

CA_Lab3

K. Ram Mohan

COE19B055

Verilog Code:

```
`include "kgp_generate.v"

// Module for Full adder
module FA (in0, in1, cin, sum);
    input in0, in1, cin;
    output sum;

    xor(sum, in0, in1, cin);
endmodule

module MUX_KGP_MODIFICATION_0_BIT (kgp_curr, kgp_pre, out);
    input[1:0] kgp_curr, kgp_pre;
    output out;

    wire[16:0] c;
    wire tmp;

    wire[1:0] neg_kgp_curr;
    wire[1:0] neg_kgp_pre;

    not(neg_kgp_curr[0], kgp_curr[0]);
    not(neg_kgp_curr[1], kgp_curr[1]);

    not(neg_kgp_pre[0], kgp_pre[0]);
```

```

not(neg_kgp_pre[1], kgp_pre[1]);

and(c[0], 0, neg_kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[0], neg_kgp_pre[0]);
and(c[1], 0, neg_kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[0], kgp_pre[0]);
and(c[2], 0, neg_kgp_curr[1], neg_kgp_curr[0], kgp_pre[0], neg_kgp_pre[0]);

and(c[3], 0, kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[1], neg_kgp_pre[0]);
and(c[4], 0, kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[1], kgp_pre[0]);
and(c[5], 0, kgp_curr[1], neg_kgp_curr[0], kgp_pre[1], neg_kgp_pre[0]);

and(c[6], 0, neg_kgp_curr[1], kgp_curr[0], neg_kgp_pre[1], neg_kgp_pre[0]);
and(c[7], 1, neg_kgp_curr[1], kgp_curr[0], neg_kgp_pre[1], kgp_pre[0]);
and(c[8], 0, neg_kgp_curr[1], kgp_curr[0], kgp_pre[1], neg_kgp_pre[0]);

or(tmp, c[0], c[1], c[2], c[3]);
or(tmp2, c[4], c[5], c[6], c[7]);
or(out, tmp, tmp2, c[8]);
endmodule

module MUX_KGP_MODIFICATION_1_BIT(kgp_curr, kgp_pre, out);
    input[1:0] kgp_curr, kgp_pre;
    output out;

    wire[16:0] c;
    wire tmp;

    wire[1:0] neg_kgp_curr;
    wire[1:0] neg_kgp_pre;

    not(neg_kgp_curr[0], kgp_curr[0]);
    not(neg_kgp_curr[1], kgp_curr[1]);

```

```
not(neg_kgp_pre[0], kgp_pre[0]);
```

```
not(neg_kgp_pre[1], kgp_pre[1]);
```

```
and(c[0], 0, neg_kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[0], neg_kgp_pre[0]);
```

```
and(c[1], 0, neg_kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[0], kgp_pre[0]);
```

```
and(c[2], 0, neg_kgp_curr[1], neg_kgp_curr[0], kgp_pre[0], neg_kgp_pre[0]);
```

```
and(c[3], 1, kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[1], neg_kgp_pre[0]);
```

```
and(c[4], 1, kgp_curr[1], neg_kgp_curr[0], neg_kgp_pre[1], kgp_pre[0]);
```

```
and(c[5], 1, kgp_curr[1], neg_kgp_curr[0], kgp_pre[1], neg_kgp_pre[0]);
```

```
and(c[6], 0, neg_kgp_curr[1], kgp_curr[0], neg_kgp_pre[1], neg_kgp_pre[0]);
```

```
and(c[7], 0, neg_kgp_curr[1], kgp_curr[0], neg_kgp_pre[1], kgp_pre[0]);
```

```
and(c[8], 1, neg_kgp_curr[1], kgp_curr[0], kgp_pre[1], neg_kgp_pre[0]);
```

```
or(tmp, c[0], c[1], c[2], c[3]);
```

```
or(tmp2, c[4], c[5], c[6], c[7]);
```

```
or(out, tmp, tmp2, c[8]);
```

```
endmodule
```

```
module KGP_MODIFICATION (kgp_curr, kgp_pre, kgp_new);
```

```
input[1:0] kgp_curr, kgp_pre;
```

```
output[1:0] kgp_new;
```

```
MUX_KGP_MODIFICATION_0_BIT kgp_mod1630(kgp_curr, kgp_pre, kgp_new[0]);
```

```
MUX_KGP_MODIFICATION_1_BIT kgp_mod1631(kgp_curr, kgp_pre, kgp_new[1]);
```

```
endmodule
```

```
// Module fpr recursive doubling based adder
```

```
module RCDA (a, b, c_input, sum, carry);
```

```

input[63:0] a, b;
input c_input;
output[63:0] sum;
output carry;

// 00-> kill, 01->propagate, 10->generate
wire[1:0] kgp[0:63];
wire[1:0] kgp1[0:63];
wire[1:0] kgp2[0:63];
wire[1:0] kgp3[0:63];
wire[1:0] kgp4[0:63];
wire[1:0] kgp5[0:63];
wire[1:0] kgp6[0:63];
wire[1:0] kgp7[0:63];
wire[1:0] kgp8[0:63];
wire[1:0] kgp_t1[0:63];
wire[1:0] kgp_t2[0:63];

wire[1:0] bit2[0:63];
wire[63:0] Cin;

reg[6:0] j, k, m, n;

genvar i;
for (i = 0; i<64; i=i+1) begin
    assign bit2[i] = {b[i], a[i]};
end

MUX_8X1_0_BIT mux00(bit2[0], c_input, kgp[0][0]);
MUX_8X1_1_BIT mux01(bit2[0], c_input, kgp[0][1]);

