

```

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 این جا می دانیم ۸ بیت اول  $i$ ، 0011011 هستند. همچنین  $i$  به معنی بیت دوم تا ششم عدد  $i$  می باشد. اما این جا  $d$  ۴ بیت دارد و از بین بیت های مشخص شده ی  $i$ ، 1101 در آن ریخته می شوند و در نهایت  $d$  به شکل 1101، نمایش داده می شود. این اتفاق در تابع  $\$display$  و با فراخوانی  $d$  توسط  $\%b$  می افتد.