Assignment 2

Design the following circuit using Verilog and create a testbench for it to check it's functionality

1- ALSU is a logic unit that can perform logical ,arithmetic and shift operations on input ports.

Input	Width	Description
A	***WIDTH***	*Input port A.*
B	***WIDTH***	*Input port B.*
Cin	***1***	*CARRY IN used if parameter FULL ADDER is "ON".*
red_op_A ✓	***1***	*When set high this indicates reduction operation would be executed on A rather than bitwise op on A and B* ***ONLY WHEN opcode indicates AND & XOR operations****.*
red_op_B ✓	***1***	*When set high this indicates reduction operation would be executed on B rather than bitwise op on A and B* ***ONLY WHEN opcode indicates AND & XOR operations****.*
opcode	***3***	*Opcode has a separate table to describe different operations to be executed.*
bypass_A ✓	***1***	*When set to high this indicates that port A will be passed to the output* ***ignoring the opcode operation.***
bypass_B ✓	***1***	*When set to high this indicates that port B will be passed to the output* ***ignoring the opcode operation.***

Outputs and parameters

Output	Width	Description
Out	***WIDTH+1***	*Out of the ALSU,****OUTPUT INCLUDES THE CARRY OUT.***
Odd parity	***1***	*Odd parity of the output "out" Arithmetic operations* ***only otherwise should be Zero.***
Invalid	***1***	*Activated when invalid cases and invalid opcodes are executed.*

parameter	DEFAULT VALUE	Description
INPUT_PRIORITY	***A***	*If there is a conflict occurs , conflicts occur in* ***two scenarios*** *, red_op_A and red_op_B are both set to high or bypass_A and bypass_B are both set to high Legal values for this parameter are A and B.*
FULL_ADDER	***"ON"***	*When this parameter has value "ON" then cin input must be taken into consideration in the addition operation between A and B This parameter takes value of "ON" and "OFF".*
WIDTH	***4***	*Length of input A & B*

Opcode	***Operation***
000	*AND*
001	*XOR*
010	*Addition*
011	*Multiplication*
100	*Subtract **if A>B** so expression will be (A-B) and **if A<B** so expression will be (B-A)*
101	*Division **(if division by zero don't do division just by pass the non zero value) and if both A and B are equal to zero pass any one of them) (A/B)***
110	*Invalid opcode*
111	*Invalid opcode*

Invalid cases

1. *Opcode bits are set to 110 or 111*
2. *red_op_a or red_op_B are set to high and the opcode is not AND or XOR operation*
3. *Division by zero .*

Test bench

In Testbench ths system function "$urandom_range" which returns randomized unsigned integer within range if you want to strict the opcode to have valid opcodes.

Requirements

1. *Randomize test bench where inputs are randomly generated.*
2. *ALL invalid cases should be tested.*
3. *Self-checking Test bench is optional (bouns).*
4. *If you need to make some directed tests as you can't get a specific case it's okay.*

2. *A designer has just made a design for ALU and it is encrypted due to privacy and there is some bugs in the design functionality and you as a verification engineer should find these errors by creating a test bench and test all cases and if you find errors you should write where they are and explain it according to the specs written .*

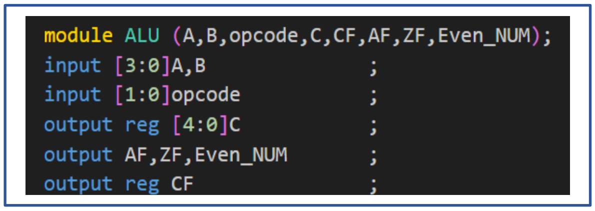
Input A (4-bits) , Input B(4-bits) , output C (5-bits) , output CF (1-bit) , output ZF(1-bit) , output Even_Num (1-bit) , AF (1-bit).

Input	Width	Description
A	4	Input port A.
B	4	Input port B.
opcode	2	Opcode has a separate table to describe different operations to be executed.

Output	Width	Description
C	5	Output port.
CF	1	Carry flag only in case of **addition only otherwise it equals to zero**.
Even_NUM	1	If output C is even number or not.
AF	1	Calculated only in case of **addition only otherwise it equals to zero.**
ZF	1	Calculated in all cases.

Opcode	Operation
00	AND
01	Addition
10	XOR
11	OR

Code Snippet



```
module ALU (A,B,opcode,C,CF,AF,ZF,Even_NUM);
input [3:0]A,B              ;
input [1:0]opcode           ;
output reg [4:0]C           ;
output AF,ZF,Even_NUM       ;
output reg CF               ;
```

Requirements

1. *Create a directed testbench to find the bugs and define each of them and explain it.*

Deliverables

1) The assignment should be submitted as a PDF file with this format Name _Assignment2 for example Abdelrahman_Essam_Assignment2 .

2) Snippets from the waveforms captured from QuestaSim for each design with inputs assigned values and output values visible.