

Universidad de Costa Rica

Tarea 3

Verificación funcional de componentes

Prof. Javier Pacheco

José Ricardo Soro Jara, B36853

Grupo 1

17 de mayo de 2020

Índice

1. Objetivos	3
2. Definición de funcionamiento	4
2.1. Entradas	4
2.1.1. Clock	4
2.1.2. Enable	4
2.1.3. Modo	4
2.1.4. D	4
2.2. Salidas	5
2.2.1. Q	5
2.2.2. rco	5
3. Ambiente de Testing	6
3.1. Plan de pruebas	6
3.2. Definitions.v	6
4. Resultados	7
4.1. Salida RCO en Contador A	7
4.2. Salida RCO en Contador B	8
4.3. Salida RCO en Contador C	8
4.4. Resumen	9
5. Conclusiones	10
6. Anexos	11
6.1. Especificaciones	11
6.2. Hoja de Fabricante	13
6.3. Resultados Contador A	19
6.4. Resultados Contador B	23
6.5. Resultados Contador C	27

1. Objetivos

- Verificar el funcionamiento de los componentes suministrados basandose en el enunciado de la tarea (Anexo 1) como especificación de funcionamiento.
- Utilizar los resultados de la simulación para determinar el componente que cumpla mejor las especificaciones.

2. Definición de funcionamiento

Tomando como base el enunciado dado; se encuentra que el contador tiene cuatro entradas que determinan el comportamiento del componente, osea sus dos salidas. Es necesario definir correctamente lo que se entendió de la especificación dada para poder justificar las conclusiones emitidas al final del documento.

2.1. Entradas

En orden jerárquico de efecto sobre las salidas se tienen las entradas

2.1.1. Clock

La entrada de clock es un pulso cuadrado constante y uniforme que determina el tiempo de operación y los momentos de ejecución del componente.

2.1.2. Enable

Una entrada binaria que determina si el componente opera o no; en caso de operar debe estar en alto (1) en cuyo caso opera según la entrada de modo. En caso de estar en bajo (0) el componente no opera y las salidas automáticamente pasan a ser 0.

2.1.3. Modo

Este componente tiene cuatro modos de operación descritos a continuación

- Suma 1 (00)
- Resta 1 (01)
- Resta 3 (10)
- Load (11)

De esta forma el modo de sumar 1 debe sumar de uno en uno hasta sobrepasar el valor máximo del contador (15) debe volver a 0. El modo de restar 1 funciona de manera similar pero restando y al rebasar el valor de 0 debe regresar al valor máximo de 15. El modo de restar 3 funciona igual al anterior pero restando de tres en tres y al sobre pasar se debe restar el sobrepaso al valor maximo de 15. En el modo de Load se copia el valor en D

2.1.4. D

. Esta entrada solo se utiliza cuando se tiene enabled en alto y el modo es 11; en este caso se copia este valor como la salida Q.

2.2. Salidas

2.2.1. Q

Esta salida representa el valor del contador y está en constante cambio

2.2.2. rco

Esta salida representa el evento de rebase y es la que presenta mayor problema para definir su comportamiento pues las especificaciones dadas no son suficientes para determinar como debe comportarse; no obstante se debe definir un comportamiento esperado para poder realizar un proceso de verificación por lo cual se definió que esta señal debe mantenerse en 0 hasta que exista una condición de rebase que significa que la señal pasa a 1 por un ciclo de reloj y luego regresa a 0. En esta señal surgió la duda de que debe hacer la señal en el modo de carga (Load); para solucionar esto se recurrió a tomar como ejemplo contadores de 4 bits comerciales por lo cual se consultó la hoja de fabricante de SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163, SN74ALS161B, SN74ALS163B, SN74AS161 y SN74AS163 (Texas instruments) y se determinó que el comportamiento esperado de manera comercial para la señal de rebase es mantenerse en 0 durante la carga como se muestra en la siguiente figura

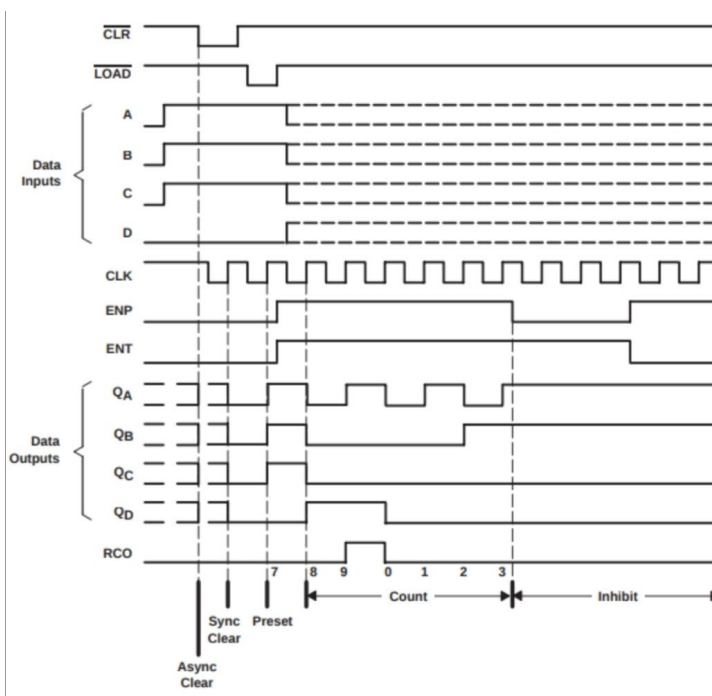


Figura 1: Señales en contador de 4 bits comercial. Elaborado por Texas Instruments

En donde se puede ver que cuando se activa el modo de carga el rco se mantiene en 0 por lo cual se tomará este como comportamiento esperado para realizar pruebas. No obstante se debe aclarar que debido a la falta de información disponible en la especificación dada, resulta imposible emitir un criterio absoluto sobre los contadores proporcionados.

3. Ambiente de Testing

Para el desarrollo de la verificación de estos componentes se utilizó el siguiente esquema de modulo para este desarrollo

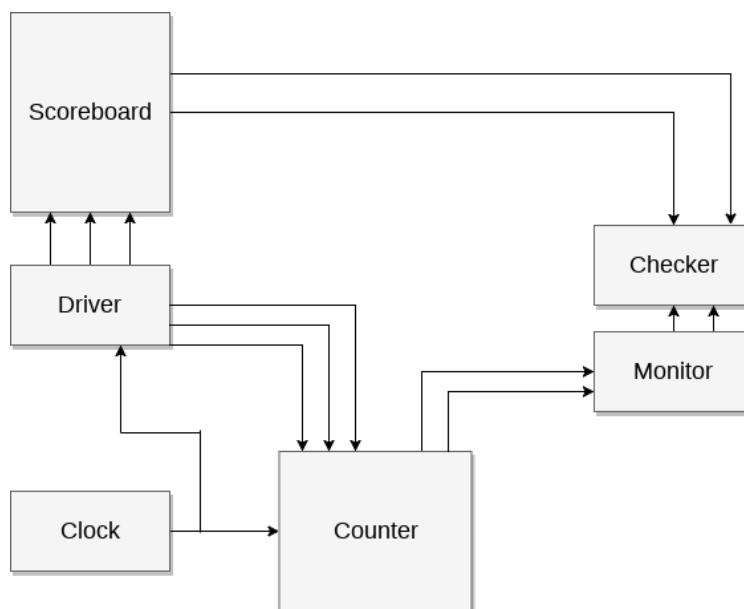


Figura 2: Ambiente de testing. Elaboración propia

Este ambiente de desarrollo se replicó tres veces para realizar pruebas de manera independiente sobre cada uno de los contadores y generar datos de estudio para cada uno. De manera adicional se desarrolló un cuarto ambiente donde se instancian los cuatro contadores; esto con fines meramente demostrativos.

