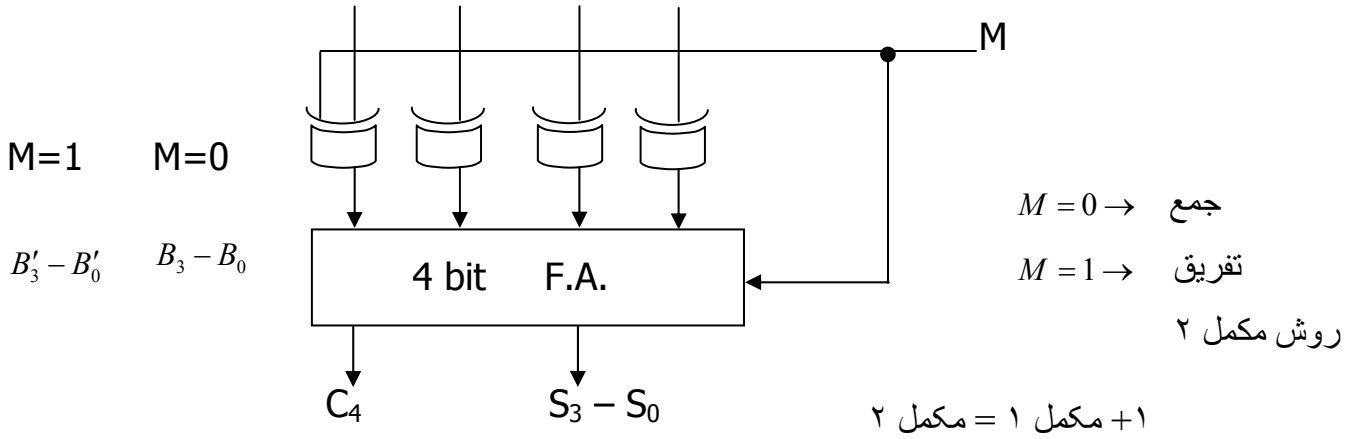
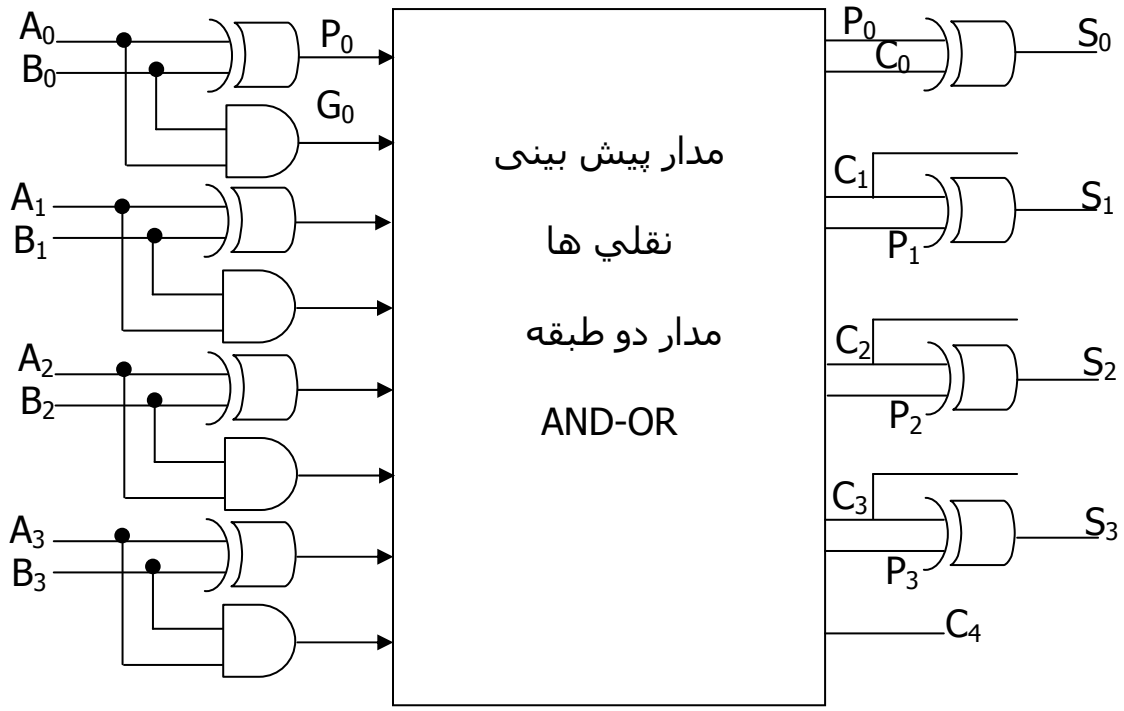
۵- طراحی مدار جمع کننده / تفریق کننده ی ۴ بیتی با روش مکمل ۲ با خط کنترل M:



نکته : در مدارهای جمع کننده تاخیر تجمعی داریم که با افزایش تعداد بیت های ورودی افزایش می یابد . برای مدار جمع کننده ی ۲ بیتی ۵ طبقه تاخیر داشتیم پس این مدار مشکل دارد پس مدار جمع کننده با نقلی پیش بینی شده طراحی شد .

۶- مدار جمع کننده با نقلی پیش بینی شده (Look Ahead Carry Generator) :



$S_0 = G_0 \oplus P$

$C_1 = G_0 + C_0 P_0$

$S_1 = C_1 \oplus P_1$

$C_2 = G_1 + C_1 P_1$

$C_2 = G_1 + P_1 G_0 + P_1 P_2 C_0$

$S_2 = G_2 \oplus P_2$

$C_3 = G_2 + C_2 P_2$

$C_3 = G_2 + P_2 G_1 + P_1 P_2 G_1 + P_1 P_2 P_0 G_0$

$P_i = A_i \oplus B_i$

$B_i = A_i \,.\, B_i$

$S_i = P_i \oplus C_i$

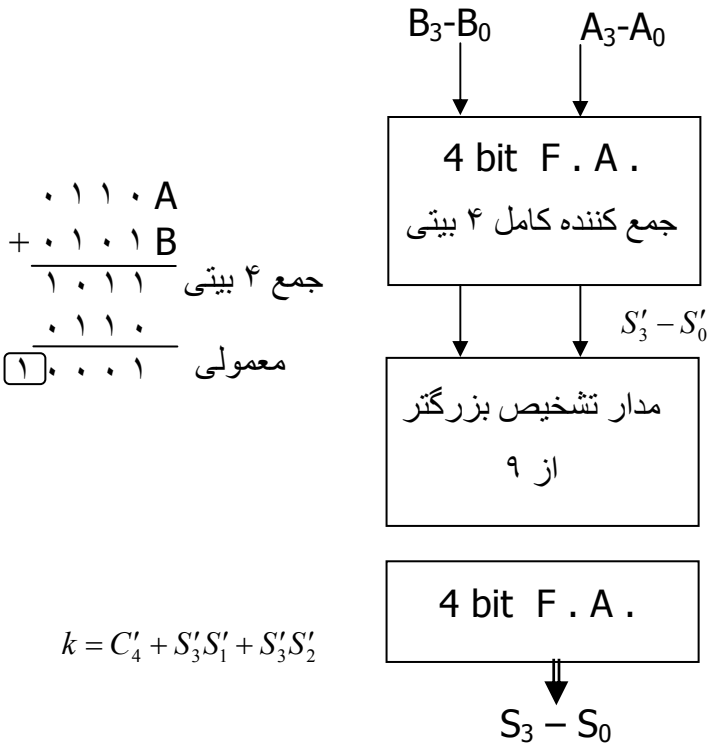
$C_i = G_{i-1} + C_{i-1} P_{i-1}, C_0$

داریم

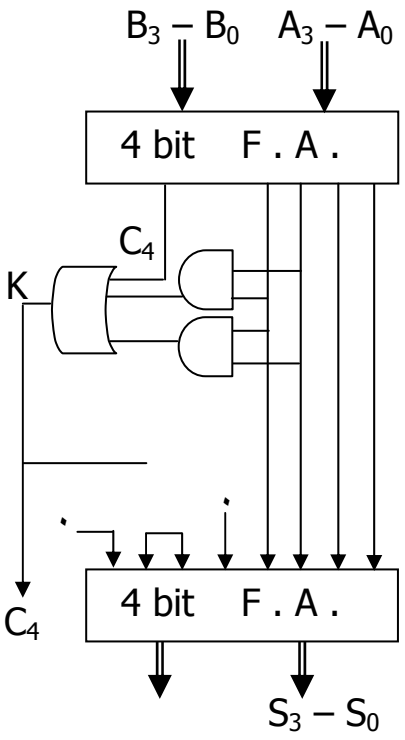
نکته : این مدار و کلیه مدارهای توسعه یافته (6,8) و ... بیتی) می توانند با این روش با 4 طبقه تاخیر طراحی شوند .

۷- مدار جمع کننده BCD :

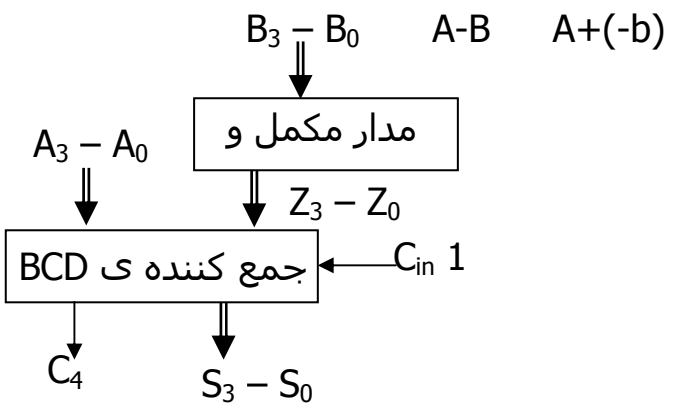
C_4'	S_3'	S_2'	S_1'	S_0'	K
۰	۰	۰	۰	۰	۰
۰	۰	۰	۰	۱	۰
۰	۰	۰	۱	۰	۰
۰	۱	۰	۰	۱	۰
۰	۱	۰	۱	۰	۱
۰	۱	۰	۱	۱	۱
۱	۰	۰	۰	۰	۱
۱	۰	۰	۰	۱	۱
۱	۰	۰	۱	۰	۱
۱	۰	۰	۱	۱	*



پیاده سازی :



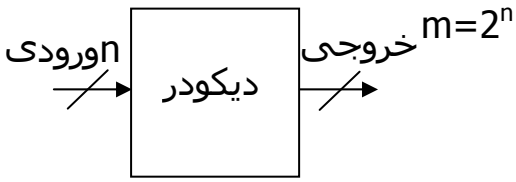
۸- مدارهای تفریق کننده ی BCD :

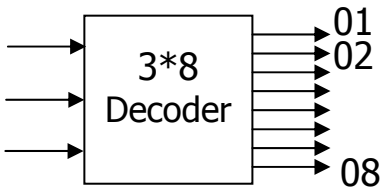


B 3	B 2	B 1	B 0	Z 3	Z 2	Z 1	Z 0
۰	۰	۰	۰	۱	۰	۰	۱
۰	۰	۰	۱	۱	۰	۰	۰
۰	۰	۱	۰	۰	۱	۱	۱
۰	۰	۱	۱	۰	۱	۱	۰
۰	۱	۰	۰	۰	۱	۰	۱
۰	۱	۰	۱	۰	۱	۰	۰
۰	۱	۱	۰	۰	۱	۱	۱
۰	۱	۱	۱	۰	۰	۱	۰
۱	۰	۰	۰	۰	۰	۰	۱
۱	۰	۰	۱	۰	۰	۰	۰

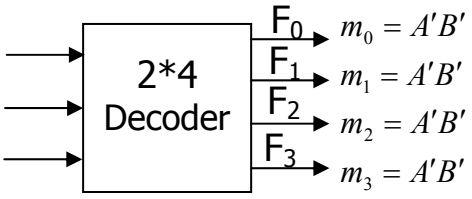
۹- مدارهای دیکودر (Decoder) :

مدار دیکودر برای n متغیر ورودی 2^n حالت آنرا ایجاد می نماید.

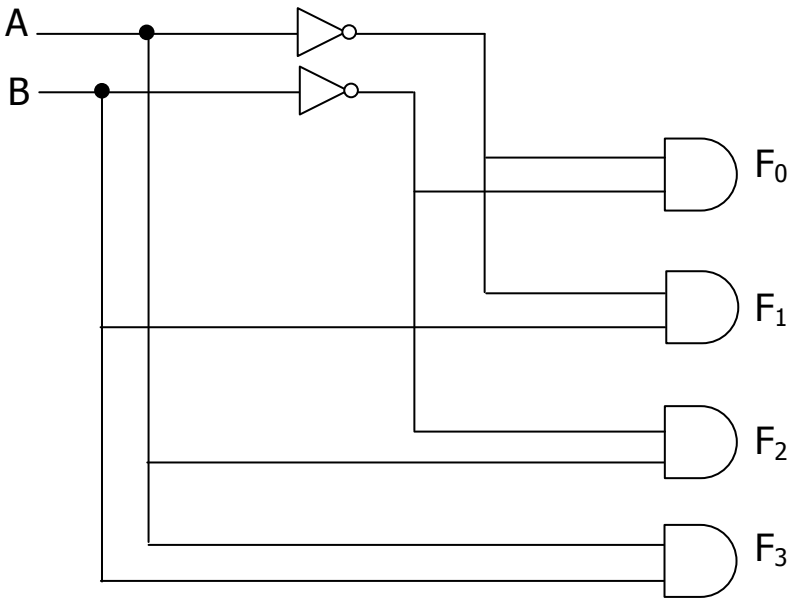




نکته : حالات مختلف متغیرها را در خروجی می دهد .



A	B	F_0	F_1	F_2	F_3
۰	۰	۱	۰	۰	۰
۰	۱	۰	۱	۰	۰
۱	۰	۰	۰	۱	۰
۱	۱	۰	۰	۰	۱



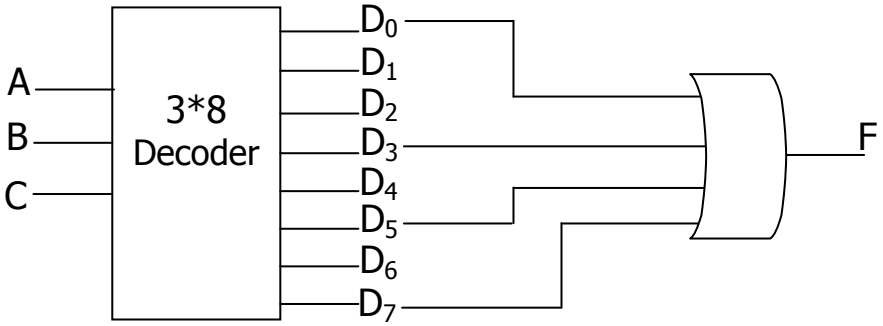
نکته : اگر به جای AND از NAND استفاده کنیم آنگاه Maxterm ها خواهیم داشت :

کاربردها :

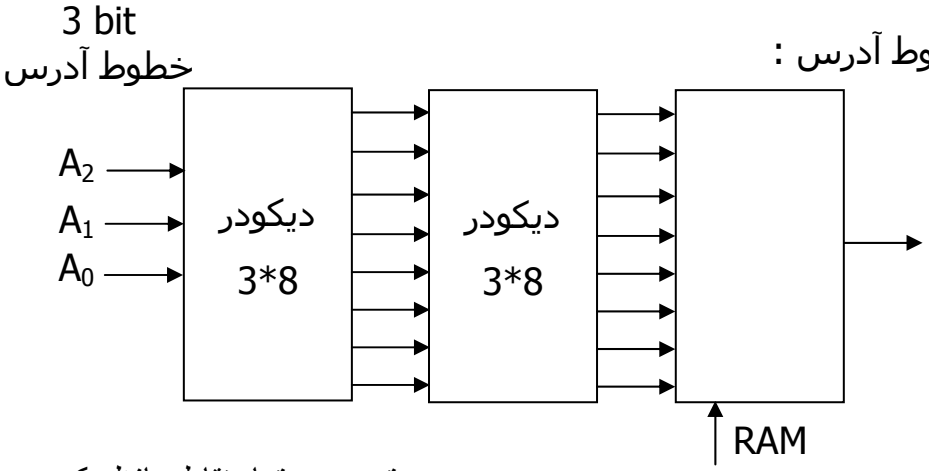
۱- در پیاده سازی توابع :

$$F = (A,B,C) = \sum m(0,3,5,7)$$

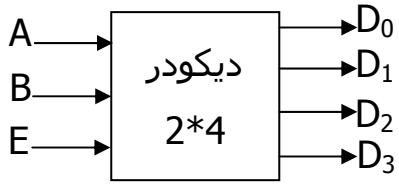
تابع را به کمك يك ديكودر 3*8 پياده سازی نمائيد .

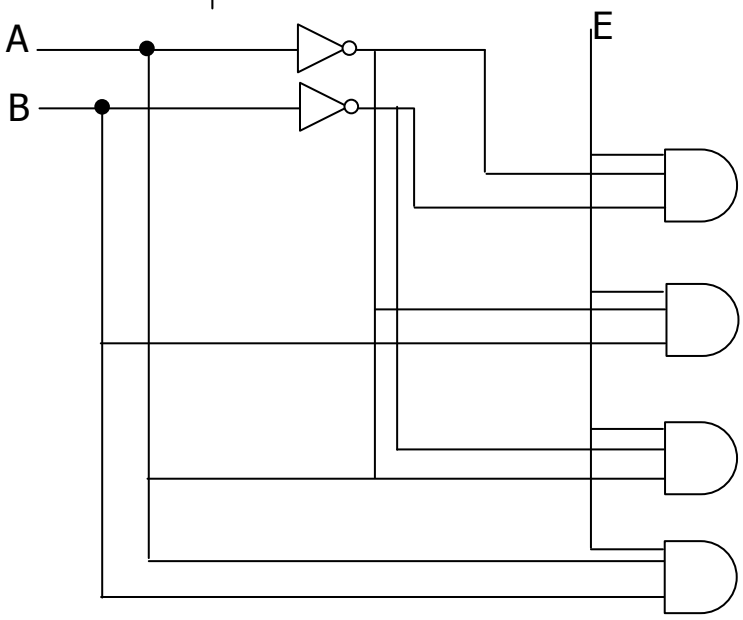


۲- در رمز گشايي خطوط آدرس :



دسترسي به تمام نقاط حافظه كه در اين مثال 8 حافظه است .





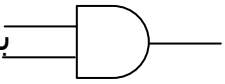
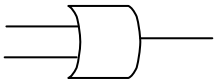
دیکودر عمل نمي كند E=0

Enable

Active High enable

E	A	B	D_0	D_1	D_2	D_3
۰	*	*	۰	۰	۰	۰
۱	۰	۰	۱	۰	۰	۰
۱	۰	۱	۰	۱	۰	۰
۱	۱	۰	۰	۰	۱	۰
۱	۱	۱	۰	۰	۰	۱

نکته : در حالت ماکسترمی نیز درصورتی کـه E=0 ، B,A نیـز Den't care مـی باشـد .

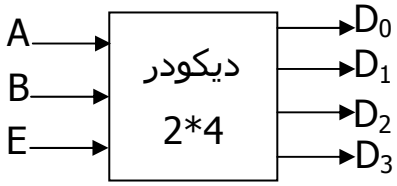
در صورتی که به جای  ،  بگـذاریم جـدول زیـر را خـواهیم داشـت .

E	A	B	D_0	D_1	D_2	D_3
1	*	*	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

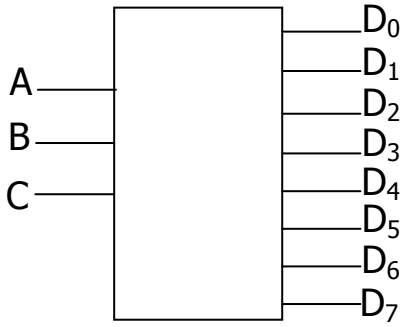
Enable

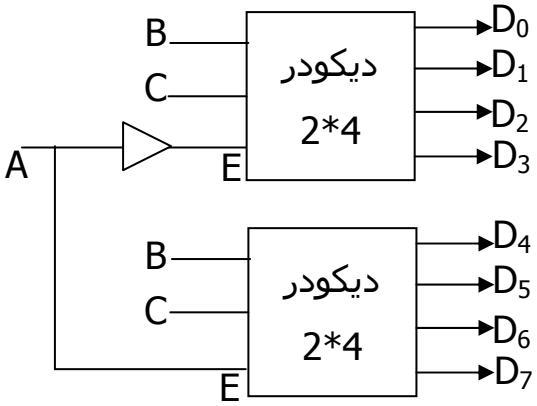
Active High enable

يعني Enable بايد NOT شـود به صورت زير



با كمك ديكودر هاي 2*4 ، ديكودر 3*8 بسازيد .



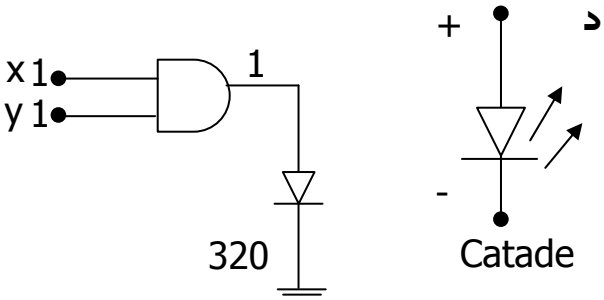


E	A	B	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰
۰	۰	۱	۰	۱	۰	۰	۰	۰	۰	۰
۰	۱	۰	۰	۰	۱	۰	۰	۰	۰	۰
۰	۱	۱	۰	۰	۰	۱	۰	۰	۰	۰
۱	۰	۰					۱			
۱	۰	۱						۱		
۱	۱	۰							۱	
۱	۱	۱								۱

دیکودر BCD به 7.seg (Seven Segment)

قطعه الکترونیکي LED : (دیود نوري) با نماد

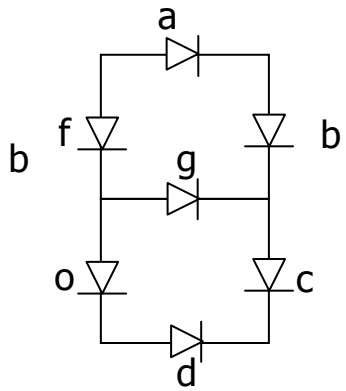
(Light Emitting Diode)



اگر [v]2 به دو سر LED اختلاف پتانسیل اعمال شود روشن مي شود .

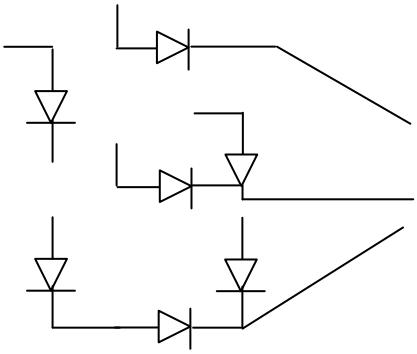
در نتیجه :

7.seg صفحه نمایش یا خروجي یك سیستم دیجیتال مي توانند باشد .



نکته :

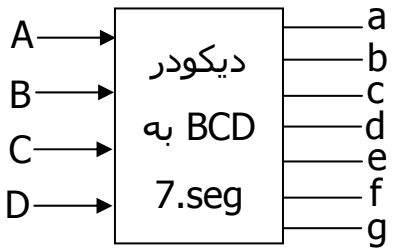
اگر کاتدها را به هم وصل کنیم و از طریق آنها فرمان بدهیم مي گویند (کاتد مشترك 7.seg) و اگر آن را به هم متصل و در کل به جز اي که مي خواهیم روي آن کار انجام دهیم وصل کنیم و از طریق کاتدها فرمان دهیم (آنرا مشترك 7.seg)



کاتد مشترک

A	B	C	D		a	b	c	d	e	f	g
۰	۰	۰	۰	0	۱	۱	۱	۱	۱	۱	۰
۰	۰	۰	۱	1	۰	۱	۱	۰	۰	۰	۰
۰	۰	۱	۰	2	۱	۱	۰	۱	۱	۰	۱
۰	۰	۱	۱	3	۱	۱	۱	۱	۰	۰	۱
۰	۱	۰	۰	4	۰	۱	۱	۰	۰	۱	۱
۰	۱	۰	۱	5	۱	۰	۱	۱	۰	۱	۱
۰	۱	۱	۰	6	۰	۰	۱	۱	۱	۱	۱
۰	۱	۱	۱	7	۱	۱	۱	۰	۰	۰	۰
۱	۰	۰	۰	8	۱	۱	۱	۱	۱	۱	۱
۱	۰	۰	۱	9	۱	۱	۱	۱	۰	۱	۱

باقي مانده Don't car هستند

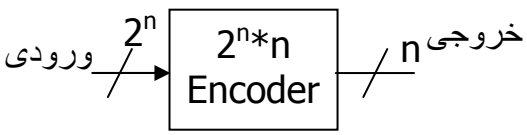


کاتد مشترک است

نکته :

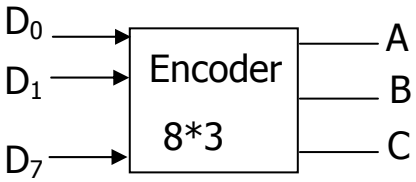
۱۰- مدارهای Encoder یا رمز گذار :

مدار انکودر برای رمزگذاری یا فشرده سازی متغیرهای ورودی بکار می رود.

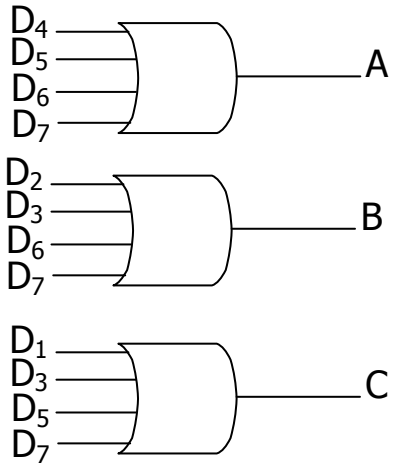


ورودی	خروجی
7	3
11	4
16	4

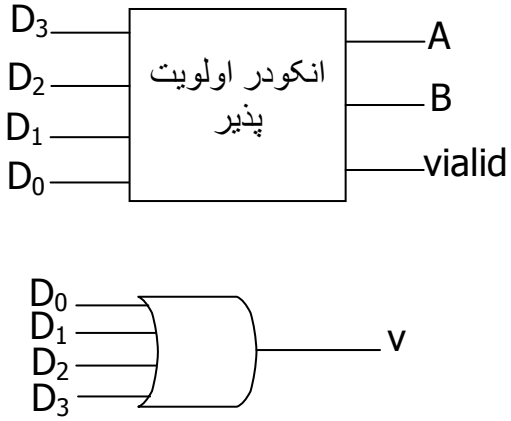
حالت متعارف



D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1								۰	۰	۰
	1							۰	۰	۱
		1			0			۰	۱	۰
			1					۰	۱	۱
				1				۱	۰	۰
		0			1			۱	۰	۱
						1		۱	۱	۰
							1	۱	۱	۱



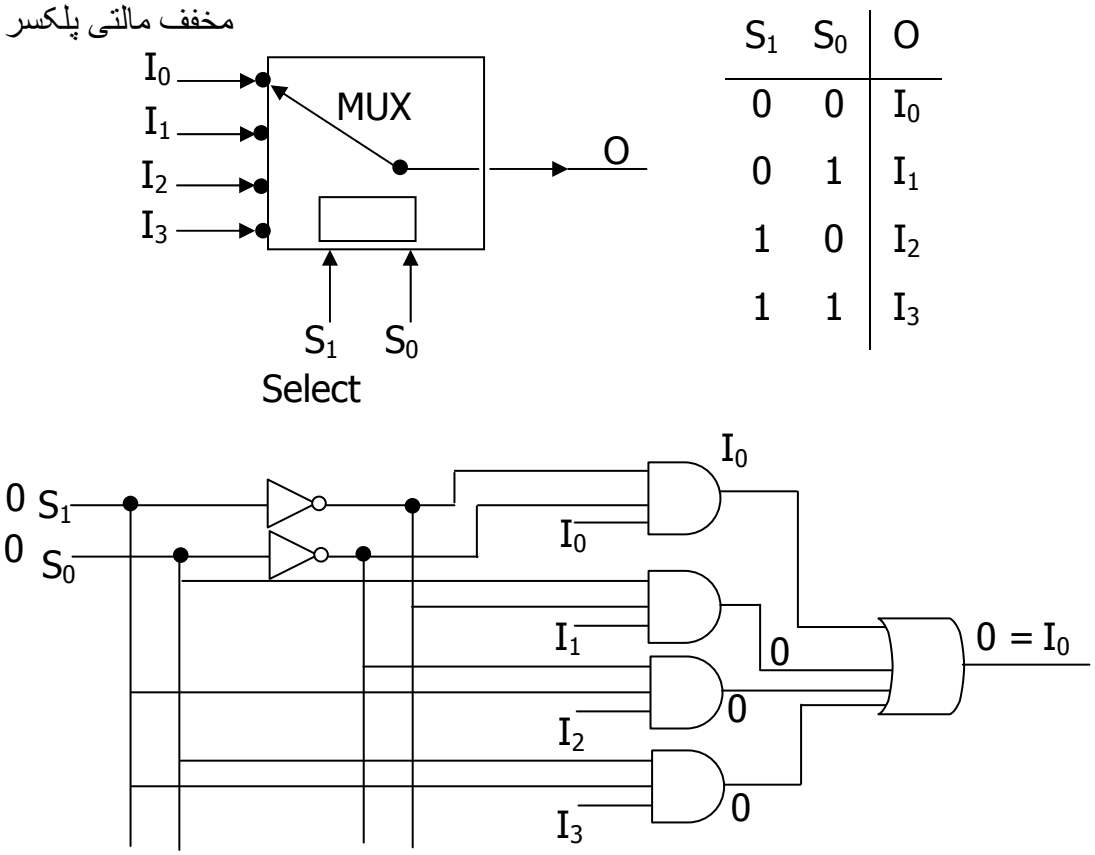
D_3	D_2	D_1	D_0	A	B	V
۱	*	*	*	۱	۱	۱
۰	۱	*	*	۱	۰	۱
۰	۰	۱	*	۰	۱	۱
۰	۰	۰	۱	۰	۰	۱
۰	۰		۰	۰	۰	۰



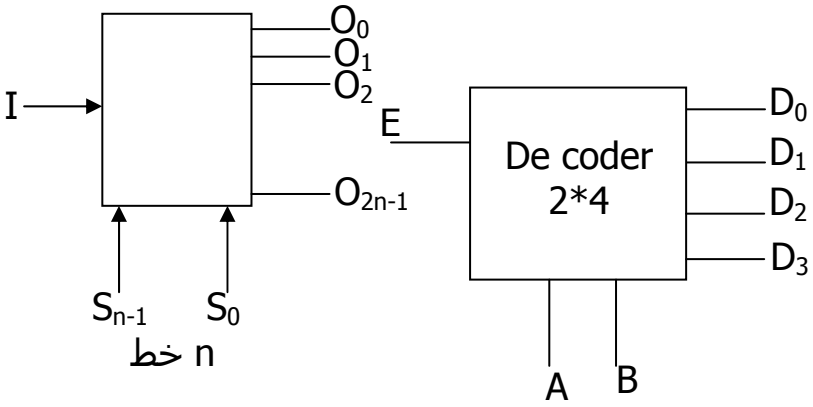
$$A = D_3 + D_2 + D_3' = (D_3 + D_2)(D_3 + D_3') = D_3 + D_2$$
$$B = D_3 + D_3'D_2'D_1 = D_3 + D_2'D_1$$

کاربرد : اولویت گذاري Interrupt مانند وقفه کیبورد ، سدی درایو ، کارت صوتي و VGA

Multiplexer : با کمك n انتخاب 2^n ورودی را روی یک خروجی می فرستد .



۱۱- مدارهای ترکیبی – دی مالتی پلکسر Demulti plexer :



شبیه سازی Demax

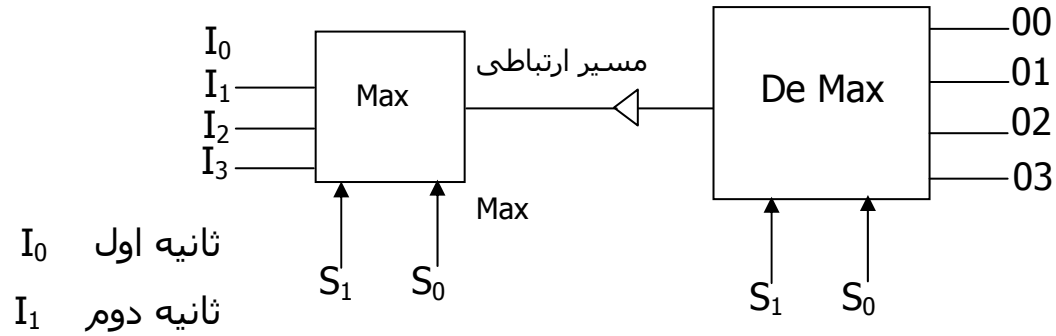
A	B	C	D_0	D_1	D_2	D_3
۰	۰	۰	۰	۰	۰	۰
۰	۰	۱	۰	۰	۰	۰
۰	۱	۰	۰	۰	۰	۰
۰	۱	۱	۰	۰	۰	۰
۱	۰	۰	۱	۰	۰	۰
۱	۰	۱	۰	۱	۰	۰
۱	۱	۰	۰	۰	۱	۰
۱	۱	۱	۰	۰	۰	۱

لذا مدار Demax همان دیکودر با یک ورودی Enable است.

کاربرد :

در ارتباط مخابراتی مانند تلفنهای منازل یک منطقه :

به این روش ارسال Time Domain Multiplexing (TDM) گویند .



I_0 ثانیه اول

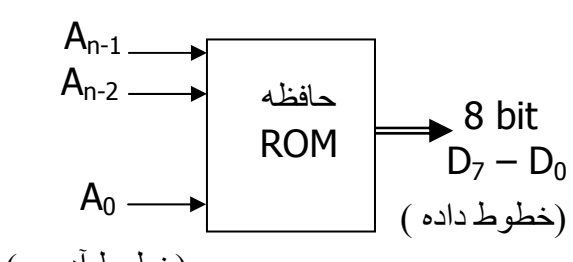
I_1 ثانیه دوم

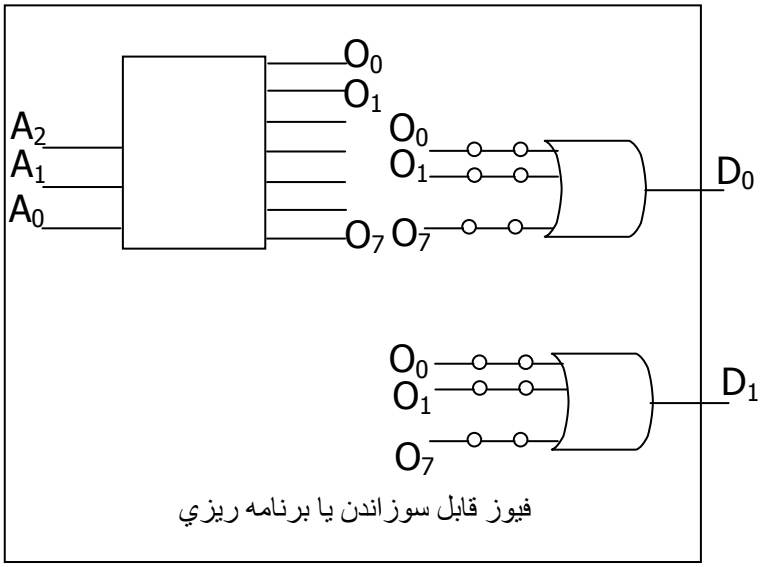
I_2 ثانیه سوم

I_3 ثانیه چهارم

۱۲- مدارهای ترکیبی – قطعات قابل برنامه ریزی (ROM)،(PLA) ،(CAL)،(PLD) و ... :

ROM یا Read Only Memory در واقع مدارهای ترکیبی هستند .





ROM 8 * 4

ROM : 8 خانه حافظه – هر خانه 4 بيتي يا 4 bit word

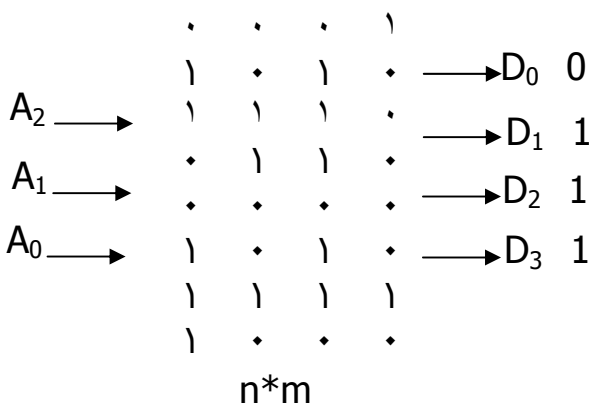
نكته : فيوز هاي 4,3,0 در D_3 سوزانده مي شوند .

نكته : ما در ROM تمام فينترمها را داريم و با or هاي قابل برنامه ريزي و با توجه به اطلاعات مشخص مي شود براي هر خروجي كدام فيوز ها سوزانده شود .

كاربرد ROM :

۱- پیاده سازی توابع

۲- یک المان حافظه



استفاده از Rom به عنوان پیاده سازی توابع :

مداري تركيبي داريم كه n ورودي و m خروجي دارد و بدين منظور مي توان از 2^n*k Rom كه $k \geq m$ استفاده نمود.

نكته :

چون فیوزهای سوخته شده دیگر قابل باز یافت نیستند لذا Rom ها فقط یکبار قابل خواندن مي باشد . براي رفع اين مشكل EPROM طراحي شده است . كه با اشعه ي ماوراء بنفش پاك مي شود و با مدارهاي الكتريكي برنامه ريزي مي شوند .

ولي EPROM ها مدت زيادي را براي طراحي پاك شدن مصرف مي كنند (حدود 15-16 دقيقه) لذا EEPROM را طراحي كردند كه بوسيله ي مدارهاي الكتريكي پاك مي شوند كه بسيار كم زمان مي برند (ميلي ثانيه).

مدارهاي ترتيبی :

مدارهای منطقی ترتیبی :

خروجی علاوه براینکه به ورودیهای مدار بستگی دارد به خروجیهای قبلی مدار (و در نتجه به ورودیهای قبلی) بستگی دارد . پس می توان گفت مدار ترتیبی حافظه دار است .

انواع مدارهای ترتیبی :

۱- سنکرون یا همزمان : همزمان با سیگنالی بنام کلاک تغییرات در خروجی مدار و در اثر ورودی صورت می گیرد .

۲- آسنکرون یا غیر همزمان : تغییرات خروجی بدون سیگنال با تغییر ورودی امکان پذیر است .

موضوعات :

۱- آشنایی با انواع فلیپ فلاپها

۲- آشنایی با برخی مدارهای ترتیبی

۳- طراحی مدارهای منطقی ترتیبی

فليپ فلاپها (Flip-Flop یا FF) :

مدار منطقي است كه مي تواند يك بيت اطلاعات را براي ما نگه دارد .

انواع FF :

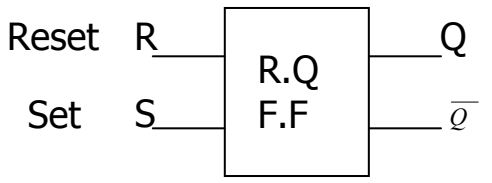
۱- Rs-FF

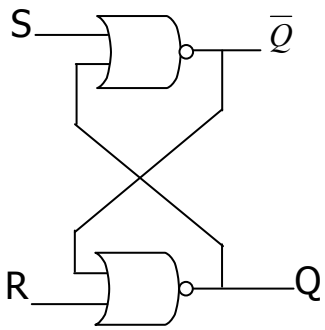
۲- D-FF

۳- jk-FF

T-FF -۴

: Rs-FF



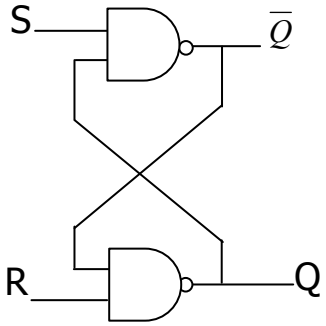


R	S	Q	$\overline{Q}$
۱	۰	۰	۱
۰	۰	۰	۱
۰	۱	۱	۰
۰	۰	۱	۰
۱	۱	۰	۰

یا Set – Reset FF

$\overline{Q}$ = NOT = Q

از خروجي Q يك Feed book يا باز خورد زديم به ورودي NOR



R	S	Q_{n-1}	$\overline{Q}_{n+1}$	وضعيت (عمل)
۱	۱	Q_n	$\overline{Q}_n$	وضعيت قبلي(Hold)
۰	۱	۰	۱	Reset
۱	۱	۰	۱	Hold
۱	۰	۱	۰	Set
۱	۱	۱	۰	Hold
۰	۰	۱	۱	غير مجاز

نكته : هرگاه ورودي را صفر كنيم تاثير منفي مي گذارد.

نكته : خروجي وضعيت قبلي به صورت يك ورودي به طور منفي منظور مي گردد.

جدول صحت (براي ساختار NOR) :

R	S	Q_n	Q_{n-1}		$\overline{Q}_{n+1}$
۰	۰	۰	۰		۱
۰	۰	۱	۱		۰
۰	۱	۰	۱		۰
۰	۱	۱	۱		۰
۱	۰	۰	۰		۱
۱	۰	۱	۰		۱
۱	۱	۰	*	Don't care	*
۱	۱	۱	*	Don't care	*

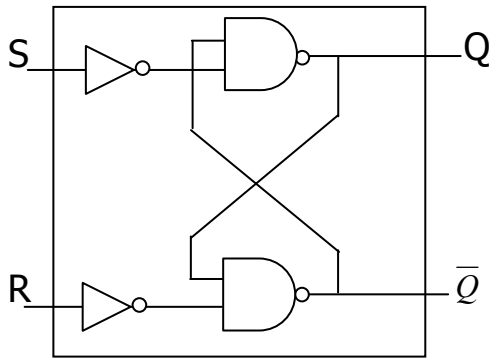
RQn / S	Q_{n-1}			
۰	۰	۱	۰	۰
۱	۱	۱	*	*

RQn / S	$\overline{Q}_{n-1}$			
۰	۱	۰	۱	۱
۱	۰	۰	*	*

$$Q_{n+1} = S + R'Q_n$$
$$Rs = 0$$
$$\overline{Q}_{n+1} = R + SQ_n$$

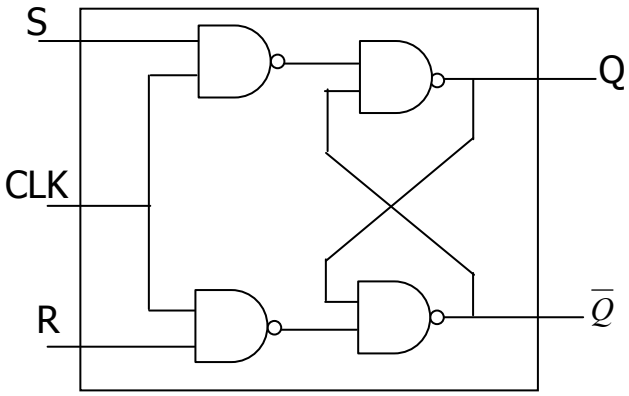
S,R با هم ترکیب نمي شود .

High Active : يك موثر است .



S	R	Q	Q'
۰	۰	Hold	
۱	۰	۱	۰
۰	۱	۰	۱
۱	۱	غیر مجاز	

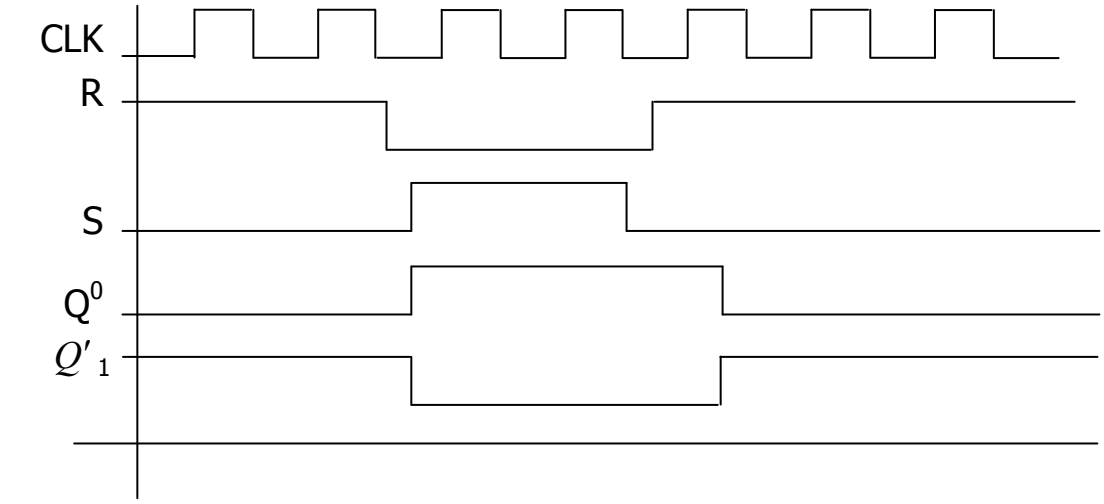
فیلیپ فلاپ با کلاك :

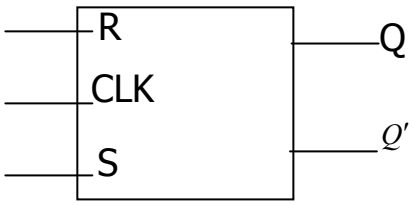


www . bme711 . ir

CLK	R	S	Q_{n-1}	Q'_{n+1}	وضعیت
۰	*	*	Q_n	Q'_n	Hold
۱	۰	۱	۱	۰	Set
۱	۱	۰	۰	۱	Reset
۱	۰	۰	Q_n	Q'_n	Hold
۱	۱	۱	۱	۱	غیر مجاز

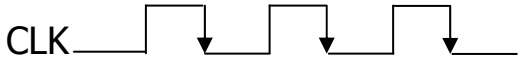
نکته : زماني خروجي تغيير مي كند كه Clock بيايد در غير اينصورت Hold يعني بايد Clock يك شود تا بعد R و S را بررسي كنيم در غير اينصورت R و S هر چه باشند Hold داريم .



