

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



sábado, 21 de octubre de 2017, Ciudad Universitaria, México, DF

Diseñe un circuito secuencial síncrono que efectúe un conteo por décadas, que realice las siguientes funciones:

1. Conteo ascendente
2. Conteo descendente
3. Iniciar en un conteo específico
4. Detener conteo

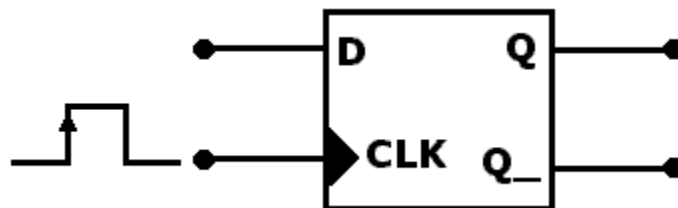
0 conteo
1 ascendente completo
10 completo descendente
11 conteo detenido

INTRODUCCION:

FLIP FLOP D (SYNCHRONIZED FOR CLOCK)

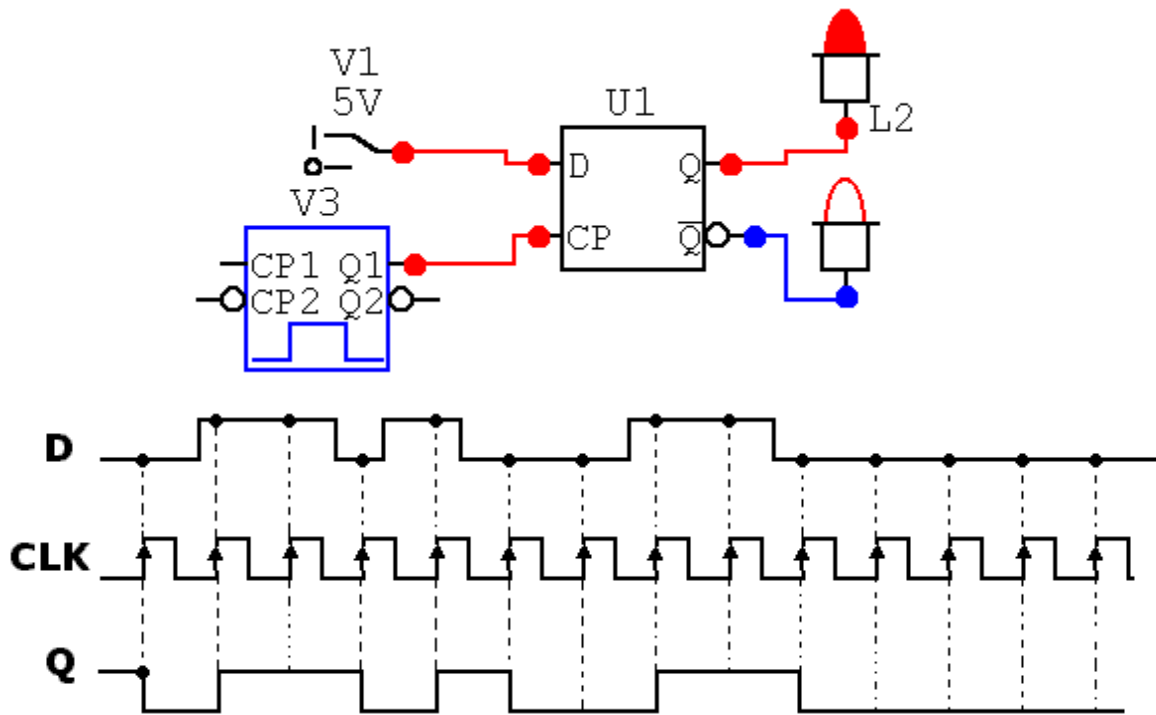
A diferencia de los flip flops S-C y J-K, este FF sólo tiene una entrada síncrona de control, D, que significa Datos. Q pasará al mismo estado que este presente en la entrada D cuando ocurra una TPP en CLK.

En otras palabras, el nivel presente en D se almacenará en el FF en el instante en que ocurre la TTP.



Entradas			Salidas
D	CLK		Q
0	↑		0
1	↑		1

Simulación en Circuit Maker incompleta:

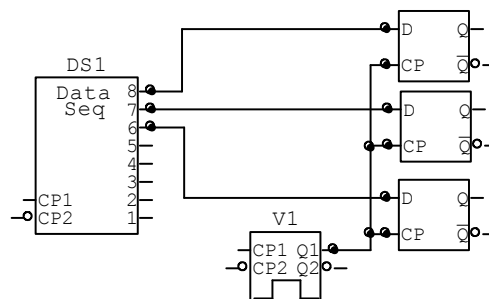


TRANSFERENCIA DE DATOS PARALELA

En este punto quizá usted se pregunte cual es la utilidad del flip flop D, puesto que según parece la salida Q es la misma que la entrada D. No del todo, recuerde Q adopta el valor de D solo en ciertos momentos, y por lo tanto no es idéntica a D.