3.1. Plan de pruebas

Como plan de pruebas se realizarán tres etapas; una etapa semi aleatoria en la que se colocan valores aleatorios en enable para evaluar la robustez del componente en al variar esta entrada. De manera seguida se inicia una segunda etapa completamente aleatoria en la que se utilizan variables aleatorias en las tres entradas distintas de Clock para evaluar la robustez en cambios abruptos en el modo y la cuenta desde cualquier valor. Por último se ejecuta la etapa completamente predefinida en la que se evalúan casos de interés para evaluar posibles casos esquina en los que podría fallar el componente. Además resulta importante resaltar que la semilla de números aleatorios se utilizó la predeterminada del sistema en caso de querer recrear resultados.

3.2. Definitions.v

También resulta importante la necesidad que hubo de generar un archivo de definición de cables que se importa dentro del modulo de tb_top ya que el compilador era incapaz de inferir los tamaños de los cables les asignaba wire de 1 bit y eso hacía imposible la compilación.

4. Resultados

Al ejecutar el plan de pruebas; se obtuvieron documentos que describen exactamente en que puntos se dieron errores de funcionamiento según la especificación definida en puntos anteriores. De esta forma se han adjuntado como anexo estos documentos al final de este documento para revisión exhaustiva de resultados y en esta sección se adjuntaran las imagenes de simulación para describir las situaciones de error y utilizar esta información para definir si el componente cumple o no estas especificaciones.

Si bien es necesario notar que la salida Q no presenta ningún problema en ningún contador como se demuestra a continuación; se debe advertir que la salida de rebase rco presenta problemas en cada contador



Figura 3: Comparación exitosa de salida Q. Elaboración propia

4.1. Salida RCO en Contador A

El plan de pruebas se ejecutó a la perfección en la primera etapa semi aleatoria demostrando robustez en cambios en la señal de enable; y mostró bastante estabilidad en la tercera etapa de pruebas predefinidas donde solo apareció un error en la señal de RCO. Para entender este error fue necesario utilizar los resultados de la prueba aleatoria donde se encontró que la señal de rco se dispara siempre que el modo es 11; no obstante al siguiente ciclo de reloj vuelve a 0; pero si se mantiene en modo 11 la señal vuelve a subir a 1 y se mantiene oscilando de esa manera cuando el comportamiento esperado es que RCO se mantenga en 0 cuando se está trabajando en modo 11. En la imagen adjunta se muestra el espacio de tiempo donde se dan estos errores en la prueba aleatoria

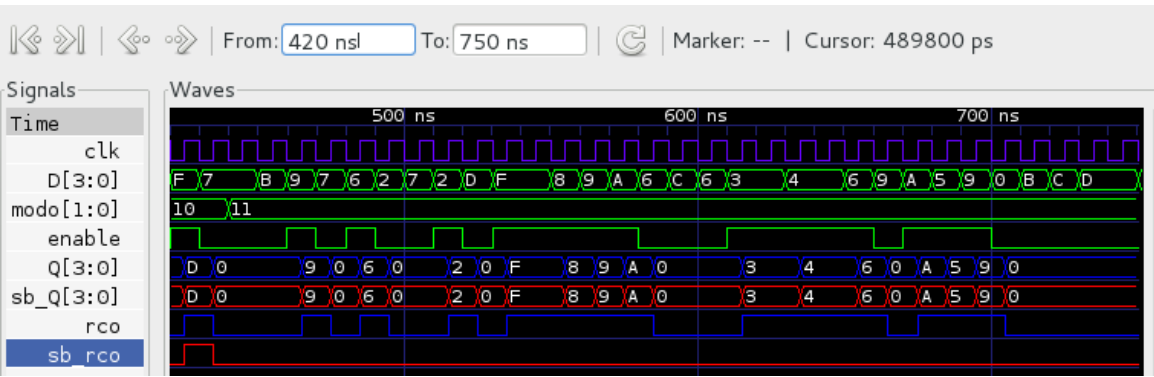


Figura 4: Funcionamiento incorrecto del contador A en modo 11. Elaboración propia

De este modo se determina que el contador A no cumple la especificación utilizada porque presenta errores en la señal RCO al estar en modo de carga (modo = 11)

4.2. Salida RCO en Contador B

El caso del contador B resultó ser el que tuvo peor desempeño pues no llegó a pasar ninguna de las etapas si presentar errores en RCO, por el contrario presentó muchos errores durante toda la ejecución de las pruebas; esto debido a que la lógica de RCO parece estar de alguna forma ligada a los cambios en modo; pues cada vez que hay un cambio en modo parece ocurrir una especie de toggle lo que hace imposible utilizar esta señal de manera adecuada ya que la información brindada por esta termina siendo prácticamente aleatoria.

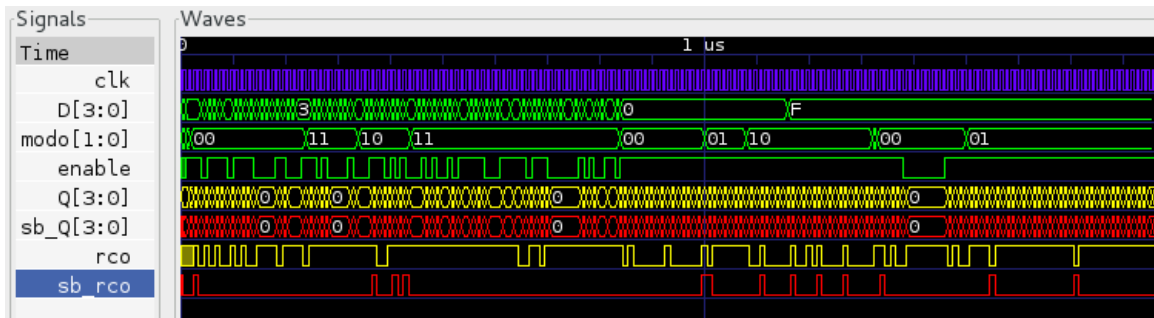


Figura 5: Funcionamiento incorrecto del contador B durante toda la prueba. Elaboración propia

Realizando un acercamiento a los valores se puede apreciar el efecto de toggle en la señal RCO además de otros comportamientos inesperados que fueron imposibles de relacionar a alguna otra señal.

