

شکل ۱-۷ مدار جمع کننده-تفریق کننده تک بیتی

Full Adder (FA)

A	B	C _i	S	C _o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = C_i \text{ xor } A \text{ xor } B$$

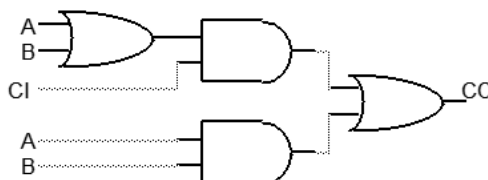
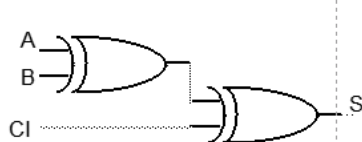
$$C_o = B C_i + A C_i + A B = C_i (A + B) + A B$$

C _i \ A B	00	01	11	10
0	0	1	0	1
1	1	0	1	0

S

C _i \ A B	00	01	11	10
0	0	0	1	0
1	0	1	1	1

C_o



30

با استفاده از یک Fulladder و یک گیت xor یک مدار جمع کننده - تفریق کننده ساختیم . در صورتی که ورودی sel یک باشد xor هر چیزی با یک نقیضش ($\sim b$) می شود در نتیجه مدار تفریق می کند . و بالعکس اگر sel صفر باشد xor هر چیزی با صفر خودش می شود در نتیجه مدار جمع می کند .

در ماژول time_calculate نیز بیت های time_out را از بیت های time_in کم کردیم .

ماژول 5 : در ماژول parking_capacity_counter ابتدا برای محاسبه ظرفیت خالی پارکینگ تمامی بیت های new_capacity را با هم جمع می کنیم (برای این کار نیاز به 17 بیت wire نیاز داریم : 6بیت اول برای محاسبه sum , carry جمع بیت های new_capacity ، و دوباره تمامی carry ها و تمامی sum ها را باهم جمع کرده و در wire های دیگر ذخیره می کنیم و در آخرین مرحله نیز جمع ها را وارد بیت های parked می کنیم)
برای محاسبه ظرفیت پر شده پارکینگ حاصل جمع بالا از عدد 1000 (معادل 8 که ظرفیت کامل پارکینگ است) تفریق میکنیم .

ماژول 7 :

A = ... inactive B = ..1 active C = .1. request

D = .11 save E = 1.. trap

خروجی‌های ماینر حالت با RegQ, RegP متغیر این در ورودی به عنوان enable به رجیسترهای ماینر می‌شوند و آن‌ها را فعال یا غیرفعال می‌کنند.