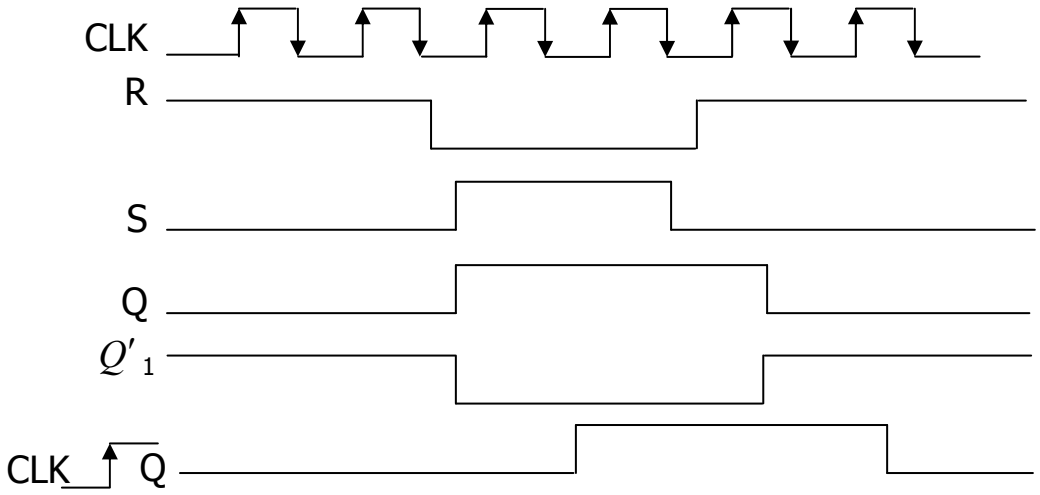


ورودي كلاك حساس به سطح يا FF يا حساس به سطح كلاك

F.F حساس به لبه كلاك : مدار داخلي از كتاب مانول ، زماني خروجي تغيير مي كند كه به لبه كلاك بيايد .



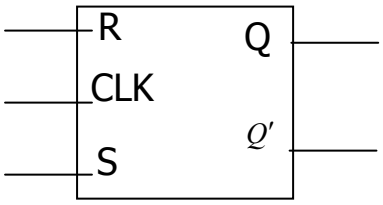
دو حالت مي تواند پيش آيد : حساس به لبه مثبت كلاك (بالا رونده ، يا حساس به لبه منفي كلاك (پايين رونده)



نكته : فقط در لحظه لبه ورودي هاي R,S را بررسي مي كند و در غير از آن هيچ كار نمي كند (Hold) حتي براي پايين رونده نيز هيچ كاري نمي كند .

نكته : F.F حساس به لبه در مدارهاي ترتيبي سنكرون استفاده مي شود و F.F هاي حساس به سطح در مدارهاي ترتيبي آسنكرون استفاده مي شود .

نمايش سمبليك :



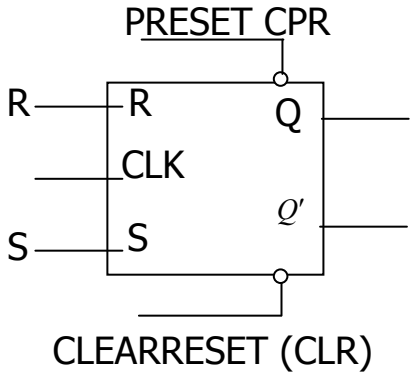
ترسیم جدول :

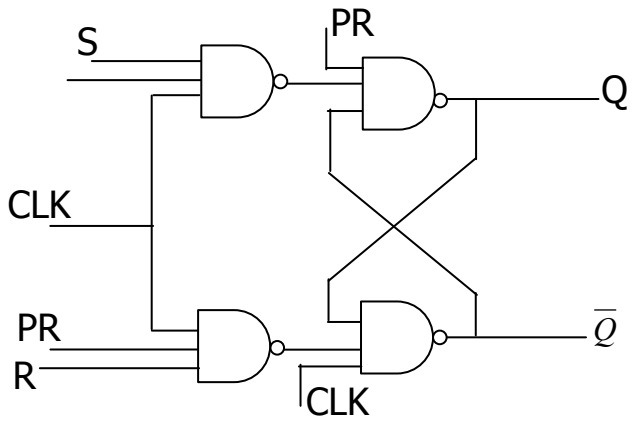
CLK	R	S	Q_{n-1}	Q'_{n+1}
۰	*	*	Q_n	Q'_n
۱	۰	۱	۱	۰
۱	۱	۰	۰	۱
۱	۱	۰	۰	۱
۱	۱	۱	۱	۱

نکته : کلاك يك سيگنال پريوديك مي باشد.

نکته : خروجي به خروجي هاي قبلي و به ورودي ها و هم به سيگنال زمانبندي بستگي دارد .

توسعه فيليپ فلاپ با كلاك :

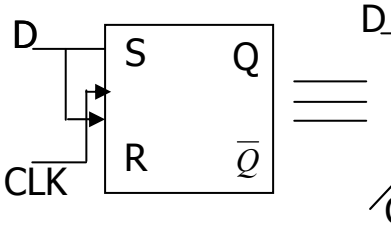


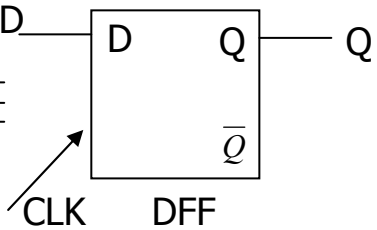


CLK	PR	CLK	R	S	Q_{n-1}	Q'_{n+1}
*	۰	۱	*	*	۱	۰
*	۱	۰	*	*	۰	۱
*	۰	۰	*	*	۱	۱
۰	۱	۱	*	*	Q_n	Q'_n
⎍	۱	۱	۰	۰	Q_n	Q'_n
⎍	۱	۱	۰	۱		
⎍	۱	۱	۱	۰		
⎍	۱	۱	۱	۱		

نکته : ورودی آسنکرون PR (CLR) بدون در نظر کلاک (زمانبندی) یا به صورت غیر همزمان روی خروجی تاثیر می گذارد .

D-FF : کاربرد در طراحی مدارهای ترتیبی و خصوصاً در رجیسترها :



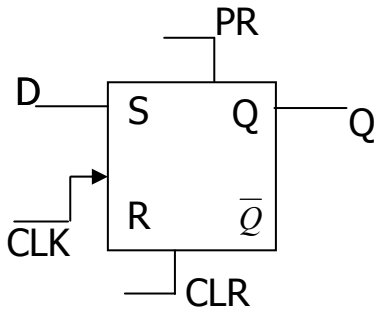


CLK	D	Q_{n-1}	وضعیت
L	*	Q_n	خط وضعیت
⎍	0	۰	Reset
⎍	1	۱	Set
⎍	D	Q_n	$Q_{n+1}=D$

مزایا :

۱- برای Set و Reset و ... از یک ورودی استفاده می کنیم .

۲- حالت غیر مجاز را از بین برده است .

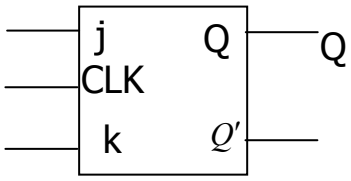


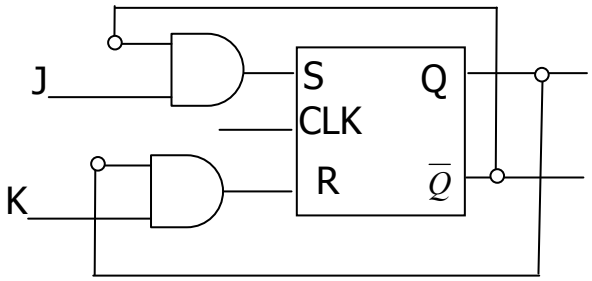
PR	CLK	CLD	D	
۰	۱	*	*	۱
۱	۰	*	*	۰
۰	۰	*	*	* غیر مجاز
۱	۱	⎍	D	D
۱	۱	۱	*	Q_n

Jk-ff : کاربرد در طراحي مدارهاي ترتيبي و مخصوصاً در شمارنده ها .

PR	CLK	CLD	Q_{n+1}	Q'_{n-1}	وضعیت
L	*	*	Q_n	Q'_n	Hold
⎍	1	0	1	0	Set
⎍	0	1	0	1	Reset
⎍	0	0	Q_n	Q'_n	Hold
⎍	1	1	Q'_n	Q_n	Toggle (NOT)

www . bme711 . ir

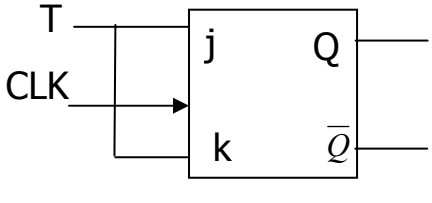


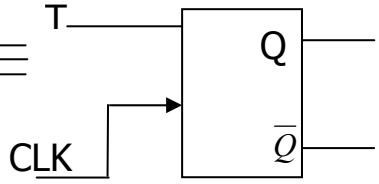


KQ_n / J	۰۰	۰۱	۱۱	۱۰
۰	۰	۱	۰	۰
۱	۱	۱	۰	۱

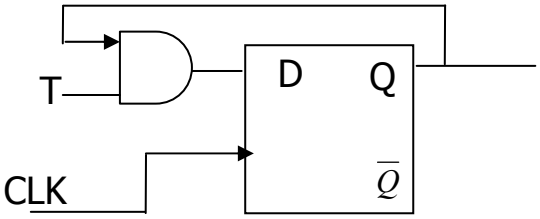
$$Q_{n+1} = Q_n k' + JQ'_n$$
$$Q'_{n+1} = J'Q'_n + kQ_n$$

T-FF : JK-FF اي است كه j و K آن به هم به هم متصل است .كاربرد و شمارنده ها .





CLK	T	Q_{n+1}	وضعيت
۰	x	Q_n	Hold
⎍	۰	Q_n	Hold
⎍	۱	Q'_n	toggle



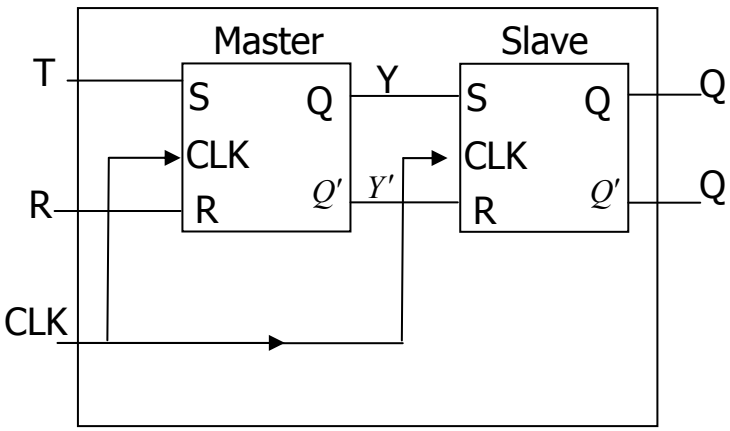
$$Q_{n+1} = T'Q_n + TQ'_n = T \oplus Q_n$$

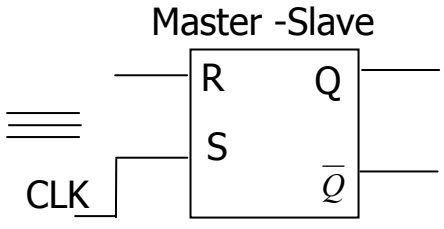
نكته : در سطح مادامي كه CLK وجود دارد و k و j هـر دو يـك باشـند و JKFF حـساس به سطح باشد خروجي نوسان مي كند براي رفع اين مشكل :

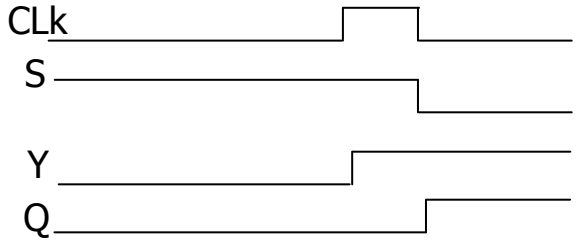
۱- استفاده از JK-FF حساس به لبه

۲- با استفاده از Master-Slave . F.F آنرا JK تبديل به حساس به لبه مي كند .

: Master-Slave . F.F

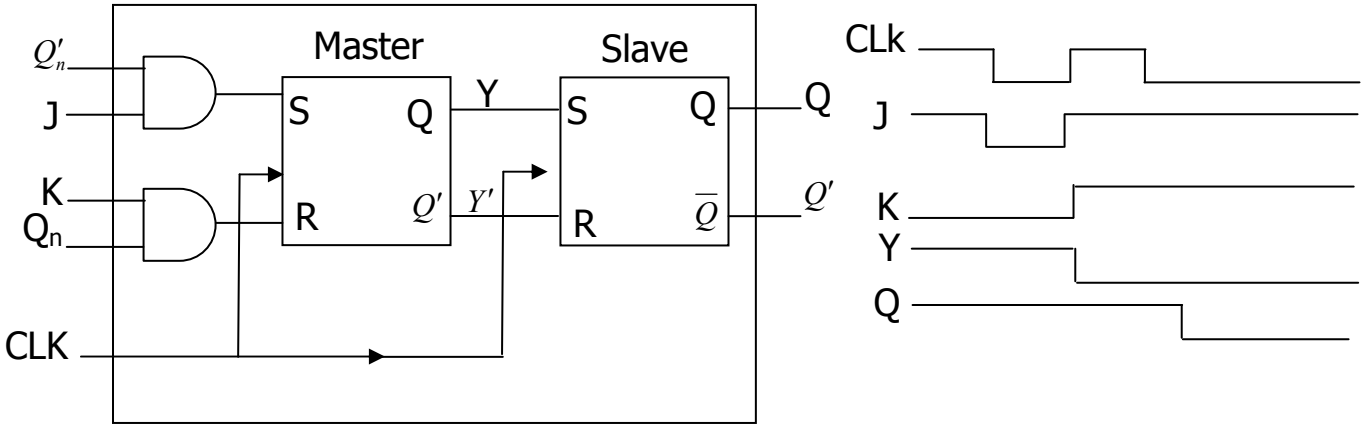






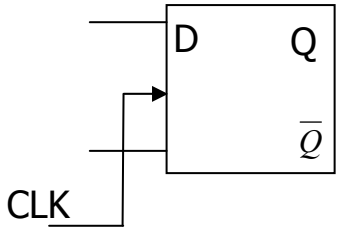
CLK	Master	Slave
Low	disActive	Active
High	Active	disActive

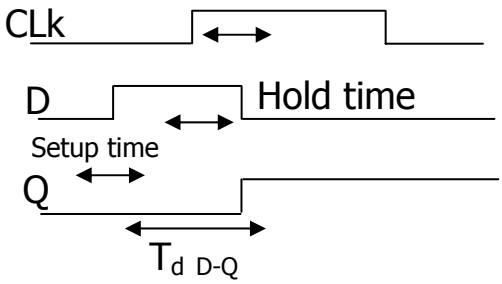
R = "C"



و بدین ترتیب مشکل حل شد .