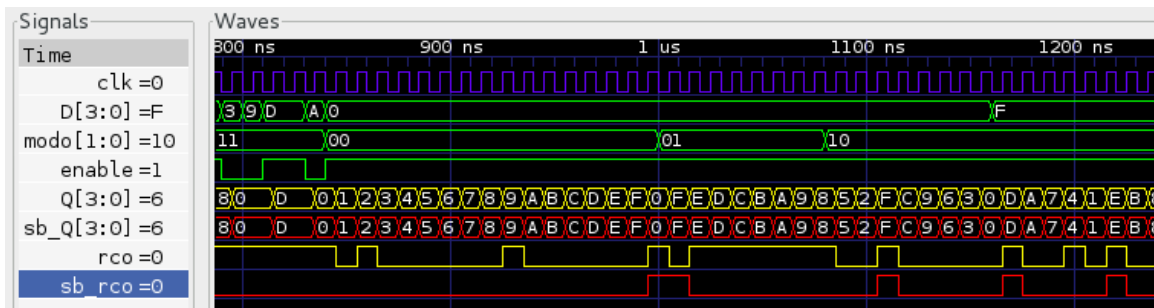


Figura 6: Comportamiento inadecuado de contador B. Elaboración propia

De esta forma se determina que la señal RCO no funciona en ningún caso de uso.

4.3. Salida RCO en Contador C

En el caso del contador C se encontró que este pasó las etapas de prueba aleatoria y semi aleatoria exitosamente; no obstante en el caso de las pruebas predefinidas se encontraron dos errores de funcionamiento en la señal RCO ambos ocurridos bajo las mismas condiciones. Este error ocurre cuando se opera en modo 10 y la señal Q se encuentra en un valor de 2 (0010); al pasar un ciclo de reloj se daría un rebase pues la señal Q queda en F (1111) no obstante en este caso específico el contador C no levante la señal de rebase como se muestra a continuación.

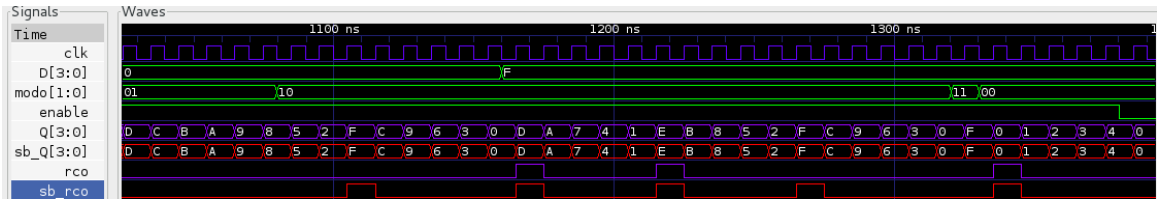


Figura 7: Funcionamiento incorrecto del contador C en modo 10. Elaboración propia

De esta forma se determina que esta señal de RCO no funciona en modo 10 bajo condiciones especiales ($Q=0010$).

4.4. Resumen

A manera de resumen se muestra la siguiente tabla de resultados

Señal	Funcionamiento	Condiciones
QA	Si	NA
QB	Si	NA
QC	Si	NA
rcoA	No	modo = 11
rcoB	No	NO FUNCIONA
rcoC	No	modo = 10; Q = 0010

Además de un resumen de la verificación

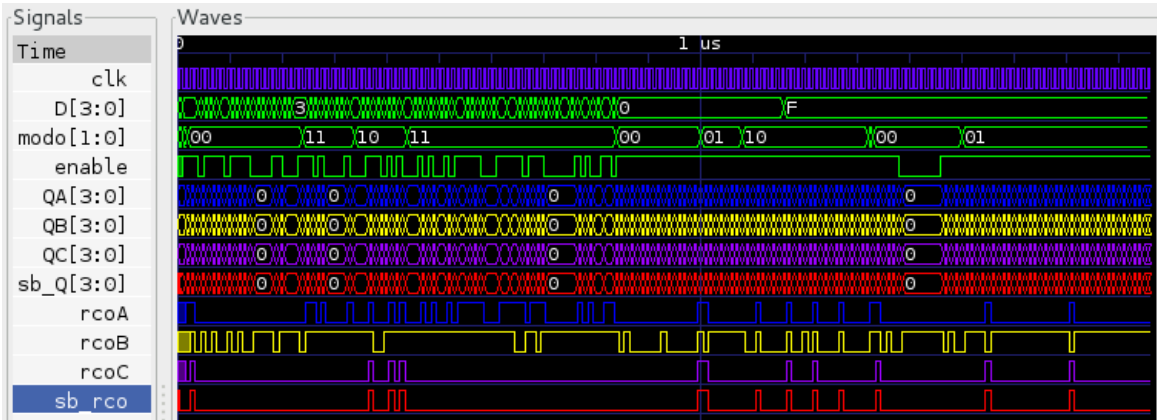


Figura 8: Resumen del proceso de verificación. Elaboración propia

5. Conclusiones

La primera conclusión a la que se llega es que no es posible dar un resultado confiable de la verificación debido a la falta de información proporcionada sobre el comportamiento esperado. No obstante dadas las condiciones de operación definidas previamente se debe decir que ningún contador aprobó el proceso de verificación.

Además se debe decir que el contador con peores resultados fue el Contador B. Por otra parte a pesar de que la cantidad de apariciones de errores en el contador A fue mayor que en el contador C; en mi opinión es más grave el error presentado en C pues este error hace que el contador no sea confiable en el modo 10 pues se perderían datos importantes de rebase. Por otra parte si el contador A si bien presenta un error importante en el modo 11; este error podría ser menos grave pues al ser un modo de carga el dato de rebase se vuelve irrelevante durante una carga y dado el comportamiento presentado en que este falso alto regresa a 0 en el siguiente ciclo indiferentemente del modo de operación este error no afectaría por ejemplo una conexión cascada de contadores para crear contadores de mayor cantidad de bits.

De esta manera se determina que ningún contador aprobó la verificación; pero el que parece ser más utilizable en aplicaciones reales tal y como está es el contador A.

6. Anexos

6.1. Especificaciones

IE0411 – Microelectrónica
I – 2020

Tarea Verificación

7 de mayo de 2020

Consideraciones generales:

- La entrega de esta tarea consiste en un informe de resultados, los cuales deben ser debidamente justificados.
- Toda imagen con resultados debe tener una buena calidad para ser calificada.
- Se deben entregar todos los ejecutables que comprueben el análisis de resultados.
- Como parte de la solución debe incluir un Makefile con su respectivo README que contenga la forma de ejecutar las pruebas.
- Se castigará severamente cualquier intento o asomo de copia durante el examen o en las soluciones.

Especificaciones

En la página del curso se encuentra una carpeta con tres archivos para la tarea 3. Cada archivo es un contador sincrónico de 4 bits, con los siguientes modos de funcionamiento:

- Cuenta hacia arriba.
- Cuenta hacia abajo.
- Cuenta de tres en tres hacia abajo.
- Carga en paralelo.

El objetivo de la tarea es realizar la verificación funcional de los tres contadores y garantizar el correcto funcionamiento de los diseños. Como parte de la verificación debe asegurarse de poder encontrar todos los errores del diseño (si existieran). Además, elaborar un informe marcando en cuales casos el diseño no cumple con la especificación; Asimismo, identificar en que casos el diseño sí cumple con la especificación.

El informe debe explicar la creación de todos los escenarios de prueba que como verificador contruyó. Se aconseja tener escenarios con valores aleatorios en las entradas y otros escenarios con pruebas específicas predefinidas por cada estudiante.

Especificación de las entradas del contador

1. **CLK**: Entrada de reloj del contador. El flanco activo de la señal CLK es el flanco creciente. Entonces, con cada flanco positivo del reloj el contador cambia de estado dependiendo del estado de las señales de MODO y si la señal ENB = 1.
2. **ENB**: Entrada de habilitación del contador. Si ENB = 1, el contador funciona normalmente respondiendo a los flancos activos de CLK para cambiar de estado de acuerdo a la señal MODO. Si ENB=0, el contador mantiene su estado actual sin importar los flancos de CLK.
3. **D[3:0]**: Entrada de datos D consta de 4 líneas. El valor que tengan las entradas D[3:0] será almacenado en Q[3:0] en el flanco activo de CLK si ENB = 1 y MODO = 11.
4. **MODO[1:0]**: Entrada de modo que consta de dos líneas y sirve para definir cuál será el próximo estado del contador al llegar el flanco activo del reloj en la entrada CLK. Si el contador se encuentra en el estado Q antes del flanco activo del reloj, luego del flanco activo, su estado será:
 - MODO = 00 – $\rightarrow Q + 1$
 - MODO = 01 – $\rightarrow Q - 1$
 - MODO = 10 – $\rightarrow Q - 3$
 - MODO = 11 – $\rightarrow D$

Especificación de las salidas del contador

1. **Q[3:0]** : Salida Q que consta de cuatro líneas que indican el estado presente del contador. El estado del contador cambia con el flanco activo de la señal CLK mientras ENB=1 y de acuerdo con el modo seleccionado con las líneas MODO[1:0].
2. **RCO** : Salida de llevo “Ripple-Carry Out” que indica cuando el contador llega a su cuenta límite para que la siguiente etapa, en contadores de más de 4 bits, se habilite para que realice su actualización de estado. Note que RCO se pone en 1 dependiendo de la señal MODO.

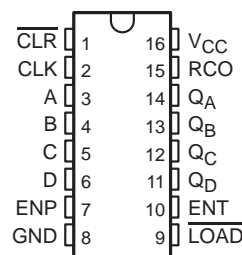
6.2. Hoja de Fabricante

SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163 SN74ALS161B, SN74ALS163B, SN74AS161, SN74AS163 SYNCHRONOUS 4-BIT DECADE AND BINARY COUNTERS

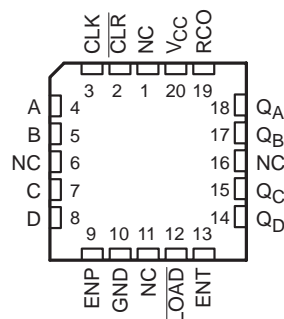
SDAS276 – DECEMBER 1994

- Internal Look-Ahead Circuitry for Fast Counting
- Carry Output for n-Bit Cascading
- Synchronous Counting
- Synchronously Programmable
- Package Options Include Plastic Small-Outline (D) Packages, Ceramic Chip Carriers (FK), and Standard Plastic (N) and Ceramic (J) 300-mil DIPs

SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163 ... J PACKAGE
SN74ALS161B, SN74ALS163B, SN74AS161, SN74AS163 ... D OR N PACKAGE
(TOP VIEW)



SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163 ... FK PACKAGE
(TOP VIEW)



NC – No internal connection

description

These synchronous, presettable, 4-bit decade and binary counters feature an internal carry look-ahead circuitry for application in high-speed counting designs. The SN54ALS162B is a 4-bit decade counter. The 'ALS161B, 'ALS163B, 'AS161, and 'AS163 are 4-bit binary counters. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincidentally with each other when instructed by the count-enable (ENP, ENT) inputs and internal gating. This mode of operation eliminates the output counting spikes normally associated with asynchronous (ripple-clock) counters. A buffered clock (CLK) input triggers the four flip-flops on the rising (positive-going) edge of the clock input waveform.

These counters are fully programmable; they may be preset to any number between 0 and 9 or 15. Because presetting is synchronous, setting up a low level at the load (LOAD) input disables the counter and causes the outputs to agree with the setup data after the next clock pulse, regardless of the levels of the enable inputs.

The clear function for the 'ALS161B and 'AS161 is asynchronous. A low level at the clear ($\overline{\text{CLR}}$) input sets all four of the flip-flop outputs low, regardless of the levels of the CLK, $\overline{\text{LOAD}}$, or enable inputs. The clear function for the SN54ALS162B, 'ALS163B, and 'AS163 is synchronous, and a low level at $\overline{\text{CLR}}$ sets all four of the flip-flop outputs low after the next clock pulse, regardless of the levels of the enable inputs. This synchronous clear allows the count length to be modified easily by decoding the Q outputs for the maximum count desired. The active-low output of the gate used for decoding is connected to $\overline{\text{CLR}}$ to synchronously clear the counter to 0000 (LLLL).

The carry look-ahead circuitry provides for cascading counters for n-bit synchronous applications without additional gating. ENP and ENT inputs and a ripple-carry (RCO) output are instrumental in accomplishing this function. Both ENP and ENT must be high to count, and ENT is fed forward to enable RCO. RCO, thus enabled, produces a high-level pulse while the count is maximum (9 or 15 with Q_A high). The high-level overflow ripple-carry pulse can be used to enable successive cascaded stages. Transitions at ENP or ENT are allowed, regardless of the level of CLK.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1994, Texas Instruments Incorporated

SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163 SN74ALS161B, SN74ALS163B, SN74AS161, SN74AS163 SYNCHRONOUS 4-BIT DECADE AND BINARY COUNTERS

