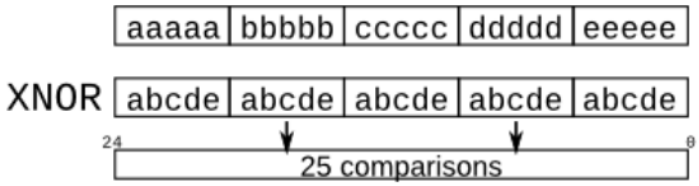
# Vector5

vector4 Previous Next module

Given five 1-bit signals (a, b, c, d, and e), compute all 25 pairwise one-bit comparisons in the 25-bit output vector. The output should be 1 if the two bits being compared are equal.

```
out[24] = ~a ^ a;    // a == a, so out[24] is always 1.
out[23] = ~a ^ b;
out[22] = ~a ^ c;
...
out[ 1] = ~e ^ d;
out[ 0] = ~e ^ e;
```




As the diagram shows, this can be done more easily using the replication and concatenation operators.

- The top vector is a concatenation of 5 repeats of each input
- The bottom vector is 5 repeats of a concatenation of the 5 inputs

설명

5개의 1-bit 신호가 주어지고 서로 겹치지 않게 5-bit 신호로 25-bit를 출력하라. 출력은 두 출력이 같다면 1이 나와야함(XNOR).

코드

HDLBits > 3_Vector5.v

```
1   module top_module (
2       input a, b, c, d, e,
3       output reg [24:0] out );
4
5       wire [24:0] temp1;
6       wire [24:0] temp2;
7
8       // 각 비트를 5번 반복하여 연결
9       assign temp1 = { {5{a}},{5{b}},{5{c}},{5{d}},{5{e}}};
10      assign temp2 =  {5{a,b,c,d,e}} // abcde를 5번 반복
11
12      always @(*) begin
13          integer i;
14          for(i=24; i>=0; i = i-1) begin
15              out[i] = ~(temp1[i] ^ temp2[i]);
16          end
17      end
18
19  endmodule
20
```

결과

```
to decrease run time.
Error (10170): Verilog HDL syntax error at top_module.v(12) near text: "always";  expecting ";". Check for and
fix any syntax errors that appear immediately before or at the specified keyword. The Intel FPGA Knowledge
Database contains many articles with specific details on how to resolve this error. Visit the Knowledge
Database at https://www.altera.com/support/support-resources/knowledge-base/search.html and search for this
specific error message number. File: /home/h/work/hdlbits.17709692/top_module.v Line: 12
Error (10112): Ignored design unit "top_module" at top_module.v(1) due to previous errors File:
/home/h/work/hdlbits.17709692/top_module.v Line: 1
Info (12021): Found 0 design units, including 0 entities, in source file top_module_wrap.v
Info (12021): Found 0 design units, including 0 entities, in source file tb_modules.sv
Error: Quartus Prime Analysis & Synthesis was unsuccessful. 2 errors, 1 warning
    Error: Peak virtual memory: 386 megabytes
    Error: Processing ended: Sun Aug 18 15:01:38 2024
    Error: Elapsed time: 00:00:00
    Error: Total CPU time (on all processors): 00:00:00
Error (23031): Evaluation of Tcl script /home/h/hdlbits/compile.tcl unsuccessful
Error: Quartus Prime Shell was unsuccessful. 8 errors, 1 warning
    Error: Peak virtual memory: 481 megabytes
    Error: Processing ended: Sun Aug 18 15:01:39 2024
    Error: Elapsed time: 00:00:02
    Error: Total CPU time (on all processors): 00:00:01
```

**Status: Compile Error**

The code did not compile. Check the error messages from Quartus.

12번째 줄 always에서 에러가 발생함.

코드를 다시 보니까 10번에서 세미콜론을 빼먹었음..

최종 코드

HDLBits > 3_Vector5.v

```
1   module top_module (
2       input a, b, c, d, e,
3       output reg [24:0] out );
4
5       wire [24:0] temp1;
6       wire [24:0] temp2;
7
8       assign temp1 = { {5{a}},{5{b}},{5{c}},{5{d}},{5{e}}};
9       assign temp2 =  {5{a, b, c, d, e}};
10
11      always @(*) begin
12          integer i;
13          for(i=24; i>=0; i = i-1) begin
14              out[i] = ~(temp1[i] ^ temp2[i]);
15          end
16      end
17
18  endmodule
19
20
```

결과

## vector5 — Compile and simulate

Running Quartus synthesis. Show Quartus messages...
Running ModelSim simulation. Show Modelsim messages...

## Status: Success!

You have solved 19 problems. See my progress...

### Warning messages that may be important

**Quartus messages (Show all)**

**Warning (13024): Output pins are stuck at VCC or GND**
This warning says that an output pin never changes (is "stuck"). This can sometimes indicate a bug if the output pin shouldn't be a constant. If this pin is not supposed to be constant, check for bugs that cause the value being assigned to never change (e.g., `assign a = x & ~x;`)

짚고 갈 점

1. replication 기법을 사용할 때, 중괄호로 하나하나 다 묶어서 처리하는 것

2. 세미콜론을 잊지 말자.

솔루션

## Solution

Show solution

```
module top_module (
    input a, b, c, d, e,
    output [24:0] out
);

    wire [24:0] top, bottom;
    assign top    = { {5{a}}, {5{b}}, {5{c}}, {5{d}}, {5{e}} };
    assign bottom = {5{a,b,c,d,e}};
    assign out = ~top ^ bottom; // Bitwise XNOR

    // This could be done on one line:
    // assign out = ~{ {5{a}}, {5{b}}, {5{c}}, {5{d}}, {5{e}} } ^ {5{a,b,c

endmodule
```

크기가 같으니까 알아서 비교 처리되는 모습을 볼 수 있다.