En la mayoría de las aplicaciones del FF D la salida Q debe adoptar el valor en su entrada D sólo en instantes definidos.

Un ejemplo de esto se ilustra en la figura siguiente:



Digamos que el data seq es un circuito lógico combinacional, las salidas 8, 7, 6 se transferirían a Q1, Q2, Q3 del FF para su almacenamiento.

Usando los FF D, los niveles presentes en 8, 7, 6 se transferirán a Q1, Q2, Q3, respectivamente al aplicar un pulso de TRANSFERENCIA a las entradas comunes CLK. Los FFs pueden almacenar estos valores para su procesamiento siguiente.

Este es un ejemplo de transferencia paralela de datos binarios, los bits 8, 7, 6 se transfieren simultáneamente.

	Var. entrada		Estado actual				salidas		Estado siguiente				Elemento de memoria			
	C ₁	C ₀	A	B	C	D	Y ₁	Y ₀	A ⁺	B ⁺	C ⁺	D ⁺	D _A	D _B	D _C	D _D
mo	0	0	0	0	0	0	0	0	A	b	c	d	a	b	c	d
m1	0	0	0	0	0	1	0	0	A	b	c	d	a	b	c	d
m2	0	0	0	0	1	0	0	0	A	b	c	d	a	b	c	d
m3	0	0	0	0	1	1	0	0	A	b	c	d	a	b	c	d
m4	0	0	0	1	0	0	0	0	A	b	c	d	a	b	c	d
m5	0	0	0	1	0	1	0	0	A	b	c	d	a	b	c	d
m6	0	0	0	1	1	0	0	0	A	b	c	d	a	b	c	d
m7	0	0	0	1	1	1	0	0	A	b	c	d	a	b	c	d
m8	0	0	1	0	0	0	0	0	A	b	c	d	a	b	c	d
m9	0	0	1	0	0	1	0	0	A	b	c	d	a	b	c	d
m10	0	0	1	0	1	0	*	*	*	*	*	*	*	*	*	*
m11	0	0	1	0	1	1	*	*	*	*	*	*	*	*	*	*
m12	0	0	1	1	0	0	*	*	*	*	*	*	*	*	*	*
m13	0	0	1	1	0	1	*	*	*	*	*	*	*	*	*	*
m14	0	0	1	1	1	0	*	*	*	*	*	*	*	*	*	*
m15	0	0	1	1	1	1	*	*	*	*	*	*	*	*	*	*

	C ₁	C ₀	A	B	C	D	Y ₁	Y ₀	A ⁺	B ⁺	C ⁺	D ⁺	D _A	D _B	D _C	D _D
m16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
m17	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0
m18	0	1	0	0	1	0	0	0	0	0	1	0	0	0	1	1
m19	0	1	0	0	1	1	0	0	0	0	1	1	0	1	0	0
m20	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	1
m21	0	1	0	1	0	1	0	0	0	1	0	1	0	1	1	0
m22	0	1	0	1	1	0	0	0	0	1	1	0	0	1	1	1
m23	0	1	0	1	1	1	0	0	0	1	1	1	1	0	0	0
m24	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	1
m25	0	1	1	0	0	1	0	1	1	0	0	1	0	0	0	0
m26	0	1	1	0	1	0	*	*	*	*	*	*	*	*	*	*
m27	0	1	1	0	1	1	*	*	*	*	*	*	*	*	*	*
m28	0	1	1	1	0	0	*	*	*	*	*	*	*	*	*	*
m29	0	1	1	1	0	1	*	*	*	*	*	*	*	*	*	*
m30	0	1	1	1	1	0	*	*	*	*	*	*	*	*	*	*
m31	0	1	1	1	1	1	*	*	*	*	*	*	*	*	*	*

	C ₁	C ₀	A	B	C	D	Y ₁	Y ₀	A ⁺	B ⁺	C ⁺	D ⁺	D _A	D _B	D _C	D _D
m32	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	1
m33	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
m34	1	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1
m35	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	0
m36	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	1
m37	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0
m38	1	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1
m39	1	0	0	1	1	1	0	0	0	0	1	0	0	1	1	0
m40	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1
m41	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
m42	1	0	1	0	1	0	*	*	*	*	*	*	*	*	*	*
m43	1	0	1	0	1	1	*	*	*	*	*	*	*	*	*	*
m44	1	0	1	1	0	0	*	*	*	*	*	*	*	*	*	*
m45	1	0	1	1	0	1	*	*	*	*	*	*	*	*	*	*
m46	1	0	1	1	1	0	*	*	*	*	*	*	*	*	*	*
m47	1	0	1	1	1	1	*	*	*	*	*	*	*	*	*	*

	C ₁	C ₀	A	B	C	D	Y ₁	Y ₀	A ⁺	B ⁺	C ⁺	D ⁺	D _A	D _B	D _C	D _D
m48	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0
m49	1	1	0	0	0	1	1	1	0	0	0	1	0	0	0	1
m50	1	1	0	0	1	0	1	1	0	0	1	0	0	0	1	0
m51	1	1	0	0	1	1	1	1	0	0	1	1	0	0	1	1
m52	1	1	0	1	0	0	1	1	0	1	0	0	0	1	0	0
m53	1	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1
m54	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0
m55	1	1	0	1	1	1	1	1	0	1	1	1	0	1	1	1
m56	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	0
m57	1	1	1	0	0	1	1	1	1	0	0	1	1	0	0	1
m58	1	1	1	0	1	0	*	*	*	*	*	*	*	*	*	*
m59	1	1	1	0	1	1	*	*	*	*	*	*	*	*	*	*
m60	1	1	1	1	0	0	*	*	*	*	*	*	*	*	*	*
m61	1	1	1	1	0	1	*	*	*	*	*	*	*	*	*	*
m62	1	1	1	1	1	0	*	*	*	*	*	*	*	*	*	*
m63	1	1	1	1	1	1	*	*	*	*	*	*	*	*	*	*

$$Y_1(C_1, C_0, A, B, C, D) = \sum (32, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57) \text{ ok}$$

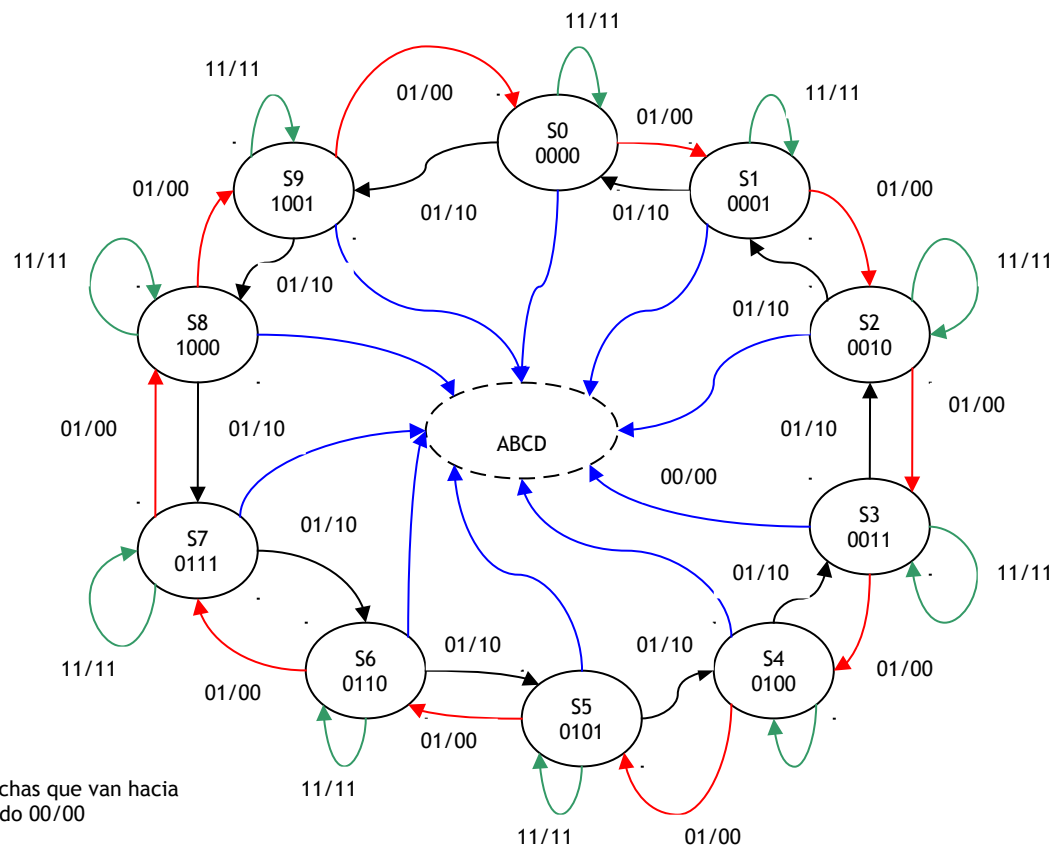
$$Y_0(C_1, C_0, A, B, C, D) = \sum (25, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57) \text{ ok}$$

$$D_A(C_1, C_0, A, B, C, D) = \sum (0-9) a + \sum (23, 24, 32, 41, 56, 57)$$

$$D_B(C_1, C_0, A, B, C, D) = \sum (0-9) b + \sum (19, 20, 21, 22, 37, 38, 39, 40, 52, 53, 54, 55)$$

$$D_C(C_1, C_0, A, B, C, D) = \sum (0-9) c + \sum (17, 18, 21, 22, 35, 36, 39, 40, 50, 51, 54, 55)$$

$$D_D(C_1, C_0, A, B, C, D) = \sum (0-9) d + \sum_{+d'} (16, 18, 20, 22, 24, 32, 34, 36, 38, 40, 49, 51, 53, 55, 57) \\ +d' (10-15, 26-31, 42-47, 58-63)$$



Minimizando:

