**Ayberk Savaş 32433 Lab 3 Assignment**

15-bit Adder-Subtractor with Overflow Detector Circuit using Verilog

**Building Blocks**

CLA_3bit uses XOR (to compute Propagate and calculate RES in the end), AND (to compute Generate) and OR (to form carry logic) gates to perform its job.

CLA_15bit_top uses CLA_3bit as a submodule (5 of them), there is a part that checks mode input to understand if the calculation is an addition or subtraction and it negates input B if mode=1 (makes it 1's complement) after that it sets Cin = mode to make it 2's complement. There is a part that assigns final carry-out and another part that detects overflows.

**Modules and Submodules**

I used 3-Bit Carry Lookahead Adders (CLA) submodules to form 15-bits Adder-Subtractor module. 3-bit CLA takes 3 inputs: 2 numbers A and B in 3-bit form and a Cin (Carry-in input) for taking the carry form the previous step. 3-bit CLA outputs 2 numbers: one is RES (result) which is a 3-bit number showing the sum of A and B, and there is Carry output which will be transmitted to the next block as Cin. It uses Propagate and Generate bitwise operations to calculate RES and Carry. By adding 5 of these 3-bit CLAs we form a 15-bits Adder-Subtractor.

**Module CLA_15bit_top**

It is a module that has 3 inputs: two 15-bit numbers A and B, and another input called mode that takes 0(addition) or 1(subtraction) to perform relevant operations. It has 2 outputs: Cout which is final carry-out and OVF which is the Overflow flag. 15-bit CLA is built up by 5 of 3-bit CLAs to calculate 15-bit cases. For addition cases it operates in the same way as 3-bit CLA but for subtraction it takes the 1's complement of B when mode is 1 and by setting Cin = mode we are assigning Cin 1 for mode 1 and by this we are making it a 2's complement.

**Testbench**

Testbench tests CLA-15-bit module by inputting different edge cases mentioned in the assignment paper. It gives some values to A, B and mode to test relevant cases and outputs of the testbench are S, Cout and OVF values.

**Test cases given in the template:**

```
    A = 15'd0;  // Set all inputs to zero.
    B = 15'd0;
    mode = 1'd0;
    #10;     // Wait 10 time units.
    A = 15'd25;
    B = 15'd50;
    mode = 1'd0; // For addition
    #10;     // Wait 10 time units.
    A = 15'd30;
    B = 15'd100;
    mode = 1'd1; // For subtraction
    #10;     // Wait 10 time units.
```

**Test cases wanted in the assignment paper:**

a. addition without a carry and without an overflow,

```
    A = 15'd25;
    B = 15'd50;
    mode = 1'd0; // For addition
        #10;     // Wait 10 time units.
```

b. addition without a carry and with an overflow,

```
  A = 15'd16383;
    B = 15'd2;
    mode = 1'd0;
    #10;
```

c. addition with a carry and without an overflow,

```
 A = 15'd63;
    B = 15'd63;
    mode = 1'd0;
    #10;
```

d. addition with a carry and with an overflow.

```
    A = 15'd16383;
    B = 15'd1;
    mode = 1'd0;
    #10;
```

e. subtraction without a carry and without an overflow,

```
    A = 15'd30;
    B = 15'd100;
    mode = 1'd1;
    #10;
```

f. subtraction without a carry and with an overflow,

```
    A = 15'd16383;
    B = -15'd16384;
    mode = 1'd1;
    #10;
```

g. subtraction with a carry and without an overflow,

```
    A = 15'd63;
    B = 15'd31;
    mode = 1'd1;
    #10;
```

h. subtraction with a carry and with an overflow.

```
    A = -15'd16384;
    B = 15'd1;
    mode = 1'd1;
    #10;
```

Here is the waveform of the output: