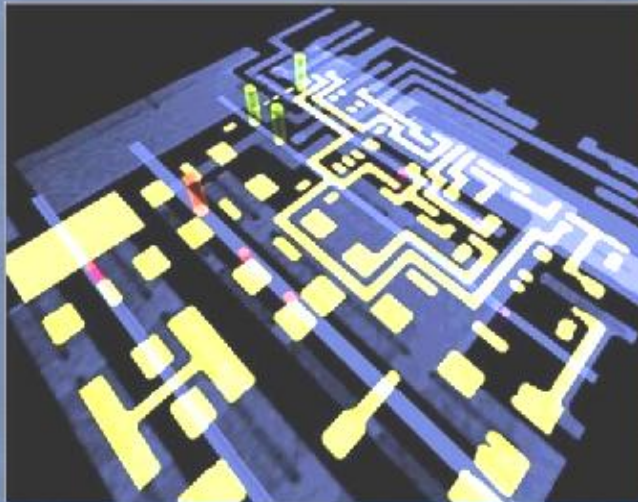


## مبانی الکترونیک دیجیتال جلسه هجدهم



### TTL

مشکلات خانواده ی DTL رو با جایگزین کردن دیود با ترانزیستور و تغییرات جزئی دیگر برطرف کردند ، به این خانواده ی جدید TTL می گویند.

**دلیل کند بودن خانواده ی DTL :**

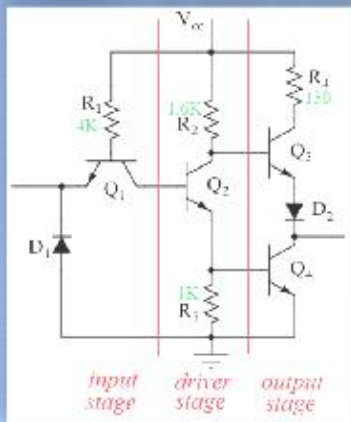
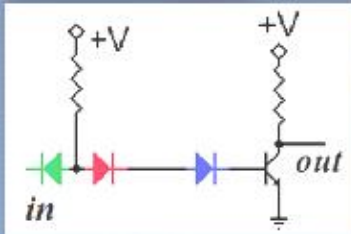
**ورودی کند:** بار موجود در بیس ترانزیستور خروجی باید در مقاومت  $5K$  تخلیه شود.  
**خروجی کند:** در صورت یک شدن خروجی، خازن خروجی باید از طریق مقاومت شارژ شود.

[http://www.play-hokey.com/digital/electronics/ttl\\_gates.html](http://www.play-hokey.com/digital/electronics/ttl_gates.html)

**مشخصات خانواده TTL**

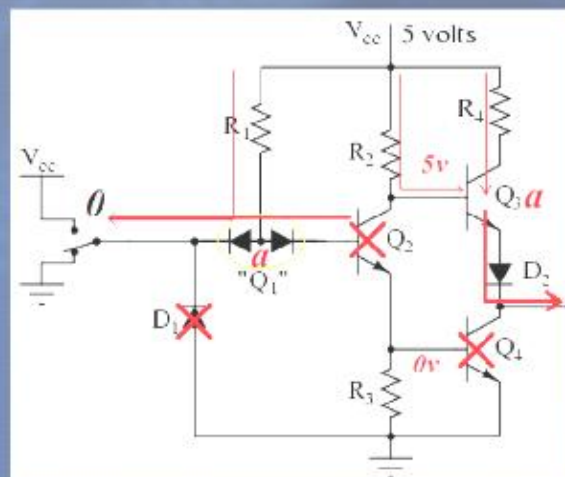
- $V_{cc} = 5v$
- Power : 11mw
- $T_p = 10ns$
- Noise Margin : 0.4v

## ساختار خانواده ی TTL

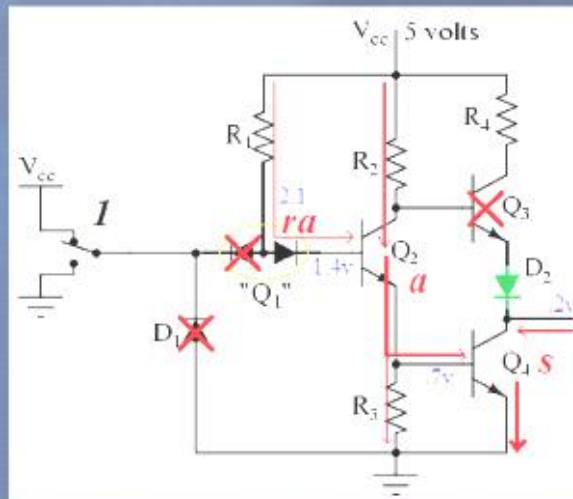


- به جای دیودهای سبز و قرمز رنگ که در DTL بودند، ترانزیستور Q1 جایگزین شد.
- دیود بیس-امیتر ترانزیستور Q2 به جای دیود آبی رنگ، حاشیه ی نویز را افزایش می دهد.
- برای افزایش سرعت شارژ خازن خروجی و کاهش مصرف توان استاتیک صفر، به جای مقاومت خروجی، ترانزیستور Q3 قرار گرفته.
- به این طبقه ی خروجی **totem pole** گفته می شود. برای تحریک این دو ترانزیستور مجبور به اضافه کردن طبقه ی میانی یا راه انداز هستیم.

