شماره دانشجویی: ۹۹۱۰۲۲۰۷

نام: ایمان محمدی

سوال ۱)

```
module q1;

    reg [4:0] a;
    reg [3:0] b;
    reg [0:5] c;
    reg [3:0] d;
    integer i = 16'h4a6c;

    initial
    begin
        a = 4'bx;
        b = 4'bz;
        c = 4'b1;
        d = i[8-:6];
        $display("%b %b %b %b, a, b, c, d);
    end

endmodule
```

خروجی کد به صورت

0xxxx zzzz 000001 1101

خواهد بود. حالا دلیل هر کدام را توضیح می‌دهیم.

در ابتدا در کد مشخص کردیم که a، ۵ بیت دارد؛ b، ۴ بیت دارد؛ c، ۶ بیت دارد و d، ۴ بیت دارد.

الف) یک عدد ۴ بیتی در a ریخته شده است و وقتی عدد یک بیتی x در چهار بیت اکستند شود، چون MSB آن x است، x-extend می‌شود و هر ۴ بیت آن x خواهد بود. در نتیجه در این رجیستر ۵ بیتی، بیت پنجم 0 خواهد ماند و در نهایت a به شکل 0xxxx، نمایش داده می‌شود. این اتفاق در تابع $display و با فراخوانی a توسط %b می‌افتد و سپس یک فاصله چاپ می‌شود.

ب) یک عدد ۴ بیتی در b ریخته شده است و وقتی عدد یک بیتی z در چهار بیت اکستند شود، چون MSB آن z است، z-extend می‌شود و هر ۴ بیت آن z خواهد بود. در نتیجه در این رجیستر ۴ بیتی، هر ۴ بیت، z خواهند شد و در نهایت b به شکل zzzz، نمایش داده می‌شود. این اتفاق در تابع $display و با فراخوانی b توسط %b می‌افتد و سپس یک فاصله چاپ می‌شود.

ج) یک عدد ۴ بیتی (همان ۱) در c ریخته شده است و وقتی عدد یک در چهار بیت اکستند شود، چون MSB آن ۱ است، zero-extend می‌شود و به شکل 0001 نمایش داده می‌شود. در نتیجه در این رجیستر ۶ بیتی، عدد به شکل 000001 خواهد شد و در نهایت c به شکل 000001، نمایش داده می‌شود. این اتفاق در تابع $display و با فراخوانی c توسط %b می‌افتد و سپس یک فاصله چاپ می‌شود.

د) در این‌جا می‌دانیم ۸ بیت اول i، 0011011 هستند. همچنین [-8:6] i به معنی بیت دوم تا ششم عدد i می‌باشد. اما این‌جا d ۴ بیت دارد و از بین بیت‌های مشخص شده‌ی i، 1101 در آن ریخته می‌شوند و در نهایت d به شکل 1101، نمایش داده می‌شود. این اتفاق در تابع display$ و با فراخوانی d توسط %b می‌افتد.

شماره دانشجویی: ۹۹۱۰۲۲۰۷

نام: ایمان محمدی

سوال ۲)

```
module q2(q, a, input [N-1:0] b, lda, ldb, clk);
    parameter N = 2;

    input [N-1:0] a
    input lda, ldb, clk;
    output [N-1:0] q;
    wire out_one;
    wire out_two;

    xor (out_one, a[N-1], ldb);
    and (b[0], lda, out_two);

endmodule
```

کد، مشکلاتی منطقی و همچنین مشکلاتی که برای کامپایل مشکل ایجاد می‌کنند، دارد.

به همه‌ی آن‌ها این زیر اشاره می‌شود.

الف) در خط اول، input [N-1 : 0] b مشکل کامپایل ایجاد می‌کند زیرا N پارامتری‌ست که در ماژول تعریف شده است و مقدارش آن‌جا مشخص می‌شود و در خط اول، این پارامتر تعریف نشده است و در نتیجه برای حل این مشکل کامپایل، input [N-1 : 0] b باید به داخل ماژول برده شود و در خط ۳ می‌تواند قرار گیرد.

ب) هر خط کد باید با سمی‌کالم (;) به اتمام برسد تا نشان‌دهنده‌ی تمام شدن دستور خوانده شده در آن خط باشد؛ در خط ۴ پس از input [N-1 : 0] a، یک ; جا افتاده با قرارگیری این semicolon در پایان این خط، این مشکل کامپایلی هم حل می‌شود.

ج) درست است که تعریف پورت‌های ماژول، یا همه در خود ماژول صورت گیرد و یا وقتی که خود ماژول تعریف می‌شود و این‌جا، در خط اول.

اما در این کد، مشابه ارور الف، b نباید در خط اول تعریف شود و درست است که در خود ماژول و مانند a و q تعریف شود. همچنین به خاطر قرارگیری عبارت input در خط اول، Ida، Idb و clk نیز به عنوان input خوانده می‌شوند و عملاً همه‌ی آن‌ها در خط اول و نیز در خود ماژول تعریف شده‌اند و این هم مشکل کامپایلری ایجاد می‌کند پس درست است که input [N-1 : 0] b با b جایگزین شود در خط اول و این عبارت در خود ماژول قرار گیرد.

(این قسمت رو به خاطر گفته‌ی تی‌ای درس در کوئرا می‌نویسم): این ۳ مشکل اگر حل شوند، مشکل کامپایلری حل می‌شود ولی بهتر است در خط آخر نیز ترتیب پورت‌های داده شده به and به شکل and (out_two, lda, b[0]) شود تا خروجی در out_two نوشته بود.

شماره دانشجویی: ۹۹۱۰۲۲۰۷

نام: ایمان محمدی

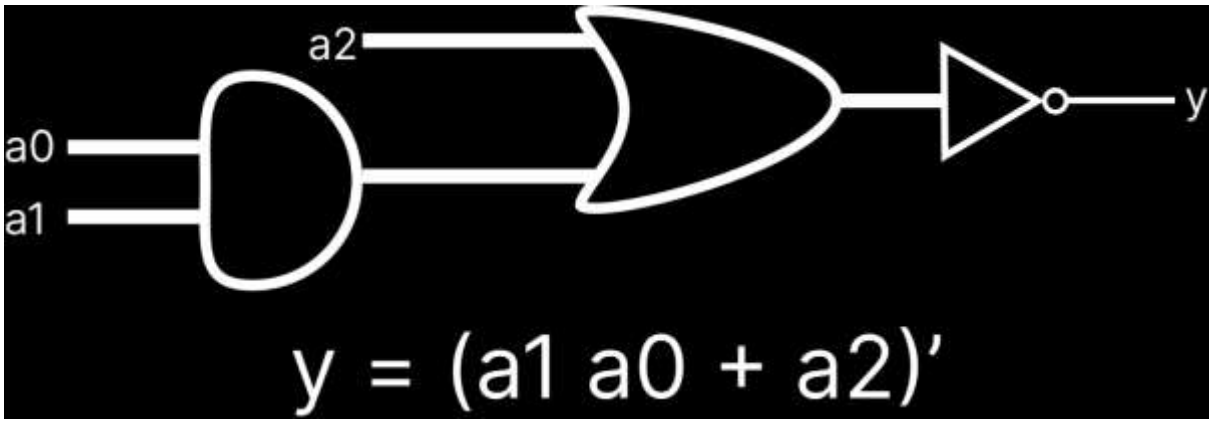
۳ – ماژولی را در نظر بگیرید که ورودی سه‌بیتی A را گرفته و در صورتی که کوچکتر از ۳ باشد خروجی y را ۱ می‌کند در غیر این‌صورت خروجی y همواره صفر است.

الف) مداری منطقی طراحی کنید که عملکرد فوق را داشته باشد.

ب) با توجه به مدار طراحی‌شده خودتان در قسمت قبل ماژولی (با سرخط روبرو) در سطح گیت (Gate Level) بنویسید که عملکرد فوق را داشته باشد.

```
module less_than_three(a, y);
```

سوال ۳) مدار طراحی شده:



ماژول نوشته شده:

```
module less_than_three(a, y);

input [2:0] a;

output y;

wire b;
wire c;

and (b, a[0], a[1]);
or (c, b, a[2]);
not (y, c);

endmodule
```

شماره دانشجویی: ۹۹۱۰۲۲۰۷

نام: ایمان محمدی

۴- جدول صحت روبرو را در نظر بگیرید.

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

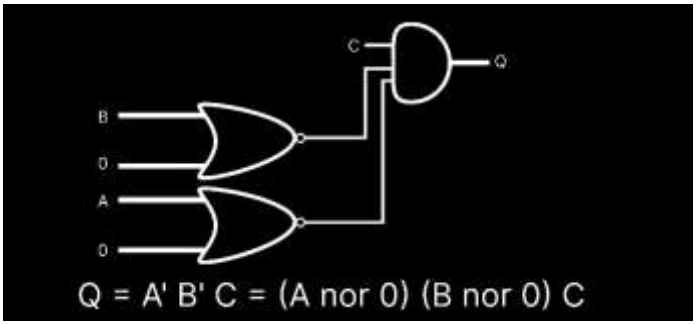
الف) مداری منطقی بدون استفاده از گیت NOT با سایر گیت‌های پایه طراحی کنید که عملکرد آن مطابق جدول صحت روبرو باشد.

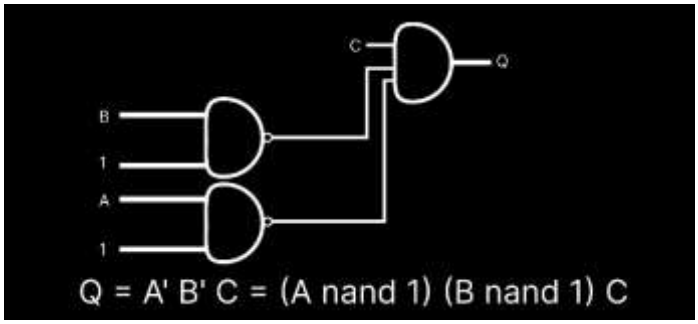
ب) با توجه به مدار طراحی‌شده خودتان در قسمت قبل ماژولی (با سرخط روبرو) در سطح گیت (Gate Level) بنویسید که عملکرد فوق را داشته باشد. (دقت کنید هم‌چنان مجاز به استفاده از گیت NOT نیستید)

```
module truth_table(A, B, C, Q);
```

سوال ۴) می‌دانیم برای not کردن، می‌توانیم مقدار مدنظرمان را با عدد ۱، nand کنیم و یا با ۰، nor کنیم.

Q = A' B' C = (A nor 0) (B nor 0) C = (A nand 1) (B nand 1) C





```
module truth_table(A, B, C, Q);

input A,B,C;

output Q;

wire A_not, B_not;

nand (A_not, A, 1);
nand (B_not, B, 1);

and (Q, A_not, B_not, C);

endmodule
```

```
module truth_table(A, B, C, Q);

input A,B,C;

output Q;

wire A_not, B_not;

nor (A_not, A, 0);
nor (B_not, B, 0);

and (Q, A_not, B_not, C);

endmodule
```