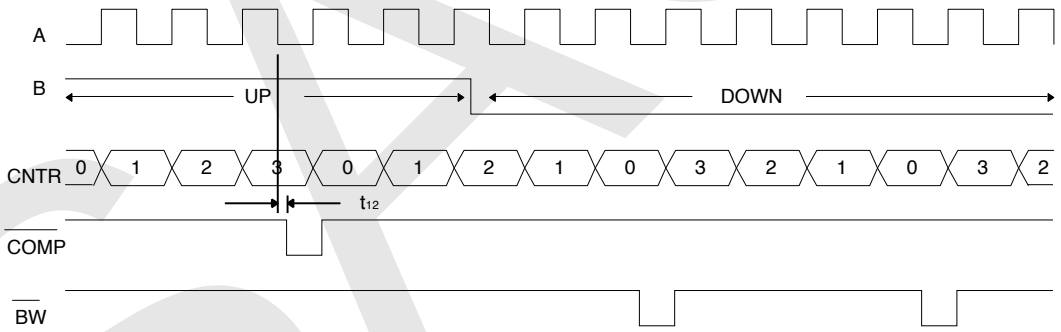


FIGURE 8. MODULO - N, NON-QUADRATURE (Shown with N = 3)

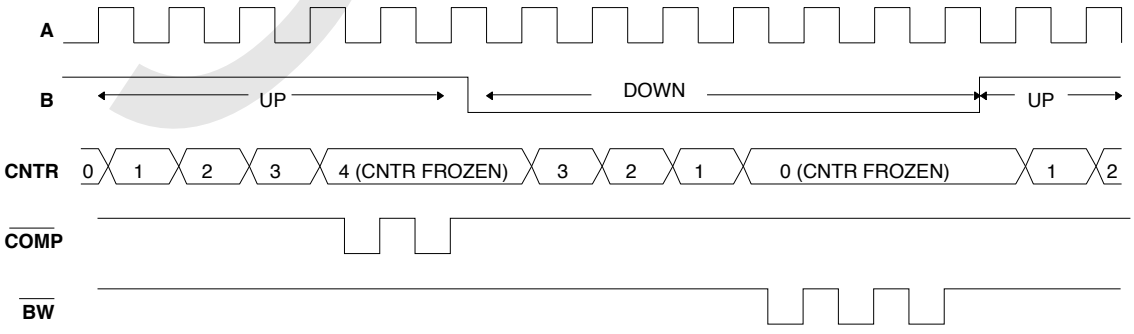


FIGURE 9. RANGE LIMIT, NON-QUADRATURE (Shown with PR = 4)

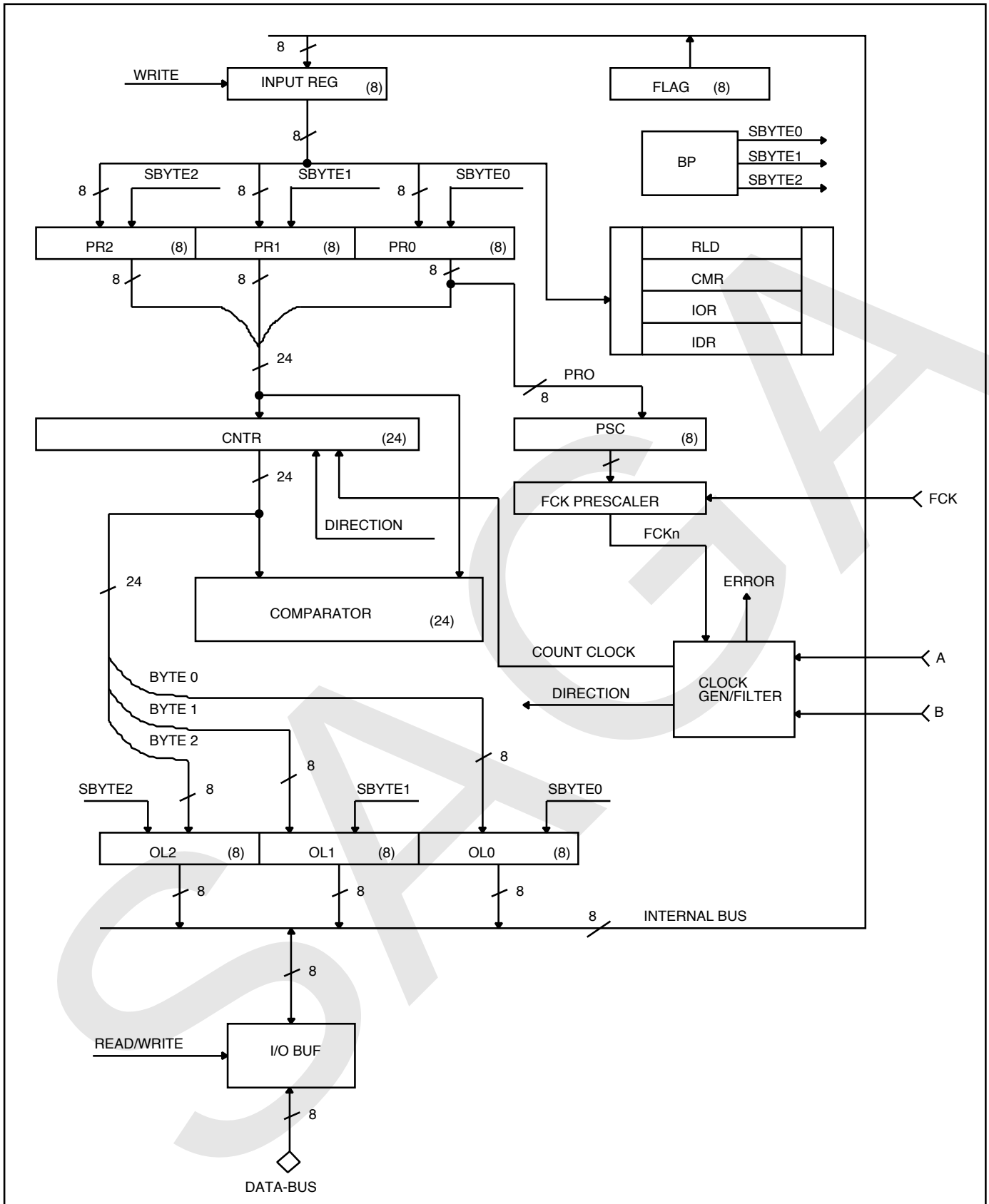


FIGURE 10. SINGLE-AXIS BLOCK DIAGRAM SHOWING MAJOR DATA PATHS

77267-042713-13

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 (X + Y CNTR)
     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(YDATA(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
   //Setup PR Reg. for Modulo N Counter to 0x123456
    outp(YDATA(Addr),0x56);    //Least significant Byte first
    outp(YDATA(Addr),0x34);    //then middle byte
    outp(YDATA(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(CMD(Addr)));
}
```