برای نوشتن این مدار ما از 4 مالتی پلکسر یکسان و همچنین از 4 فول ادر یکسان استفاده کردیم.

مالتی پلکسر 4 به 1 صفر دارای سه ورودی B0 و S1 و S2 و همچنین سه تا وایر B0_not و Zero (برای ایجاد ورودی صفر) one (برای ایجاد ورودی یک) و خروجی Y0 است. همچنین برای not کردن B0 دستور خط دهم را مینویسیم.

با استفاده از کد های خط 12 تا 15 تعیین میکنیم که کدام خط مالتی پلکسر فعال شود و به خروجی فرستاده شود. یعنی هنگامی که S1 و S2 هر دو صفر باشند خط مربوط به B0، وقتی S1 صفر و S2 یک باشد خط مربوط به B0_not ، وقتی S1 یک باشد و S2 صفر خط مربوط به عدد صفر و هنگامی که S1 و S2 هر دو یک باشند خط مربوط به عدد صفر فعال خواهد شد.

و به همین ترتیب ماژول مربوط به مالتی پلکسر های دیگر نیز نوشته شده اند.

```
1  module mux4to1_0 (
2      input wire B0,
3      input wire S1,
4      input wire S2,
5      output wire Y0
6  );
7      wire B0_not;
8      wire zero = 0;
9      wire one = 1;
10     not (B0_not, B0);
11
12     assign Y0 = (S1 == 0 && S2 == 0) ? B0 :
13         (S1 == 0 && S2 == 1) ? B0_not :
14         (S1 == 1 && S2 == 0) ? zero :
15         (S1 == 1 && S2 == 1) ? one : 0;
16
17 endmodule
```

حال میرسیم به سراغ فول ادر ها

برای مثال در فولدر ادر صفرم 3 ورودی A0 و Y0 و Cin0 تعریف میشوند و Y0 همان خروجی مالتی پلکسر صفرم ما است. این ماژول همچنین دارای دو خروجی است که خروجی D0 حاصل XOR 3 ورودی A0 و Y0 و Cin0 است و خروجی دوم یعنی Cout0 حاصل OR شدن (Y0 XOR Cin0) AND A0 و Cin0 AND Y0 می باشد.

به همین ترتیب برای 3 ماژول فول ادر بعدی نیز عمل میکنیم.

```
81  module full_adder_0 (
82      input wire A0,
83      input wire Y0,
84      input wire Cin0,
85      output wire D0,
86      output wire Cout0
87  );
88
89      assign D0 = A0 ^ Y0 ^ Cin0;
90      assign Cout0 = (A0 & (Y0 ^ Cin0)) | (Y0 & Cin0);
```

حال میرسیم به ماژول اصلی یعنی Top_module

از این ماژول برای ایجاد تمام سیم ها و اتصالات استفاده میکنیم.

در این نماژول تمامی وردی ها و خروجی هارا بار دیگر تعریف میکنیم تا بتوانیم

چون Y3 Y2 Y1 Y0 خروجی های مالتی پلکسر بودن اونارو به صورت وایر تعریف کردیم.

همچنین Cout0 و Cout1 و Cout2 را نیز به صورت وایر تعریف کردیم زیرا از cout فول ادر قبلی خارج شده و به cin فول ادر بعدی وارد میشوند.

از آنجایی که Cin0 به عنوان ورودی اولیه به فول ادر اول وارد میشود نیازی به تعریف آن از نوع وارد نیست

و همچنین Cout3 را نیز به عنوان وایر تعریف نکردیم زیرا به عنوان سیمی بین دو ماژول نمیباشد.

```
137  module top_module (
138      input wire B0, B1, B2, B3,
139      input wire S1, S2,
140      input wire A0, A1, A2, A3,
141      input wire Cin0,
142      output wire D0, D1, D2, D3,
143      output wire Cout3
144  );
145
146      wire Y0, Y1, Y2, Y3;
147      wire Cout0, Cout1, Cout2;
```

حال نوبت به خراخوانی ورودی ها و خروجی ها در ماژول اصلی میرسد تا کار به اتمام برسد.

برای اینکار در ماکس صفر B0 در B0 ، S1 در S1 ، S2 در S2 و Y0 در Y0 ریخته میشود و به همین شکلی ماکس های یک تا سه نیز فراخوانی میشوند.

```
149     mux4to1_0 mux_inst_0 (
150         .B0(B0),
151         .S1(S1),
152         .S2(S2),
153         .Y0(Y0)
154     );
```

و همینطور برای فول ادر صفر A0 در A0 ، Y0 در Y0 ، Cin0 در Cin0 ، D0 در D0 و Cout0 را در Cout0 میریزیم و به همین شکلی فول ادر های یک تا سه نیز فراخوانی میشوند.

```
179     full_adder_0 adder_inst_0 (
180         .A0(A0),
181         .Y0(Y0),
182         .Cin0(Cin0),
183         .D0(D0),
184         .Cout0(Cout0)
185     );
```