CS6230 Assignment 1: Design and verification of MAC unit

Santhosh S P ee21b119, Pranav Vishwanath ee21b108

28 October 2024

1 Introduction

The following MAC unit performs the operation A*B + C where A and B are 8-bit integers and C is a 32-bit integer.

2 Design

2.1 32-bit adder

The full adder unit is defined by the following function:

```
function Bit#(2) fa_op_1bit(Bit#(1) a, Bit#(1) b, Bit#(1) c);
 Bit#(1) temp = a^b;
 Bit#(1) sum = temp^c;
 Bit#(1) carry_out = (a & b)|(c & temp);
 return {carry_out, sum};
endfunction
```

The 32-bit adder is constructed in constructed in ripple carry fashion.

```
function Bit#(33) add_op_32(Bit#(32) a, Bit#(32) b, Bit#(1) c_in);
 Bit#(32) sum = 0;
 Bit#(33) carry = 0;
 carry[0] = c_in;
 for (Integer i = 0; i<32; i=i+1)
  begin
   Bit#(2) fa_res = fa_op_1bit(a[i], b[i], carry[i]);
   sum[i] = fa_res[0];
   carry[i+1] = fa_res[1];
  end
 return {carry[32], sum};
endfunction
```

2.2 8-bit multiplier

The 8-bit multiplication is implemented by repeated shifting and addition of the multiplicand to a partial sum. The following function mul8 implements this. It uses the 8-bit ripple carry adder implemented by the function

```
add_op_8
```

```
function Bit#(16) mul8(Bit#(8) multiplicand, Bit#(8) multiplier);
 Bit#(8) temp_lower = 0;
 Bit#(8) temp_upper = 0;
 for (Integer i = 0; i<8; i=i+1)
  begin
   Bit#(8) to_add = (multiplier[i] == 0) ? 0 : multiplicand;
   Bit#(9) part_sum = add_op_8(temp_upper, to_add, 0);
   temp_lower[i] = part_sum[0];
   temp_upper = part_sum[8:1];
  end
 return {temp_upper, temp_lower};
endfunction
```

2.3 The MAC operation

In the function mac, we extend the result of the multiplication A*B to 32-bits and send it to the 32-bit ripple carry adder with C as the other input.

```
function Bit#(33) mac(Bit#(8) a, Bit#(8) b, Bit#(32) c);
 Bit#(32) inter = signExtend(mul8(a,b));
 Bit#(33) res = add_op_32(inter,c,0);
 return res;
endfunction
```

3 Testing it with user-given inputs

Change the values of A, B and C inside the getResult() function present in the testbench module

```
mkTb_mac_wrap
```

Enter the following commands in the terminal one by one:

```
bsc -verilog mac.bsv
bsc -o sim -e mkTb_mac_wrap mkTb_mac_wrap.v
./sim
```