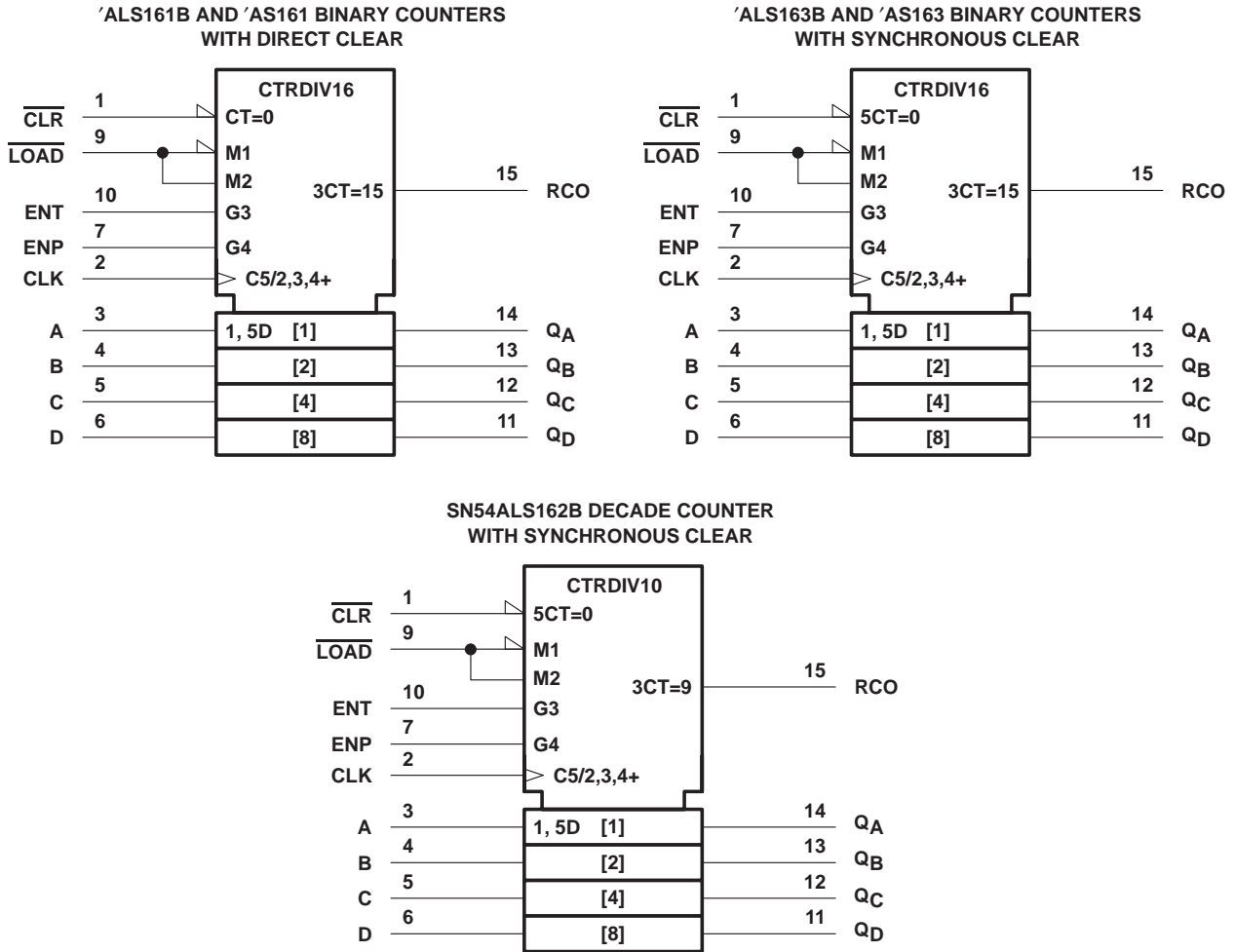
SDAS276 – DECEMBER 1994

description (continued)

These counters feature a fully independent clock circuit. Changes at control inputs (ENP, ENT, or $\overline{\text{LOAD}}$) that modify the operating mode have no effect on the contents of the counter until clocking occurs. The function of the counter (whether enabled, disabled, loading, or counting) is dictated solely by the conditions meeting the stable setup and hold times.

The SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, and SN54AS163 are characterized for operation over the full military temperature range of -55°C to 125°C . The SN74ALS161B, SN74ALS163B, SN74AS161, and SN74AS163 are characterized for operation from 0°C to 70°C .

logic symbols†

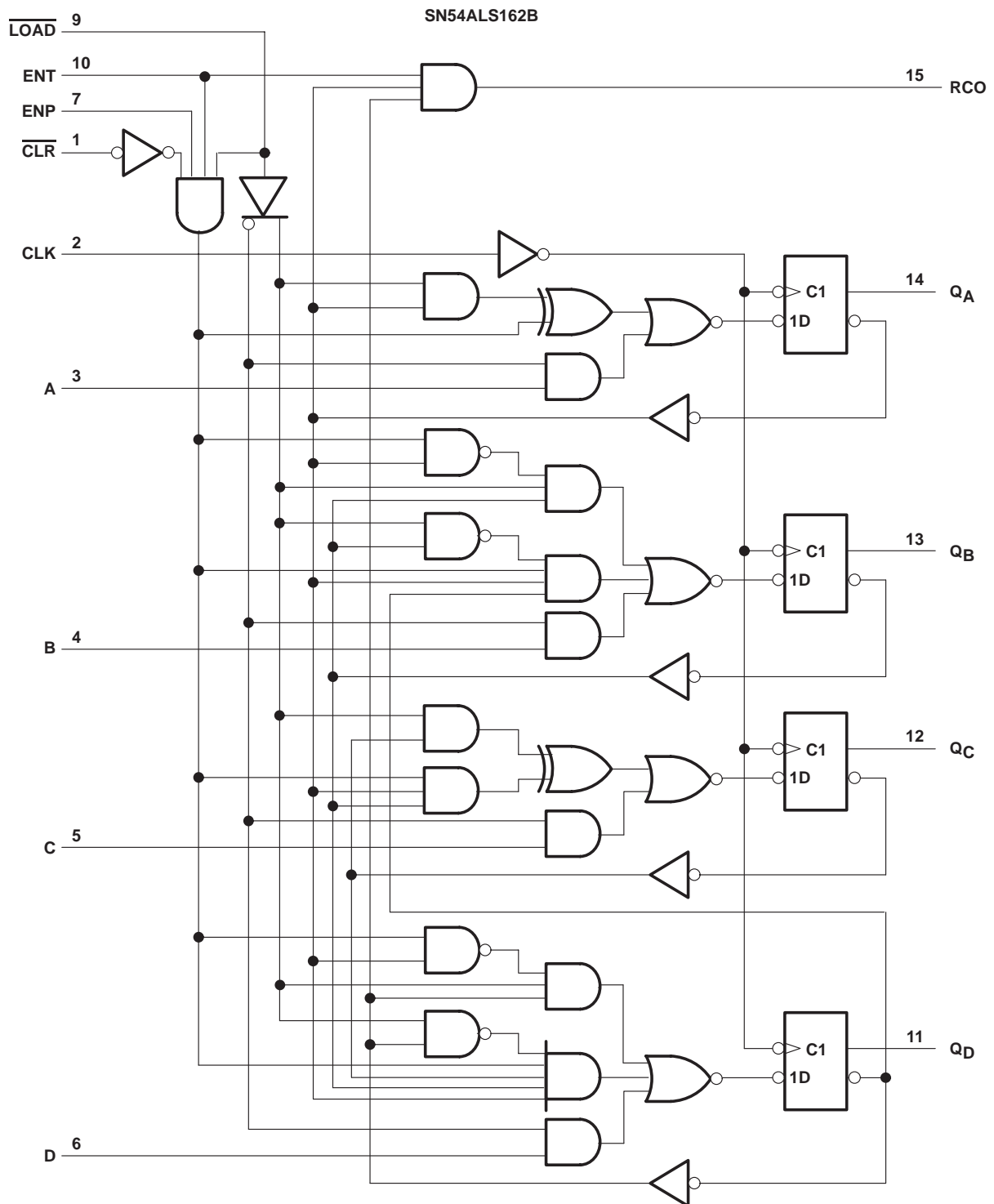


† These symbols are in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12. Pin numbers shown are for the D, J, and N packages.

SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163
 SN74ALS161B, SN74ALS163B, SN74AS161, SN74AS163
 SYNCHRONOUS 4-BIT DECADE AND BINARY COUNTERS

SDAS276 – DECEMBER 1994

logic diagram (positive logic)

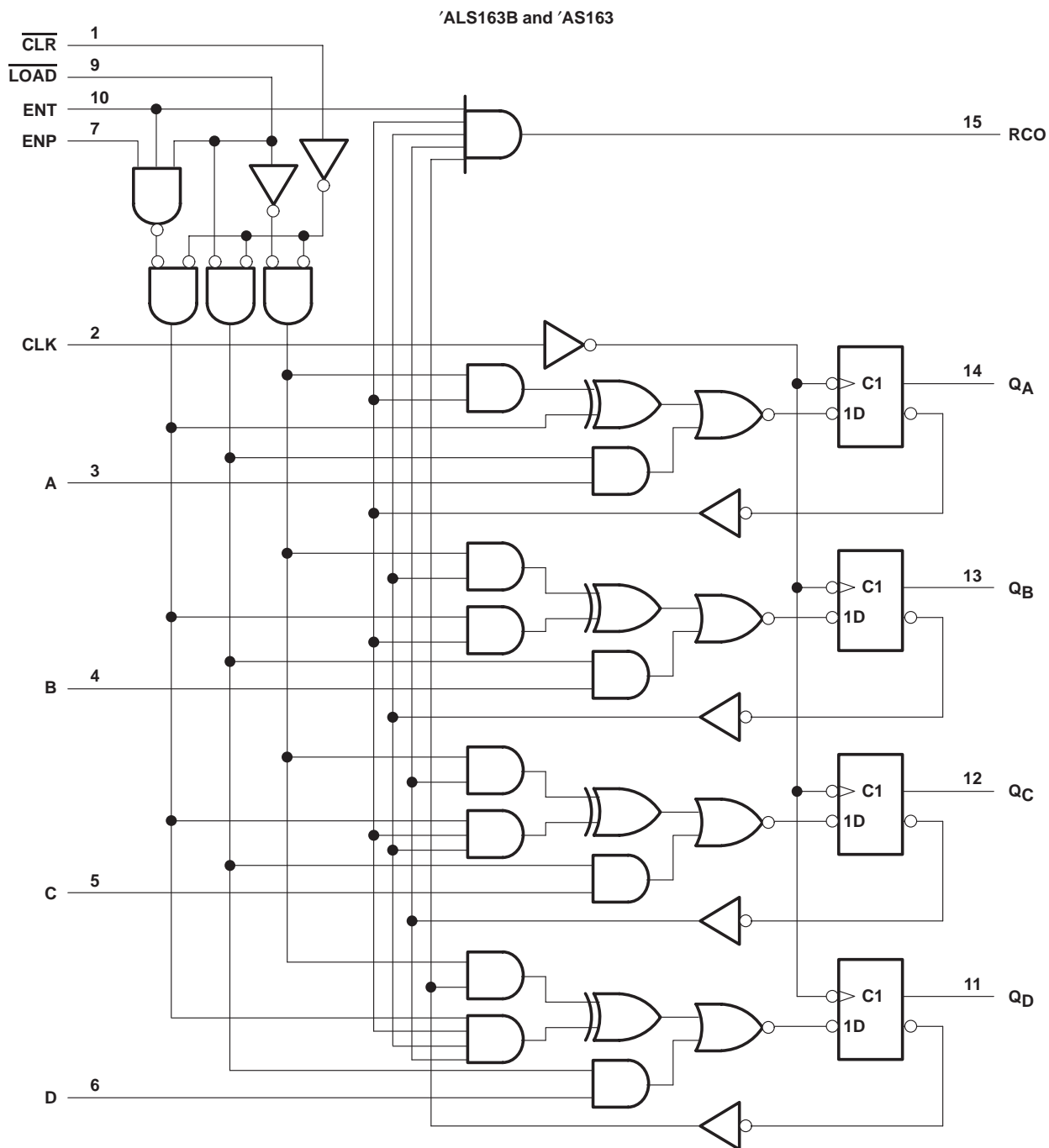


Pin numbers shown are for the J package.

SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163
 SN74ALS161B, SN74ALS163B, SN74AS161, SN74AS163
 SYNCHRONOUS 4-BIT DECADE AND BINARY COUNTERS

SDAS276 – DECEMBER 1994

logic diagram (positive logic)



Pin numbers shown are for the D, J, and N packages.

'ALS161B and 'AS161 synchronous binary counters are similar; however, $\overline{\text{CLR}}$ is asynchronous.

SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163
SN74ALS161B, SN74ALS163B, SN74AS161, SN74AS163
SYNCHRONOUS 4-BIT DECADE AND BINARY COUNTERS

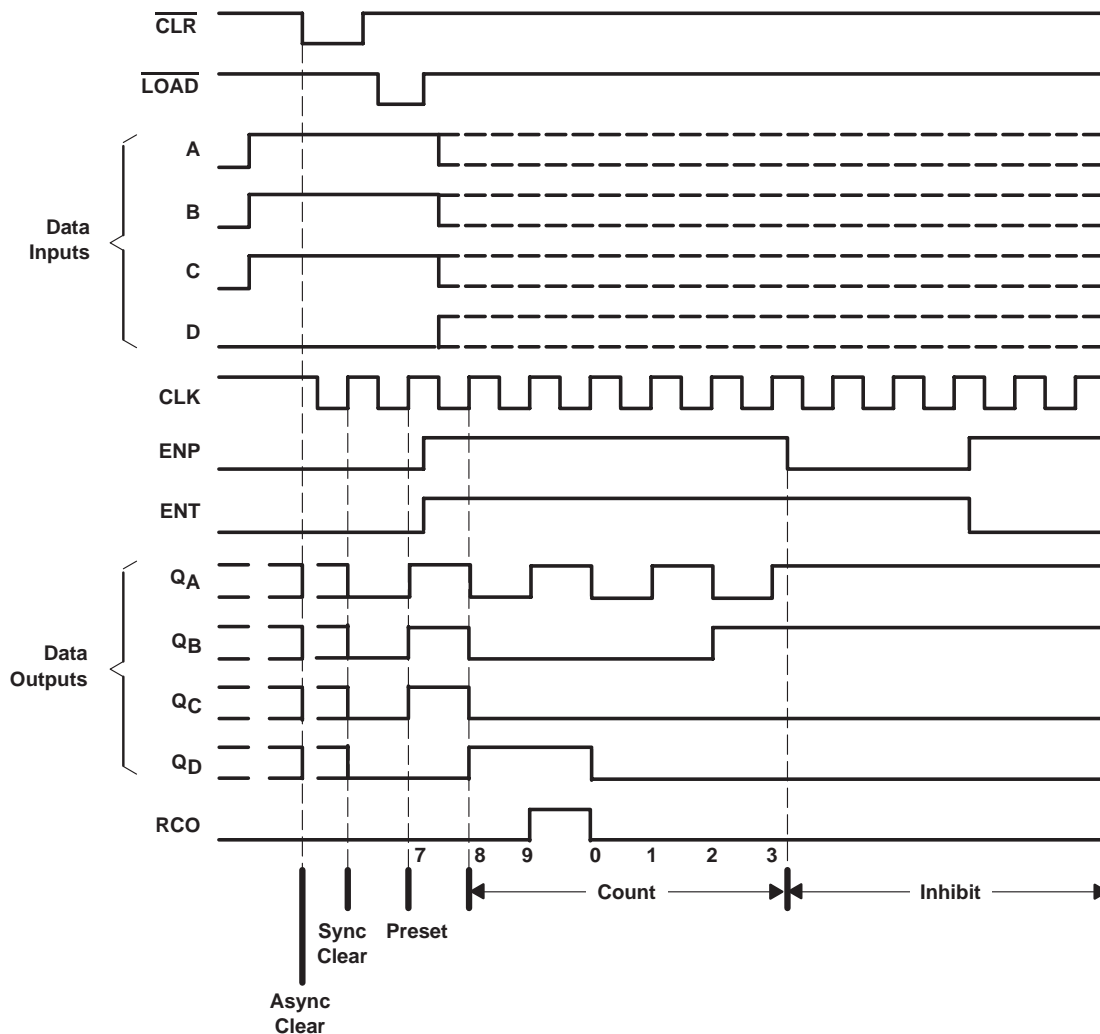
SDAS276 – DECEMBER 1994

typical clear, preset, count, and inhibit sequences

SN54ALS162B

The following sequence is illustrated below:

1. Clear outputs to zero (SN54ALS162B is synchronous)
2. Preset to BCD 7
3. Count to 8, 9, 0, 1, 2, and 3
4. Inhibit



SN54ALS161B, SN54ALS162B, SN54ALS163B, SN54AS161, SN54AS163 SN74ALS161B, SN74ALS163B, SN74AS161, SN74AS163 SYNCHRONOUS 4-BIT DECADE AND BINARY COUNTERS

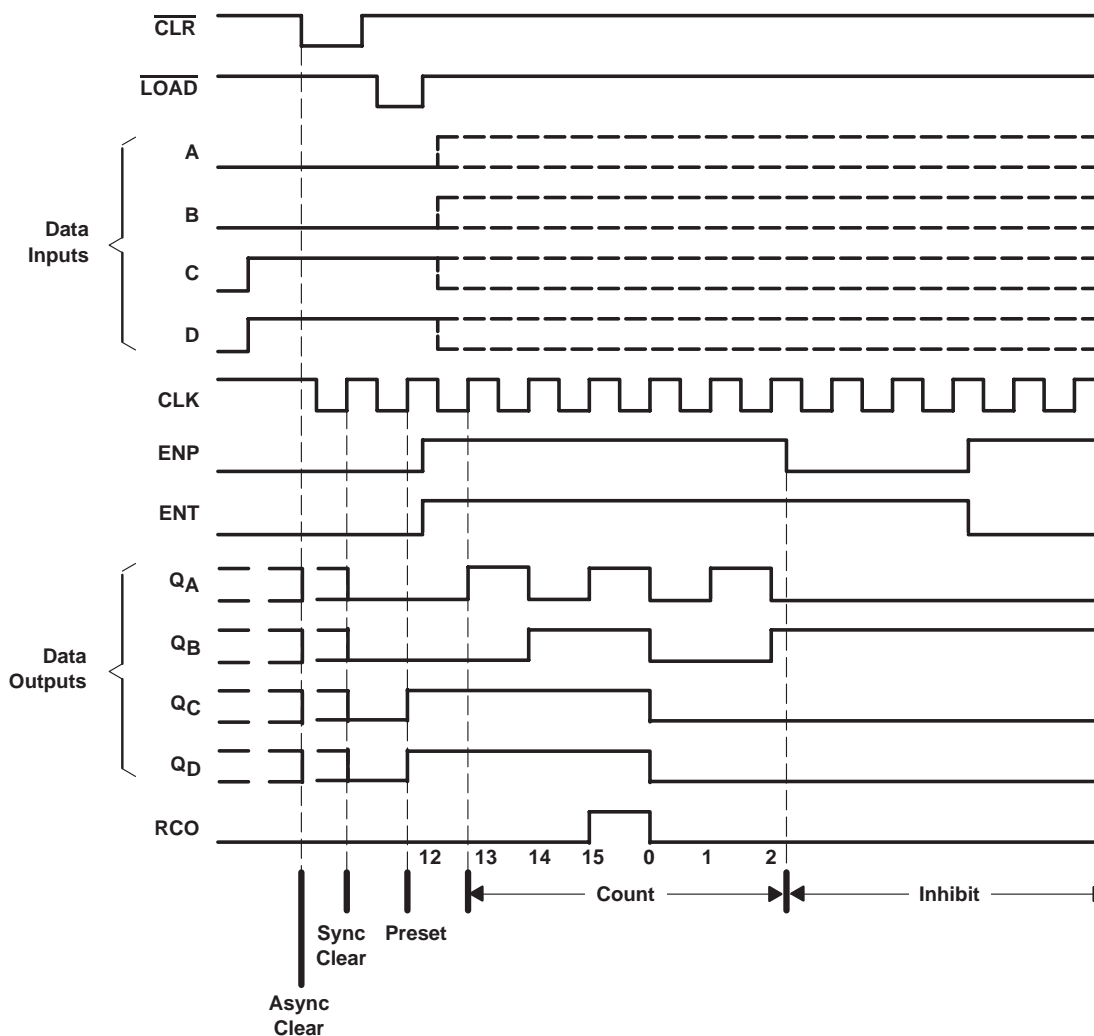
SDAS276 – DECEMBER 1994

typical clear, preset, count, and inhibit sequences

'ALS161B, 'AS161, 'ALS163B, and 'AS163

The following sequence is illustrated below:

1. Clear outputs to zero ('ALS161B and 'AS161 are asynchronous; 'ALS163B and 'AS163 are synchronous.)
2. Preset to binary 12
3. Count to 13, 14, 15, 0, 1, and 2
4. Inhibit



6.3. Resultados Contador A

```
time=    0, Simulation Start
time=    0, Starting Reset
time=   30, Reset Completed
time=   30, Starting Test
time=   30, Starting Semi Random Test
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
time=  230, Starting Completely Random Test
PASS
Time=250 Error dut: Q=0011, rco=1, scoreboard: Q=0011, rco=0
Time=260 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
PASS
Time=280 Error dut: Q=0011, rco=1, scoreboard: Q=0011, rco=0
PASS
PASS
PASS
PASS
Time=330 Error dut: Q=1011, rco=1, scoreboard: Q=1011, rco=0
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
Time=470 Error dut: Q=1001, rco=1, scoreboard: Q=1001, rco=0
PASS
Time=490 Error dut: Q=0110, rco=1, scoreboard: Q=0110, rco=0
PASS
PASS
```

[illegible]

[illegible]

Time=1330 Error dut: Q=1111, rco=1, scoreboard: Q=1111, rco=0

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

time= 1660, Test Completed

time= 1660, Simulation Completed

6.4. Resultados Contador B

```
time=    0, Simulation Start
time=    0, Starting Reset
time=   30, Reset Completed
time=   30, Starting Test
time=   30, Starting Semi Random Test
PASS
Time=50 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
PASS
Time=70 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
PASS
PASS
Time=100 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
PASS
Time=120 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
PASS
time=  230, Starting Completely Random Test
PASS
Time=250 Error dut: Q=0011, rco=1, scoreboard: Q=0011, rco=0
Time=260 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
Time=270 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=280 Error dut: Q=0011, rco=1, scoreboard: Q=0011, rco=0
Time=290 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=300 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=310 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=320 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=330 Error dut: Q=1011, rco=1, scoreboard: Q=1011, rco=0
Time=340 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=350 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=360 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
PASS
PASS
PASS
Time=400 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
PASS
Time=420 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
PASS
Time=440 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=450 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=460 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=470 Error dut: Q=1001, rco=1, scoreboard: Q=1001, rco=0
Time=480 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=490 Error dut: Q=0110, rco=1, scoreboard: Q=0110, rco=0
Time=500 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=510 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
```

Time=520 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
Time=530 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=540 Error dut: Q=1111, rco=1, scoreboard: Q=1111, rco=0
Time=550 Error dut: Q=1111, rco=1, scoreboard: Q=1111, rco=0
Time=560 Error dut: Q=1000, rco=1, scoreboard: Q=1000, rco=0
Time=570 Error dut: Q=1001, rco=1, scoreboard: Q=1001, rco=0
Time=580 Error dut: Q=1010, rco=1, scoreboard: Q=1010, rco=0
Time=590 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=600 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=610 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=620 Error dut: Q=0011, rco=1, scoreboard: Q=0011, rco=0
Time=630 Error dut: Q=0011, rco=1, scoreboard: Q=0011, rco=0
Time=640 Error dut: Q=0100, rco=1, scoreboard: Q=0100, rco=0
PASS
PASS
Time=670 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=680 Error dut: Q=1010, rco=1, scoreboard: Q=1010, rco=0
PASS
Time=700 Error dut: Q=1001, rco=1, scoreboard: Q=1001, rco=0
Time=710 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=720 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=730 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=740 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=750 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=760 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=770 Error dut: Q=1010, rco=1, scoreboard: Q=1010, rco=0
time= 830, Starting Scripted Test
Time=840 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
PASS
Time=860 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
PASS
PASS
PASS
PASS
PASS
PASS
Time=930 Error dut: Q=1001, rco=1, scoreboard: Q=1001, rco=0
PASS
PASS
PASS
PASS
PASS
PASS
PASS
Time=1010 Error dut: Q=1111, rco=0, scoreboard: Q=1111, rco=1
Time=1020 Error dut: Q=1110, rco=1, scoreboard: Q=1110, rco=0

Time=1030 Error dut: Q=1101, rco=1, scoreboard: Q=1101, rco=0
Time=1040 Error dut: Q=1100, rco=1, scoreboard: Q=1100, rco=0
Time=1050 Error dut: Q=1011, rco=1, scoreboard: Q=1011, rco=0
Time=1060 Error dut: Q=1010, rco=1, scoreboard: Q=1010, rco=0
Time=1070 Error dut: Q=1001, rco=1, scoreboard: Q=1001, rco=0
Time=1080 Error dut: Q=1000, rco=1, scoreboard: Q=1000, rco=0
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
Time=1200 Error dut: Q=0100, rco=1, scoreboard: Q=0100, rco=0
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
PASS
Time=1330 Error dut: Q=1111, rco=1, scoreboard: Q=1111, rco=0
PASS
PASS
Time=1360 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
PASS
PASS
Time=1390 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1400 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1410 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1420 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1430 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1440 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1450 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1460 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
PASS
Time=1480 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0

PASS

PASS

```
Time=1510 Error dut: Q=0011, rco=1, scoreboard: Q=0011, rco=0
Time=1520 Error dut: Q=0010, rco=1, scoreboard: Q=0010, rco=0
Time=1530 Error dut: Q=0001, rco=1, scoreboard: Q=0001, rco=0
Time=1540 Error dut: Q=0000, rco=1, scoreboard: Q=0000, rco=0
Time=1550 Error dut: Q=1111, rco=0, scoreboard: Q=1111, rco=1
Time=1560 Error dut: Q=1110, rco=1, scoreboard: Q=1110, rco=0
Time=1570 Error dut: Q=1101, rco=1, scoreboard: Q=1101, rco=0
Time=1580 Error dut: Q=1100, rco=1, scoreboard: Q=1100, rco=0
Time=1590 Error dut: Q=1011, rco=1, scoreboard: Q=1011, rco=0
Time=1600 Error dut: Q=1010, rco=1, scoreboard: Q=1010, rco=0
Time=1610 Error dut: Q=1001, rco=1, scoreboard: Q=1001, rco=0
Time=1620 Error dut: Q=1000, rco=1, scoreboard: Q=1000, rco=0
Time=1630 Error dut: Q=0111, rco=1, scoreboard: Q=0111, rco=0
Time=1640 Error dut: Q=0110, rco=1, scoreboard: Q=0110, rco=0
Time=1650 Error dut: Q=0101, rco=1, scoreboard: Q=0101, rco=0
Time=1660 Error dut: Q=0100, rco=1, scoreboard: Q=0100, rco=0
time= 1660, Test Completed
time= 1660, Simulation Completed
```

6.5. Resultados Contador C

[illegible]

[illegible]

time= 830, Starting Scripted Test

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

Time=1110 Error dut: Q=1111, rco=0, scoreboard: Q=1111, rco=1

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

Time=1270 Error dut: Q=1111, rco=0, scoreboard: Q=1111, rco=1

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

PASS

time= 1660, Test Completed

time= 1660, Simulation Completed