تاخیر



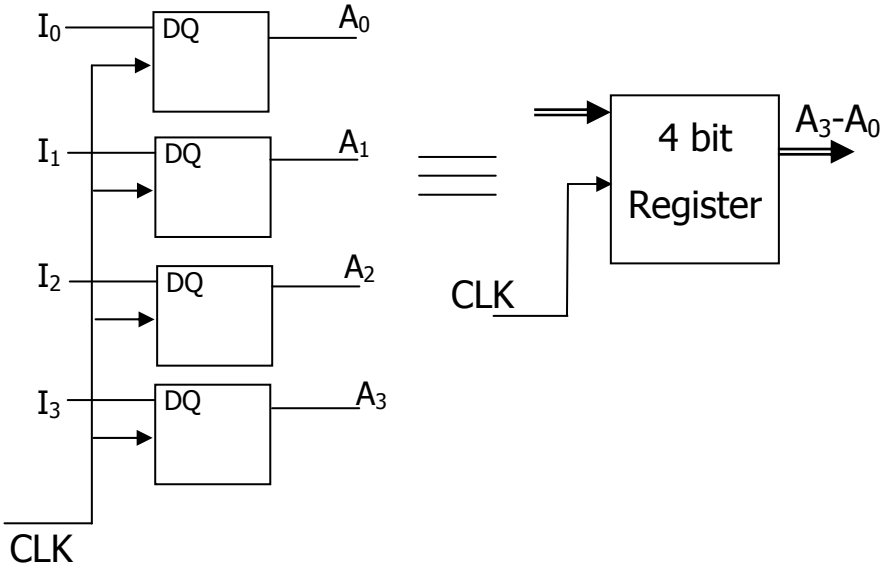


نکته : باید کمی قبل از آمدن CLK «1» شـود تـا بتواننـد تغییـرات لازم را در مـدار بدهـد (آماده باشـد) که به این زمان Setup time گویند (حداقل زمان کمی می بایسـت قبـل از آمدن به کلاک ورودی (Stable بماند) همچنین بـرای انجـام تغییـرات لازم در مـدارات داخلی D نباید همزمان با آمدن CLK صفر شود بلکه باید کمی بگذرد که بـه ایـن مـدت

Hold time گویند (= مدت زمانی که لازم است تا ورودی D پس از آمدن به کلاک ثابت بماند تا تغییر مناسب در خروجی صورت گیرد .

رجیستر ها یا ثباتها :

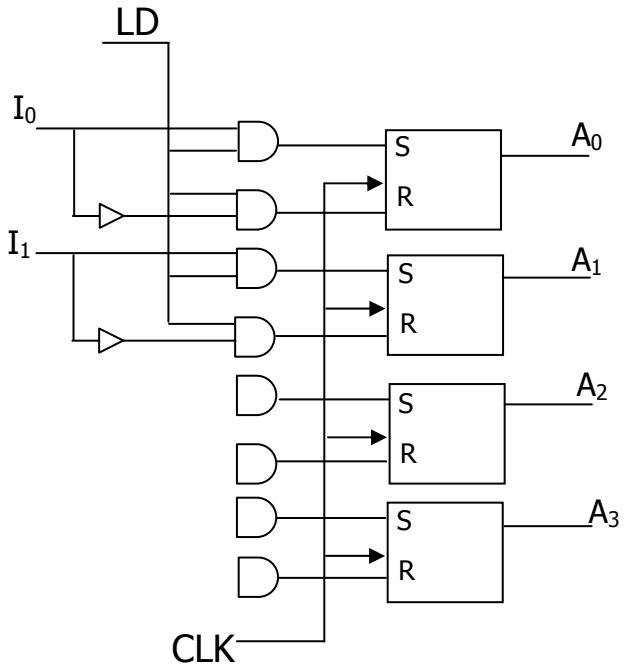
به مجموعه ای از FF ها که با کلاک مشترک کار می کنند و برای ثبت اطلاعات دودویی بکار می روند ثبات یا ریجستر گویند .

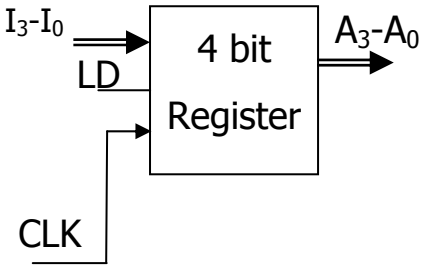


CLK	in puts	Out puts
L	x	خروجيهاي قبلي
↓	I3-I0	I3-I0

نکته : اگر D-FF حساس به لبه باشد کلاک ورودی ، CLK نام دارد و اگر حساس به سطح باشد ورودی کلاک G (گیت) نام دارد .

نکته : چون بایستی CLK به کلاک سیستم ، هیچ کنترلی روی ثبات ندایم (زیرا کلاک سیستم تواماً می آید) لذا از یک ورودی دیگر به نام LD (Laod) استفاده می کنیم که به چند تو مایه های Enable است .

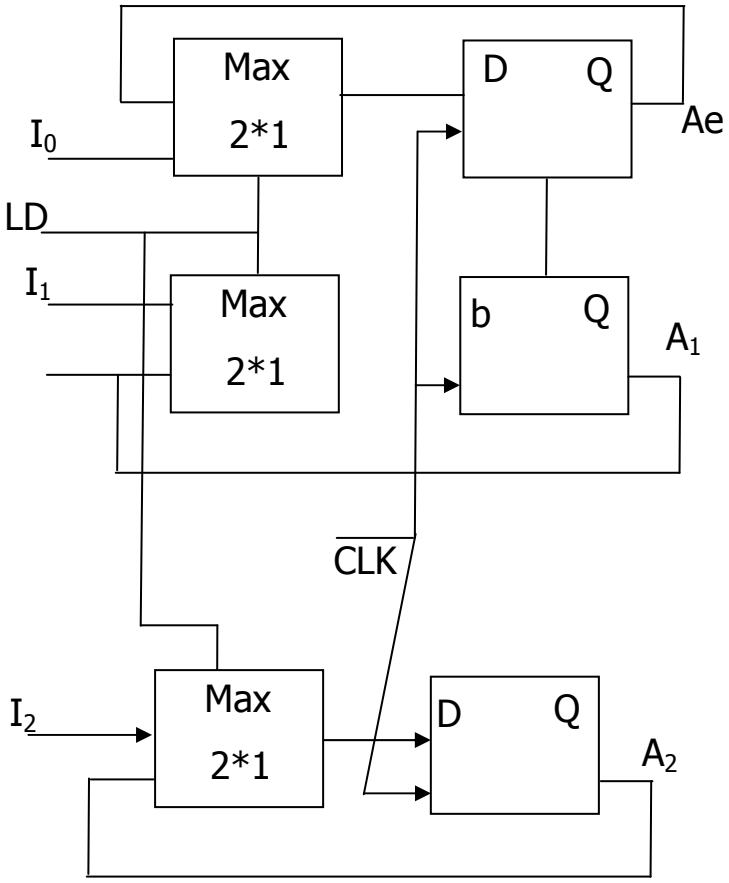


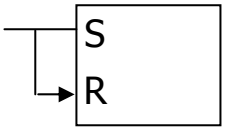


LD	CLK	in puts	Out puts
o	x	x	Rfgd
۱	↓	I_3-I_0	I_3-I_0

LD	I_0	CLK	A_0
۰	x	↓	قبلي
۱	۰	↓	۰
۱	۱	↓	۱

اگر ثبات ۴ بیتی با کمک D-FF و با ورودی LD :





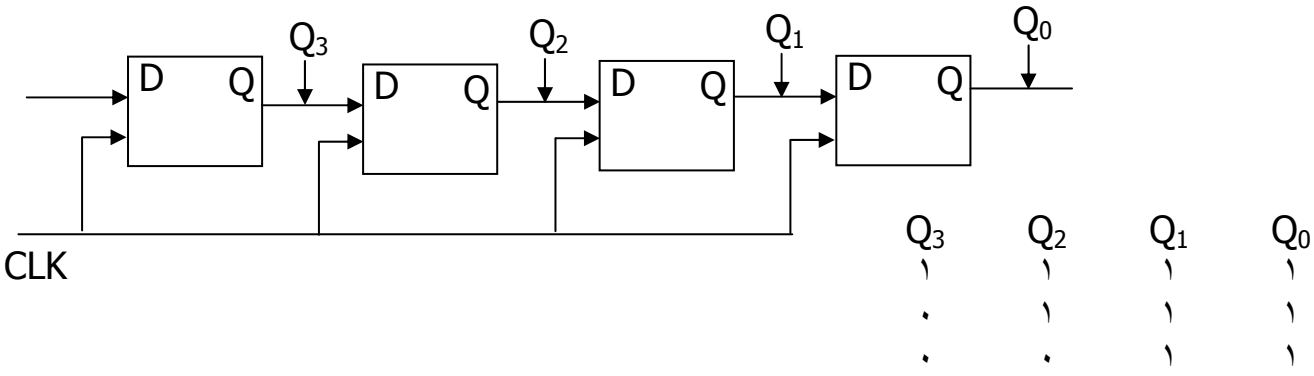
شیفت رجیسترها :

برای جابه جایی اطلاعات باینتری (شیفت) به سمت چپ یا راست.

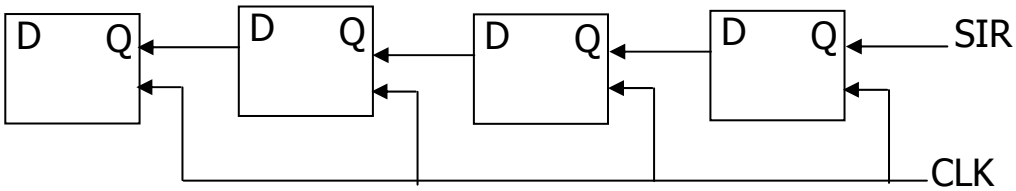
کاربرد : ضرب ، تقسیم ، جابه جائی اطلاعات .

۱۰	1	0	1	0
۵	0	1	0	1
۱۰	1	0	1	0

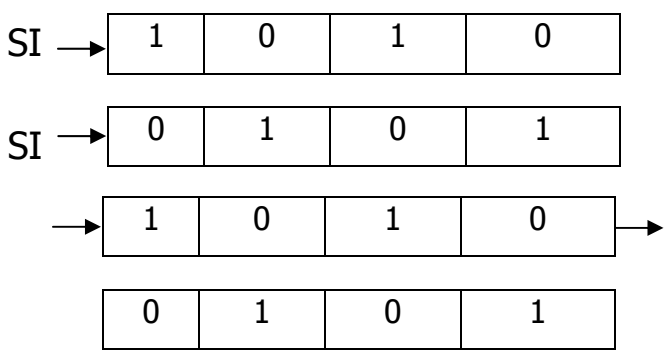
مثال :



مدار شیفت رجیستر شیفت به چپ :



تقسیم بندی با توجه به ورودی و خروجی :

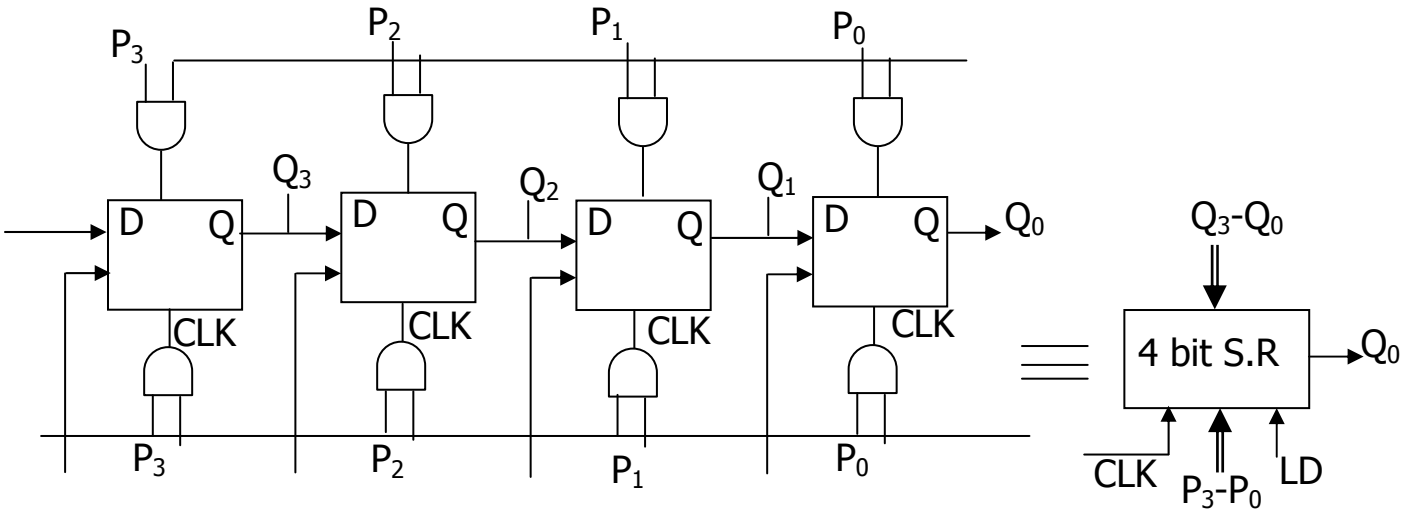


۱- ورودی سری – خروجی سری SI/SO

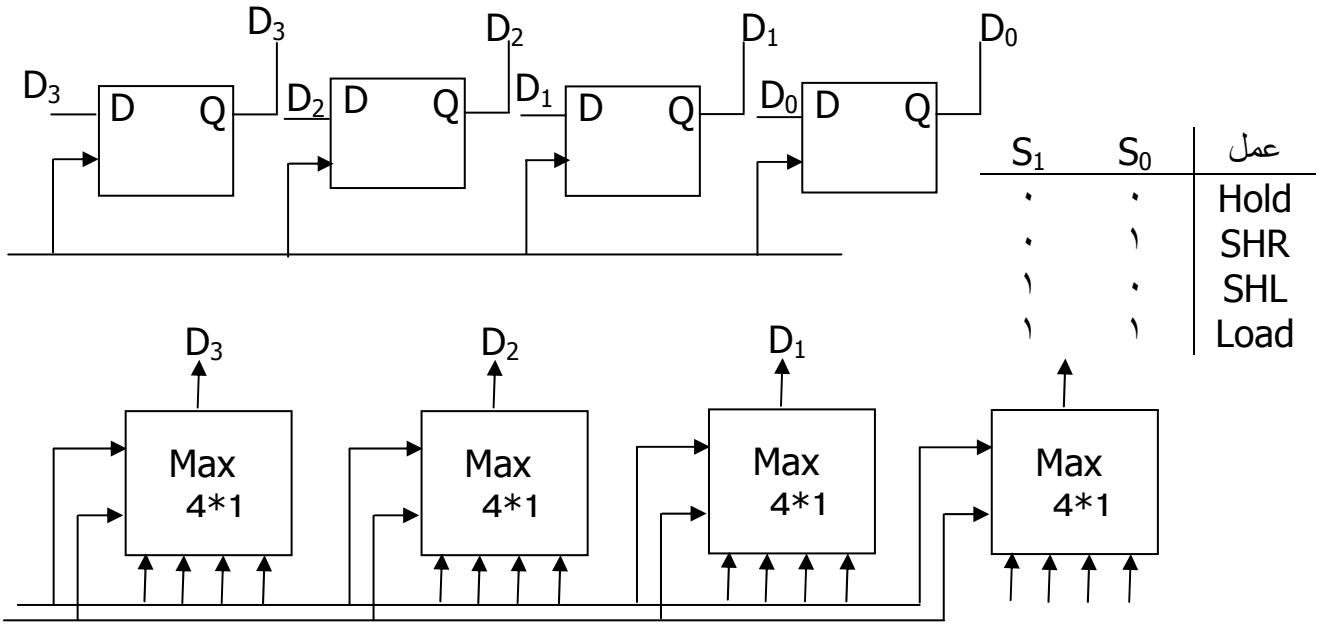
۲- ورودی سری – خروجی موازی SI/PO

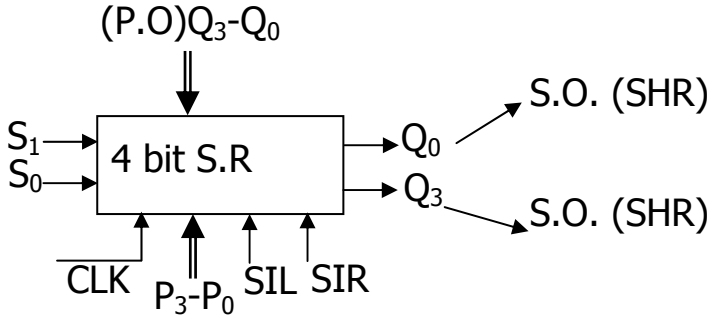
۳- وودی موازی – خروجی سری PI/SO

۴- ورودی موازی – خروجی موازی PI/PO



هدف : طراحی شیفت رجیستری که با دو خط انتخاب S_1 , S_A اعمال باردهی ، جابه جایی به چپ یا راست به همراه نگهداری اطلاعات را انجام می دهد . که به آن شیفت رجیستری عمومی یا همه منظوره گفته می شود .





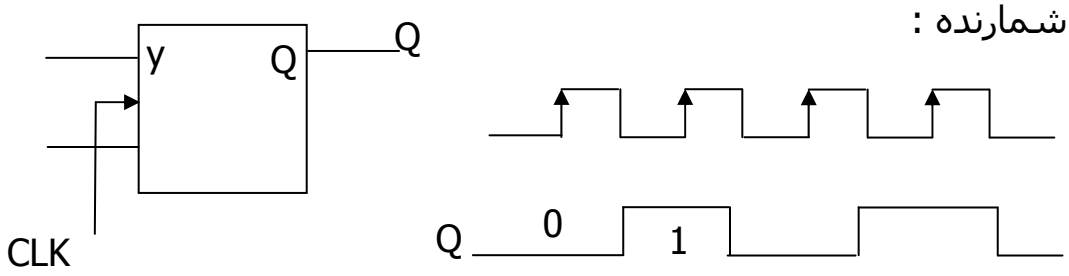
www . bme711 . ir

شمارنده ها :

درحالت کلی : ۱- سنکرون یا همزمان : تغییر خروجی تمام F.F ها با تاخیر ثابتی نسبت به CLK صورت می گیرد .

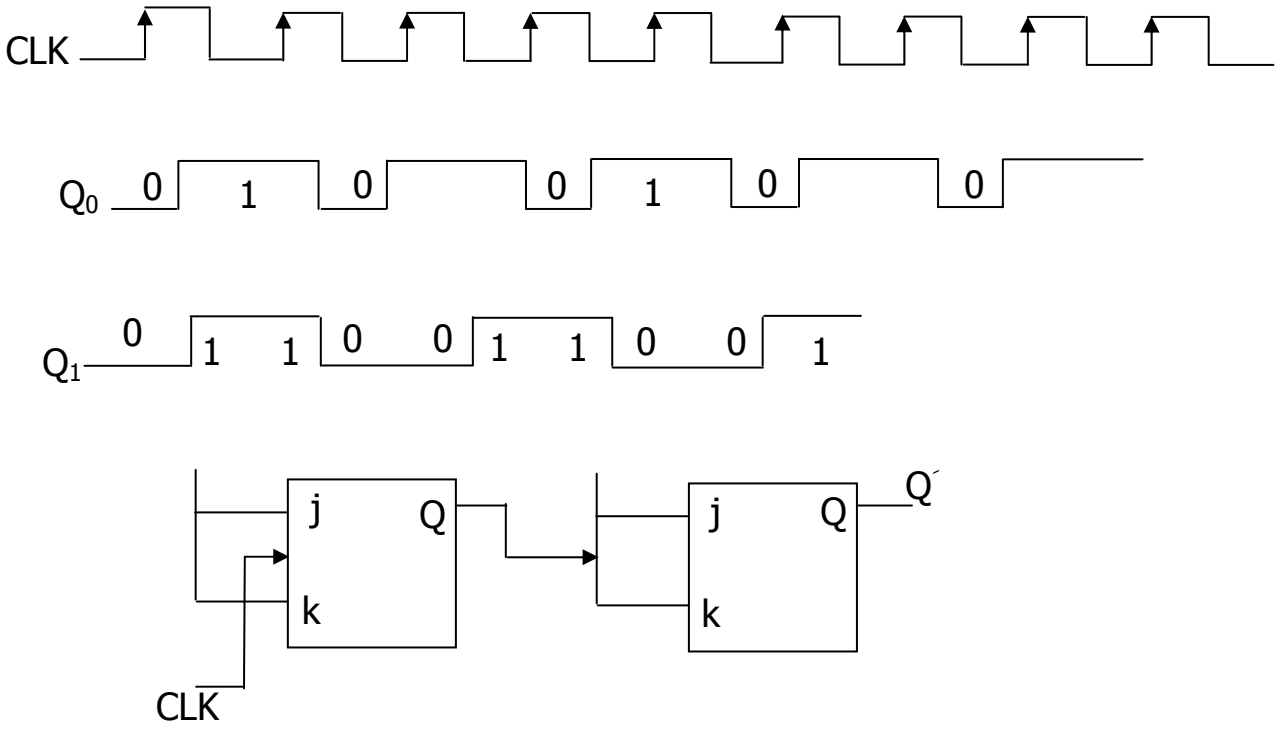
۲- آنسکرون یا غیر همزمان یا ضربان : تغییر خروجی تمام F.F ها با تاخیر متغیر (ثبت به CLK) صورت می گیرد .

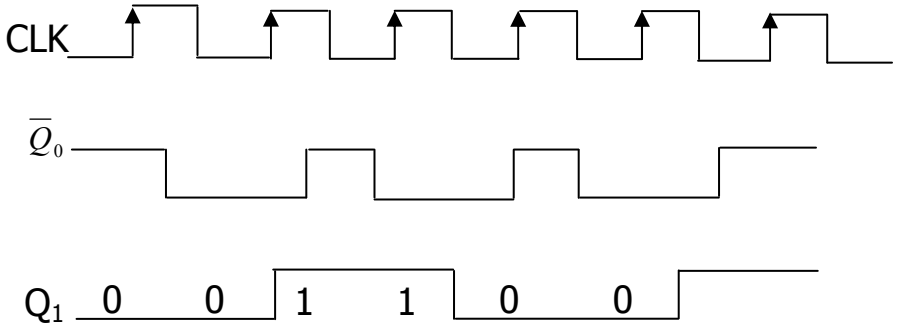
ساده ترین شمارنده :

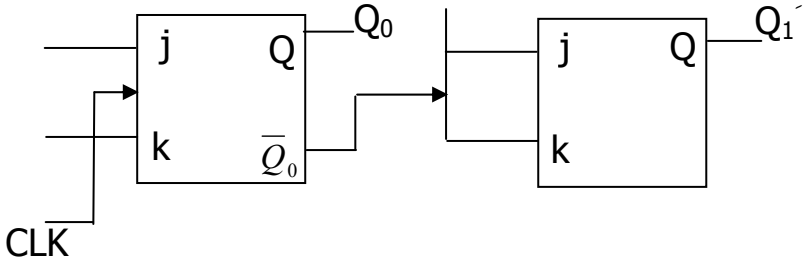


خروجی تقسیم بر 2 $F_Q = \frac{f_{CLK}}{2}$

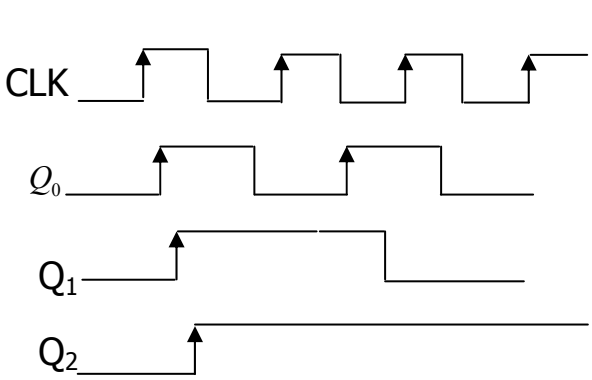
توسعه شمارنده :



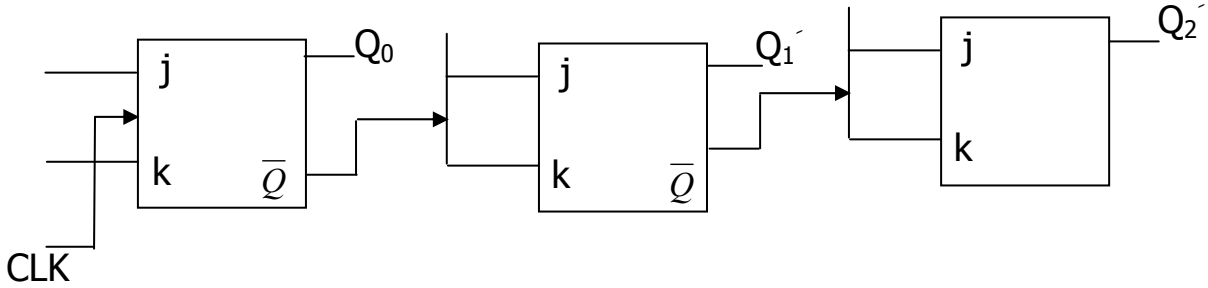


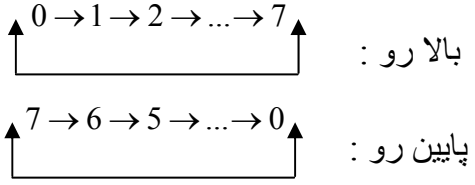


شمارنده آسنکرون



$Q_1 - Q_0$	$0 \to 1$	$2 \to 3$
0	0	
0	1	
1	0	
1	1	

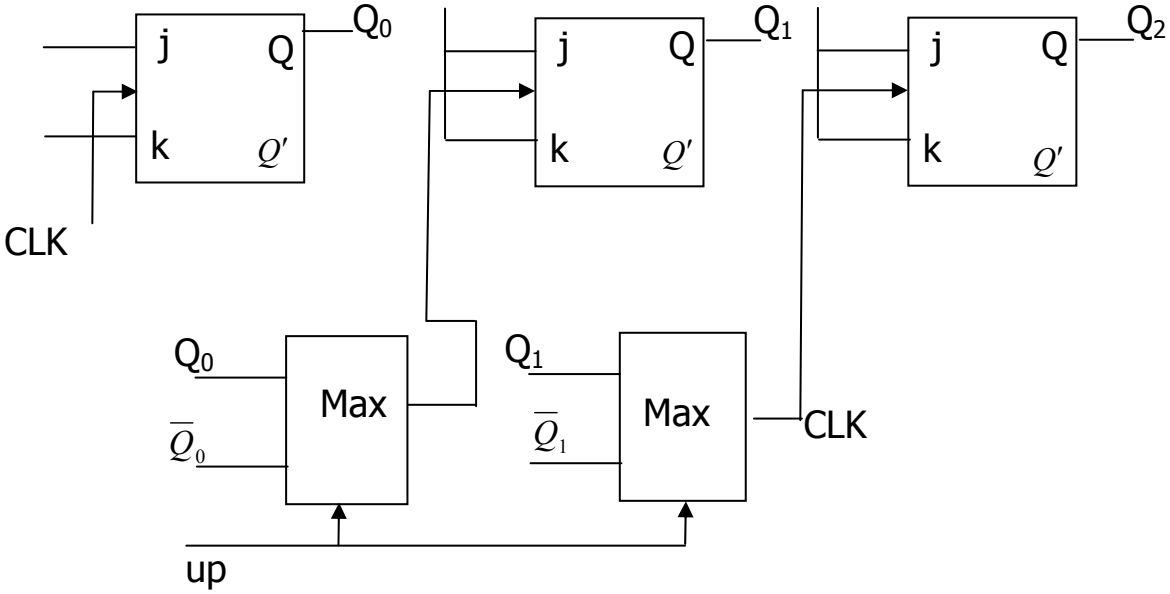




بالا رو :

پایین رو :

مثال : شمارنده آسنکرونی طراحی کنید که با یک خط کنترل به سمت بالا یا پایین شمارش کند .



نکته :

- در 2^n لزومی ندارد که ورودیهای k,j حتماً به ۱ وصل می شود .

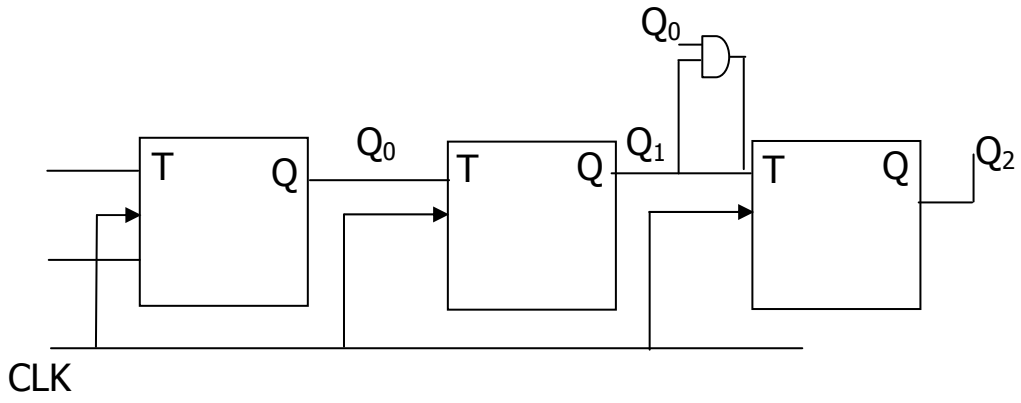
- برای شمارنده های آنسکرون غیر 2^n روش خاصی وجود ندارد .

نحوه شمارش	خروجیها	ورودیهای
پایین رو	Q ها	Q ها
بالا رو	$\overline{Q}$ ها	Q ها
بالا رو	Q ها	$\overline{Q}$ ها
پایین رو	$\overline{Q}$ ها	$\overline{Q}$ ها

شمارنده سنکرون :

Q_2	Q_1	Q_0	
۰	۰	۰	۰
۰	۰	۱	۱
۰	۱	۰	۲
۰	۱	۱	۳
۰	۰	۰	۴
۱	۰	۱	۵
۱	۱	۰	۶
۱	۱	۱	۷
۰	۰	۰	۸

www . bme711 . ir



نکته :

فقط تاخیر کلاک تا خروجی را داریم . چون این AND قبل از آمدن کلاک خروجی اش را آماده می کند .

$$\text{Max } F_{CLK} = \frac{۱}{\text{تاخیر AND} + \text{تاخیر FF}}$$

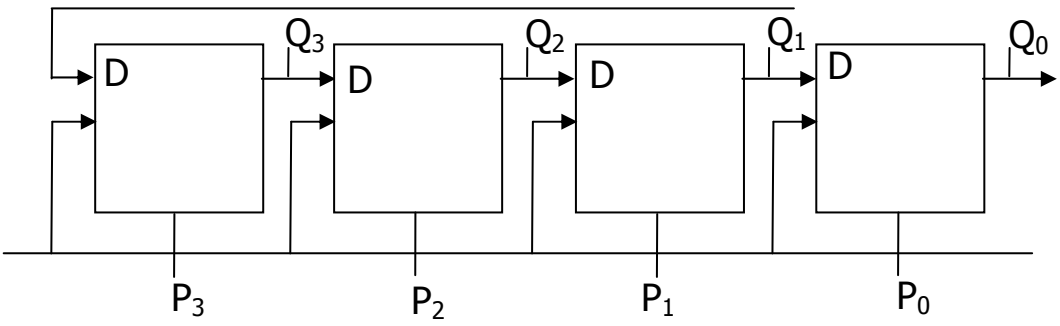
ولی برای آنسکرون : $Max\,F_{CLK} = \frac{1}{ntd}$

مثال : اگر $FF_{td} = 20_{ns}$ باشد آنگاه بیشینه ی F_{CLK} چقدر است ؟ (برای شمارنده ی آنسکرون ۴ بیتی)

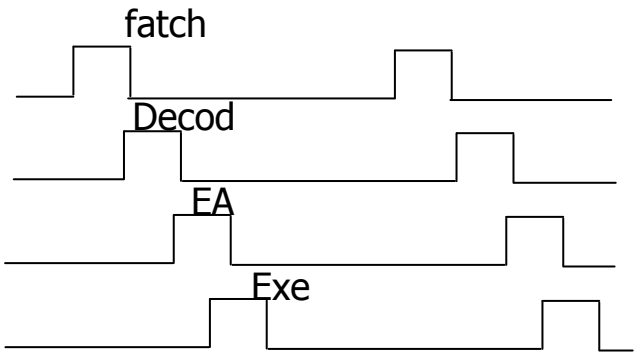
$$td_{tot} = 4 * td = 80_{ns} \qquad \rightarrow F_{\max} = 12.5\,MH$$

شمارنده های خاص :

۱- شمارنده حلقوی



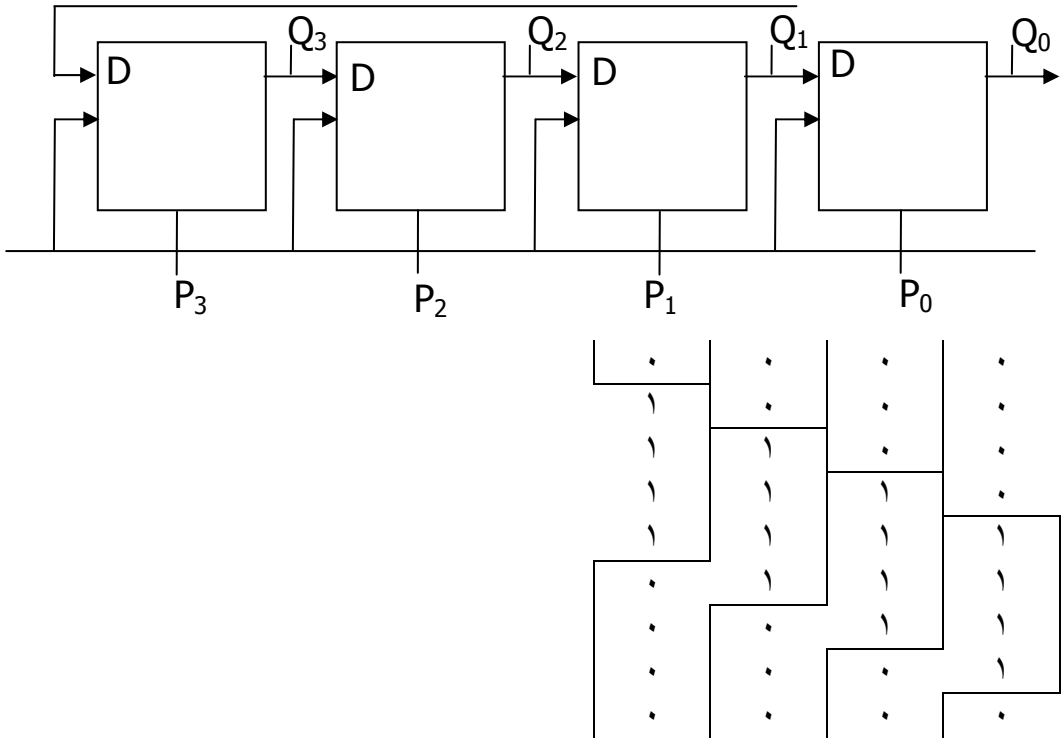
www . bme711 . ir



	P_3	P_2	P_1	P_0
عمل Load	۱	۰	۰	۰

	Q_3	Q_2	Q_1	Q_0
ترتیب شمارش :	۱	۰	۰	۰
	۰	۱	۰	۰
	۰	۰	۱	۰
	۰	۰	۰	۱
	۱	۰	۰	۰

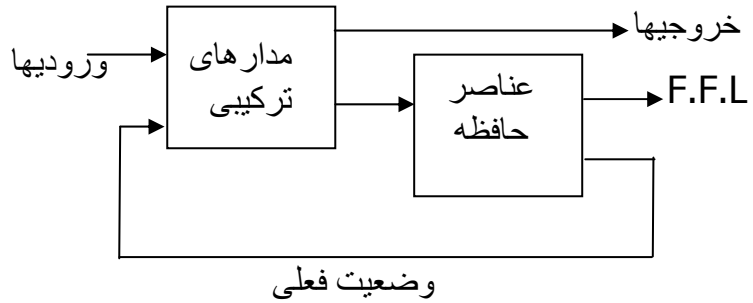
۲- شمارنده ی جانسون :



۰	۰	۰	۰
۱	۰	۰	۰
۱	۱	۰	۰
۱	۱	۱	۰
۱	۱	۱	۱
۰	۱	۱	۱
۰	۰	۱	۱
۰	۰	۰	۱
۰	۰	۰	۰

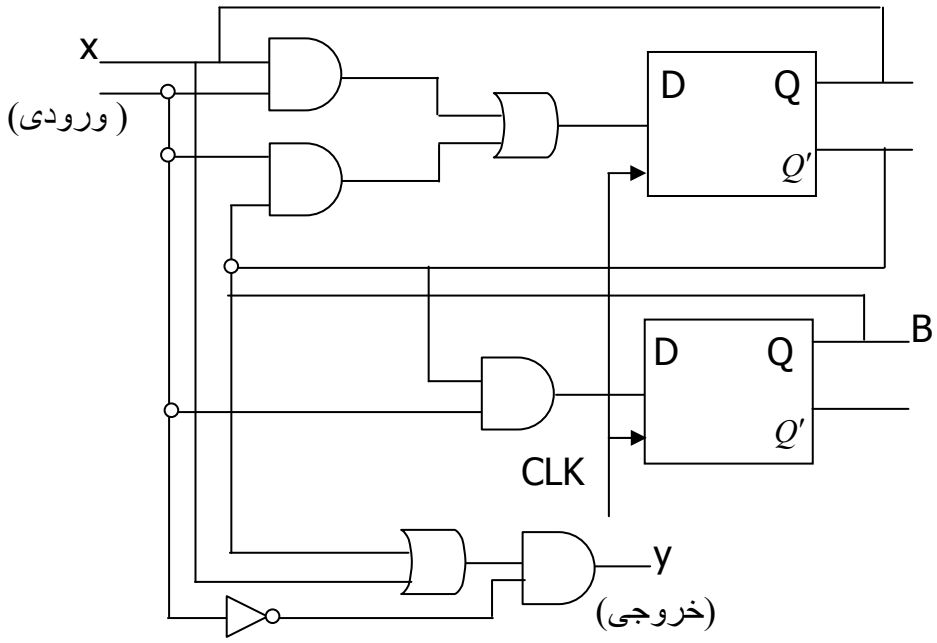
طراحی مدارهای منطقی ترتیبی :

ساختار کلی مدارهای ترتیبی :



مثال از مدار منطقی ترتیبی :

$$A(t+1) = A(t)x(t) + B(t)x(t)$$
$$B(t+1) = A'(x)x(t)$$
$$y = (A+B)x'$$
$$\downarrow \quad x'(t)$$
$$A(t)$$



جدول حالت :

ترتیب زمانی ورودیها ، خروجیها و وضعیت FF ها را در جدولی بنام جدول حالات می توان بیان نمود . این جدول شامل قسمتهای حالت فعلی ، ورودی ، حالت بعدی و خروجی است .

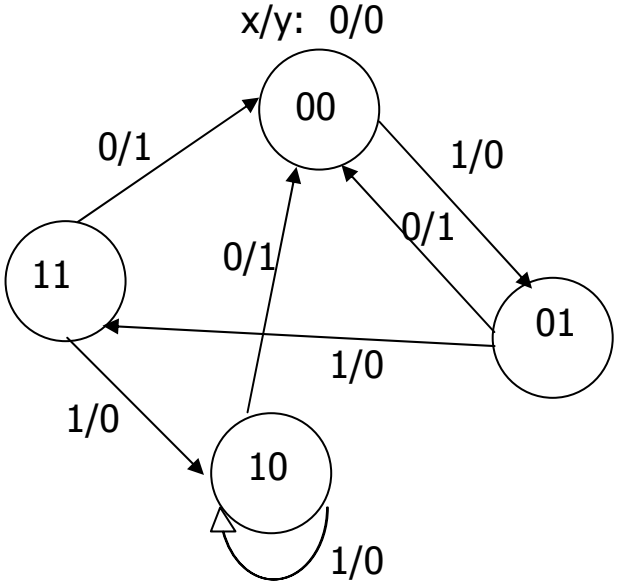
جدول حالت مثال بالا :

حالت فعلی		ورودی	حالت بعدی		خروجی
A	B	x	A	B	y
۰	۰	۰	۰	۰	۰
۰	۰	۱	۰	۱	۰
۰	۱	۰	۰	۰	۱
۰	۱	۱	۱	۱	۰
۱	۰	۰	۰	۰	۱
۱	۰	۱	۱	۰	۰
۱	۱	۰	۰	۰	۱
۱	۱	۱	۱	۰	۰

طراحی جدول به روش دیگر :

A	B	x	A		B		y	
		x=0	x=1		x=0		x=1	
۰	۰	۰	۰	۱	۰		۰	
۰	۱	۰	۱	۱	۱		۰	
۱	۰	۰	۱	۰	۱		۰	
۱	۱	۰	۱	۰	۱		۰	

دیاگرام حالت (State Diagram) :

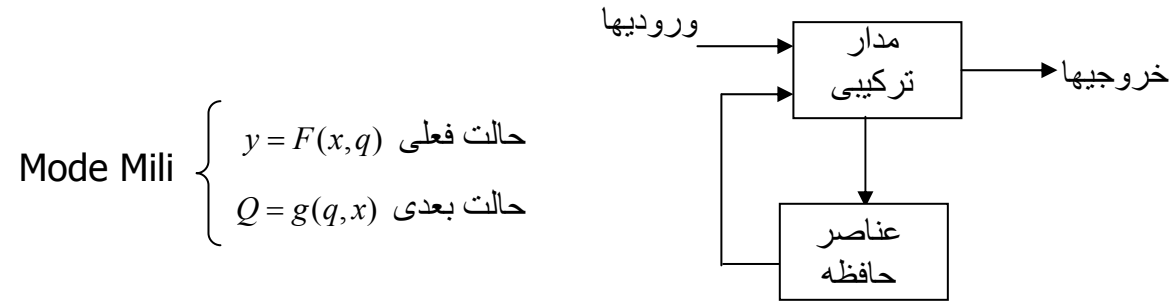


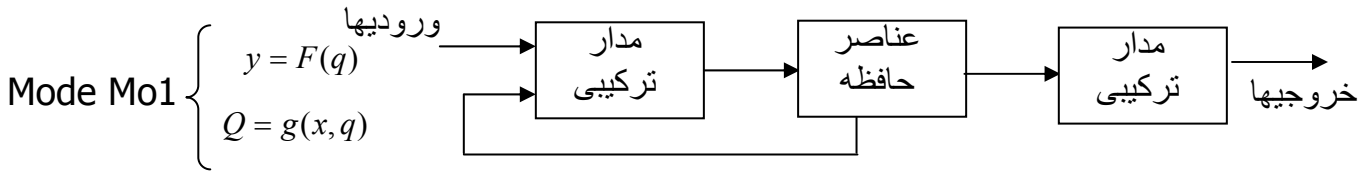
انواع مدارهای ترتیبی :

مدهای میلی و مور :

۱- مد میلی : خروجی بر اساس حالت فعلی ورودیها مشخص می شود .

۲- مد مور : خروجی فقط از روی حالات فعلی مشخص می شود ، به صورت مستقیم به ورودی ربط ندارد یا در رابطه ی خروجی ورودیها دیده نمی شود .

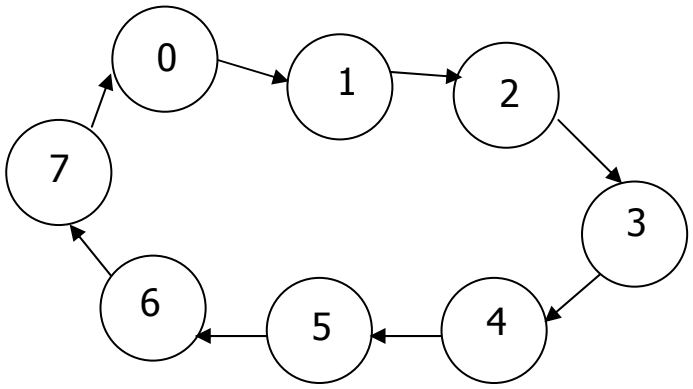




نکته : در مد میلی اگر ورودی تغییر کند حتی بدون آمدن CLK ممکن است که خروجی تغییر کند ولی در مد مور چنین نیست لذا برای رفع این مشکل ورودیهای مد میلی را با CLK همزمان می کنند .

نکته : مدار صفحه ی قبلی مثالی از یک مد و شمارنده های مثالی از مد مور هستند .

نکته : در مد مور نمایش State Diagram تنها برحسب ورودی است یعنی :



جدول تحریک F.F ها :

S.R را چه در نظر بگیریم تا به حالت Q ها برسیم .

Q(t+1)	Q(t+1)	S	R	D	J	K	T
۱	۰	۰	x	۰	۰	x	۰
۰	۱	۱	۰	۱	۱	x	۱
۱	۰	۰	۱	۰	x	۱	۱
۱	۱	x	۰	۱	x	۰	۰

www . bme711 . ir

مراحل طراحی مدارهای ترتیبی :

۱- توصیف لفظی عملکرد مدارد : ۱- دیاگرام حالت ۲- دیاگرام زمانی

۲- با توجه به اطلاعات مفروض در مورد مدار جدول حالات تنظیم شود .

۳- تعداد حالات را در صورت امکان کاهش دهید .

۴- اگر درجدول حالات ، سمبل های حرفی وجود دارد ، آنها را با مقادیر دودویی جایگزین کنید (تشخیص حالت)

۵- تعداد F.F ها را مشخص کنید (با توجه به جدول حالات) و به هر کدام سمبل حرفی تخصیص دهید .

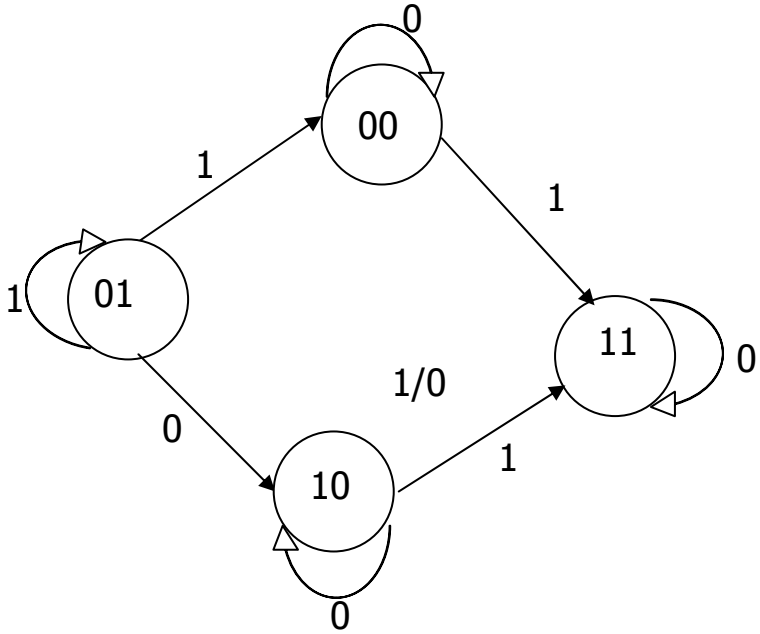
۶- نوع F.F ها را مشخص کنید (معمولاً در صورت مساله)

۷- باتوجه به جدول حالات ، جداول تحریک و خروجی را بدست آورید .

۸- ساده سازی روابط خروجی ، ورودیهای تحریک از روی جدول خروجی و تحریک با کمک روشهای متعارف مثل روش کارنو .

مثال : دیاگرام حالت :

دیاگرام حالت زیر مفروض است . مدار ترتیبی لازم را به کمک jkff رسم کنید .



فعلی		ورودی	بعدی	
A	B	x	A	B
۰	۰	۰	۰	۰
۰	۰	۱	۰	۱
۰	۱	۰	۱	۰
۰	۱	۱	۰	۱
۱	۰	۰	۱	۰
۱	۰	۱	۱	۱
۱	۱	۰	۱	۱
۱	۱	۱	۰	۰

3 امکان ندارد

4 نداریم

5 دوتا FF

JKFF 6

www . bme711 . ir

J_A	K_A	J_B	K_B
۰	x	۰	x
۰	x	۱	x
۱	x	x	۱
۰	x	x	۰
x	۰	۰	x
x	۰	1	x
x	۰	x	۰
x	۱	x	۱

AB

A				
	۰	۰	۰	۱
	x	x	x	x

$: j_A = Bx'$

Bx

A				
	x	x	x	x
	۰	۰	۱	۰

$: k_A = Bx$

ox

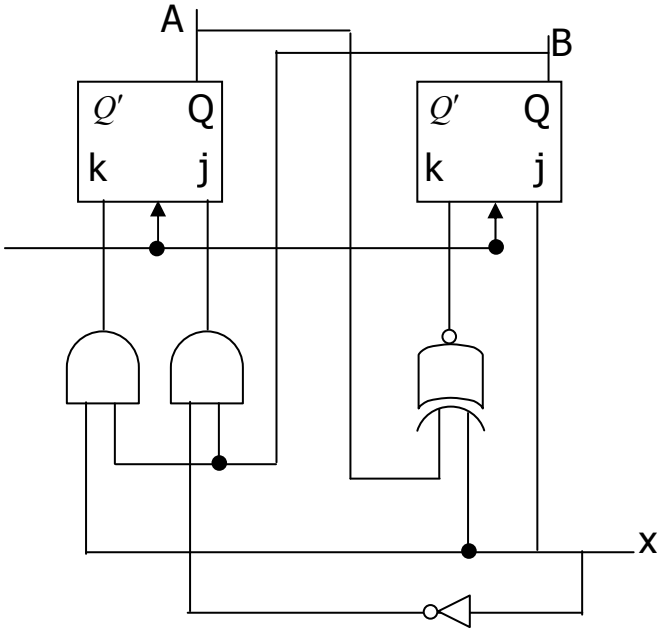
A				
	۰	۱	x	x
	۰	۱	x	x

$: j_B = x$

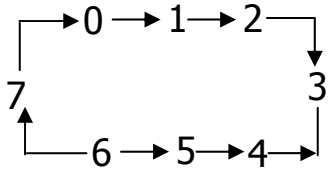
Bx

A				
	x	x	۰	۱
	x	x	۱	۰

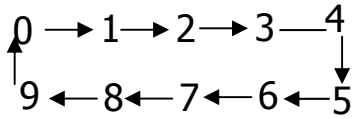
$: k_B = (A \oplus x)'$



تمرین : شما رنده اي طراحي كنيد كه ترتيب زير بشما رد با كمك T.F.F



مثال : شما رونده BCD با كمك T.FF :



(فعلي)				(بعدي)			
Q	Q	Q	Q	Q	Q	Q	Q
8	4	2	1	8	4	2	1
۰	۰	۰	۰	۰	۰	۰	۱
۰	۰	۰	۱	۰	۰	۱	۰
۰	۰	۱	۰	۰	۰	۱	۱
۰	۰	۱	۱	۰	۱	۰	۰
۰	۱	۰	۰	۰	۱	۰	۱
۰	۱	۰	۱	۰	۱	۱	۰
۰	۱	۱	۰	۰	۱	۱	۱
۰	۱	۱	۱	۱	۰	۰	۰
۱	۰	۰	۰	۱	۰	۰	۱
۱	۰	۰	۱	۰	۰	۰	۰

جدول تحريك :

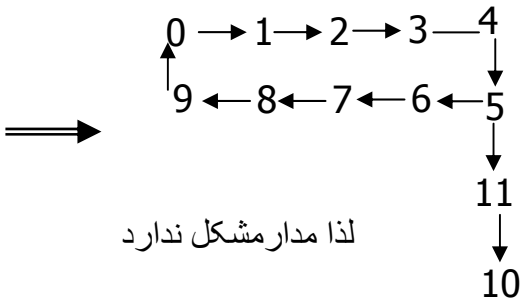
T_8	T_4	T_2	T_1
۰	۰	۰	۱
۰	۰	۱	۱
۰	۰	۰	۱
۰	۱	۱	۱
۰	۰	۰	۱
۰	۰	۱	۱
۰	۰	۰	۱
۱	۱	۱	۱
۰	۰	۰	۱
۱	۰	۰	۱
*	*	*	*

$$T_{Q_8} = Q_{D_8} Q_{A_1} + Q_{C_4} Q_{B_2} Q_{A_1}$$
$$T_{Q_4} = Q_2 Q_1$$
$$T_{Q_2} = Q_2 Q_1$$
$$T_{Q_1} = 1$$

نکته : اگر بعد از بستن مدار مقدار اولیه ۱۲ یا ۱۳ و ... بود چون پیش بینی نشده اند لذا مدار به درستی عمل نخواهد کرد لذا باید این حالات را بررسی کرد .

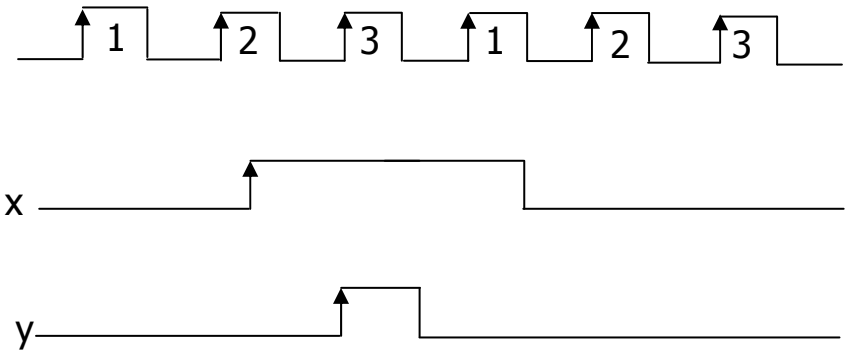
Q_8	Q_4	Q_2	Q_1
۱	۰	۱	۰
۱	۰	۱	۱
۱	۱	۰	۰
۱	۱	۰	۱
۱	۱	۱	۰
۱	۱	۱	۱

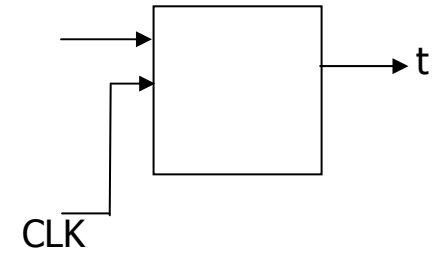
Q_8	Q_4	Q_2	Q_1
۱	۰	۱	۱
۰	۱	۱	۰
۱	۱	۰	۱
۰	۱	۰	۰

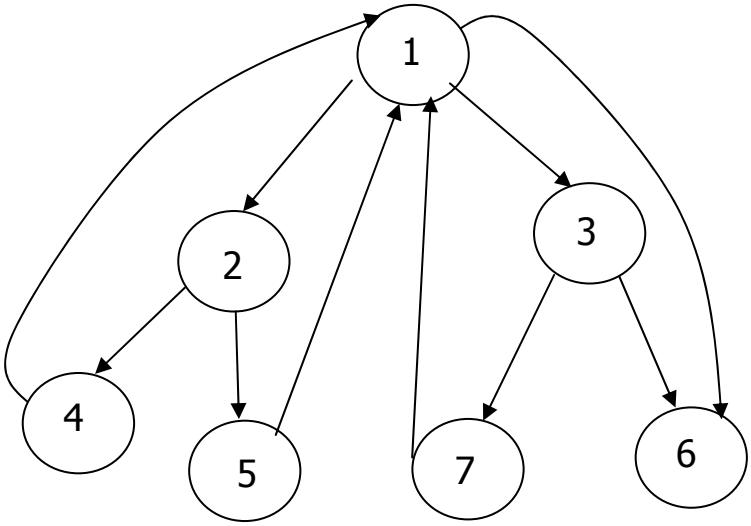


نکته : می توانستم از همان ابتدا مدار را بر این مبنا طراحی کنیم که اعداد Dont Care به صفر بروند .

مثالی از آخر : مداری طراحی کنید که به صورت پریودیک پالس ساعت ورودی x را بررسی و در صورت فرود بودن تعداد یکهای دریافتی ، خروجی یک شود .







جدول حالات :

حالتهاي مشابه :

به ازاي ورودیهاي یکسان

حالت بعدي و خروجي

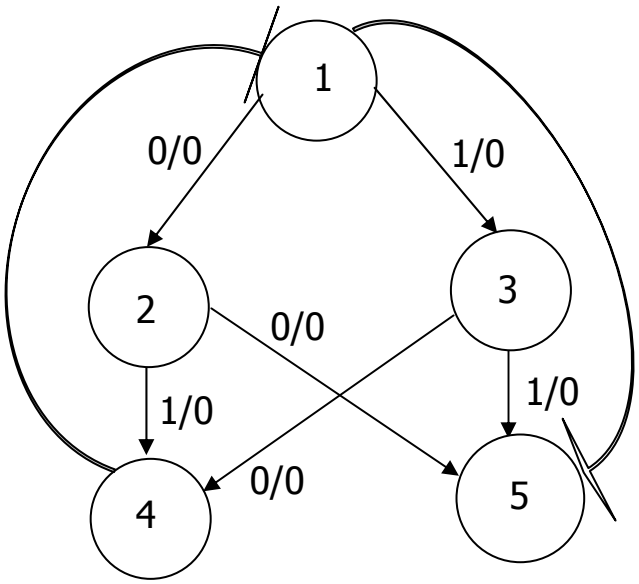
حالت فعلي	حالت بعدي		خروجي	
	x=0	x=1	x=0	x=1
1	2	3	0	0
2	5	4	0	0
3	6	7	0	0
4	1	1	1	0
5	1	1	0	1
6	1	1	1	0
7	9	1	0	1

www . bme711 . ir

یکسان نتیجه شود.

لذا 4 با 6 و 5 با 7 مشابه هستند در نتیجه مي توان گفت 6 و 7 زيادي هستند .

: Up dateed State Diagram



تخصيص حالات : سعي مي شود از يك حالت به حالت ديگر مجاور منطقي در جدول كار نو باشند .

	حالت فعلي			بعدي		خروجي	
				x=0	x=1	x=0	x=1
1	۰	۰	۰	۰۰۱	۰۱۱	۰	۰
2	۰	۰	۱	۱۱۱	۱۰۱	۰	۰
3	۰	۱	۱	۱۰۱	۱۱۱	۰	۰
4	۱	۰	۱	۰۰۰	۰۰۰	۱	۰
5	۱	۱	۱	۰۰۰	۰۰۰	۰	۱
	A	B	C				

با كمك T.F.F

خلاصه ...

$$T_A = C$$
$$T_B = AB + \overline{C}x + \overline{A}C\overline{x}$$
$$T_C = A + \overline{C}$$
$$y = A\overline{B}\overline{x} + ABx$$