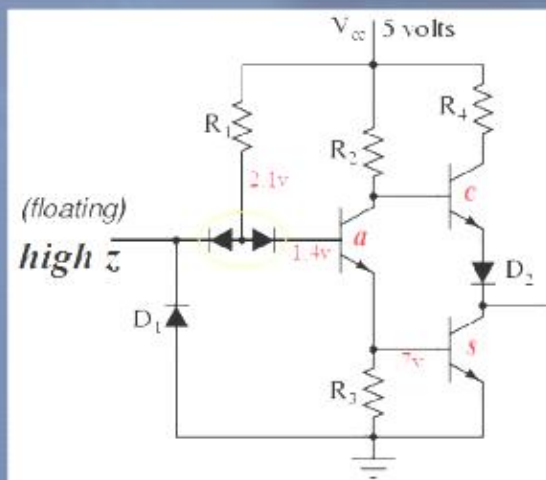
## TTL : not , in=0



TTL : not , in=1



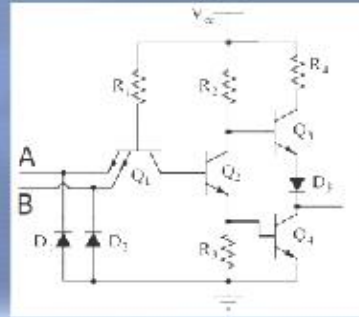
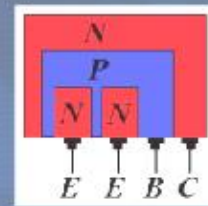
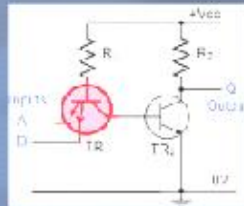
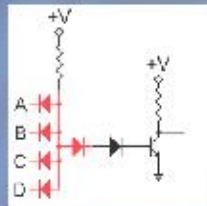
TTL : not , in=Z



## TTL : nand

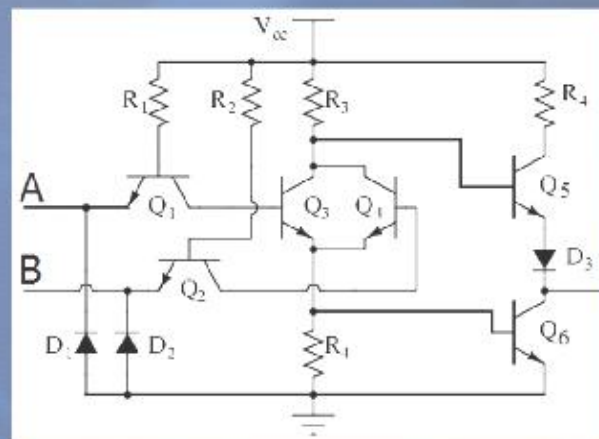
ترانزیستور BJT با چندین امیتر

- اگر جریان از بیس به سمت هر کدام از امیترها جاری شود، ترانزیستور روشن می شود.



## TTL : nor

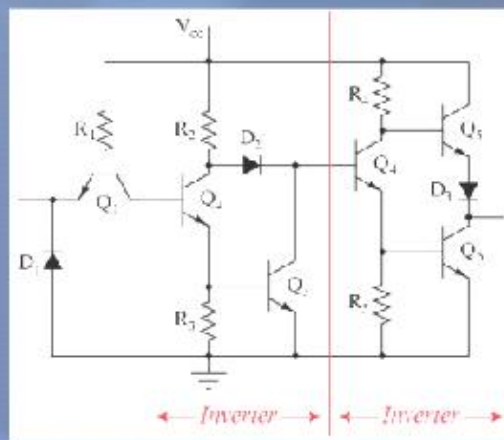
- در طبقه ی راه انداز تابع nor پیاده سازی می شود.
- برای پیاده سازی توابعی به فرم AOI چه کاری می توان کرد؟



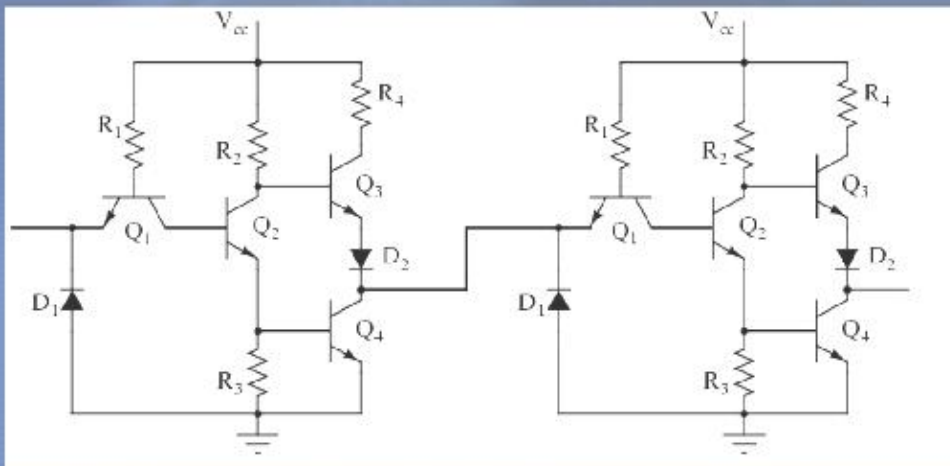


## TTL : buffer

- بافر از دو معکوس کننده ی پشت سر هم تشکیل شده.
- برای پیاده سازی توابع **or** و **and** نیز از همین روش استفاده می کنند.



## به هم بستن TTL



## جلسه آینده...

✓ گیت با خروجی کنترل شونده

✓ گیت با خروجی کلکتور باز

✓ آشنایی با خانواده ی ECL