```

MUX_4X1_KGP muxkgp1(bit2[1], kgp[1]);
MUX_4X1_KGP muxkgp2(bit2[2], kgp[2]);
MUX_4X1_KGP muxkgp3(bit2[3], kgp[3]);
MUX_4X1_KGP muxkgp4(bit2[4], kgp[4]);
MUX_4X1_KGP muxkgp5(bit2[5], kgp[5]);
MUX_4X1_KGP muxkgp6(bit2[6], kgp[6]);
MUX_4X1_KGP muxkgp7(bit2[7], kgp[7]);
MUX_4X1_KGP muxkgp8(bit2[8], kgp[8]);
MUX_4X1_KGP muxkgp9(bit2[9], kgp[9]);
MUX_4X1_KGP muxkgp10(bit2[10], kgp[10]);
MUX_4X1_KGP muxkgp11(bit2[11], kgp[11]);
MUX_4X1_KGP muxkgp12(bit2[12], kgp[12]);
MUX_4X1_KGP muxkgp13(bit2[13], kgp[13]);
MUX_4X1_KGP muxkgp14(bit2[14], kgp[14]);
MUX_4X1_KGP muxkgp15(bit2[15], kgp[15]);
MUX_4X1_KGP muxkgp16(bit2[16], kgp[16]);
MUX_4X1_KGP muxkgp17(bit2[17], kgp[17]);
MUX_4X1_KGP muxkgp18(bit2[18], kgp[18]);
MUX_4X1_KGP muxkgp19(bit2[19], kgp[19]);
MUX_4X1_KGP muxkgp20(bit2[20], kgp[20]);
MUX_4X1_KGP muxkgp21(bit2[21], kgp[21]);
MUX_4X1_KGP muxkgp22(bit2[22], kgp[22]);
MUX_4X1_KGP muxkgp23(bit2[23], kgp[23]);
MUX_4X1_KGP muxkgp24(bit2[24], kgp[24]);
MUX_4X1_KGP muxkgp25(bit2[25], kgp[25]);
MUX_4X1_KGP muxkgp26(bit2[26], kgp[26]);
MUX_4X1_KGP muxkgp27(bit2[27], kgp[27]);
MUX_4X1_KGP muxkgp28(bit2[28], kgp[28]);
MUX_4X1_KGP muxkgp29(bit2[29], kgp[29]);
MUX_4X1_KGP muxkgp30(bit2[30], kgp[30]);
MUX_4X1_KGP muxkgp31(bit2[31], kgp[31]);

MUX_4X1_KGP muxkgp32(bit2[32], kgp[32]);
MUX_4X1_KGP muxkgp33(bit2[33], kgp[33]);
MUX_4X1_KGP muxkgp34(bit2[34], kgp[34]);
MUX_4X1_KGP muxkgp35(bit2[35], kgp[35]);
MUX_4X1_KGP muxkgp36(bit2[36], kgp[36]);
MUX_4X1_KGP muxkgp37(bit2[37], kgp[37]);
MUX_4X1_KGP muxkgp38(bit2[38], kgp[38]);
MUX_4X1_KGP muxkgp39(bit2[39], kgp[39]);
MUX_4X1_KGP muxkgp40(bit2[40], kgp[40]);
MUX_4X1_KGP muxkgp41(bit2[41], kgp[41]);
MUX_4X1_KGP muxkgp42(bit2[42], kgp[42]);
MUX_4X1_KGP muxkgp43(bit2[43], kgp[43]);
MUX_4X1_KGP muxkgp44(bit2[44], kgp[44]);
MUX_4X1_KGP muxkgp45(bit2[45], kgp[45]);
MUX_4X1_KGP muxkgp46(bit2[46], kgp[46]);
MUX_4X1_KGP muxkgp47(bit2[47], kgp[47]);
MUX_4X1_KGP muxkgp48(bit2[48], kgp[48]);
MUX_4X1_KGP muxkgp49(bit2[49], kgp[49]);
MUX_4X1_KGP muxkgp50(bit2[50], kgp[50]);
MUX_4X1_KGP muxkgp51(bit2[51], kgp[51]);
MUX_4X1_KGP muxkgp52(bit2[52], kgp[52]);
MUX_4X1_KGP muxkgp53(bit2[53], kgp[53]);
MUX_4X1_KGP muxkgp54(bit2[54], kgp[54]);
MUX_4X1_KGP muxkgp55(bit2[55], kgp[55]);
MUX_4X1_KGP muxkgp56(bit2[56], kgp[56]);
MUX_4X1_KGP muxkgp57(bit2[57], kgp[57]);
MUX_4X1_KGP muxkgp58(bit2[58], kgp[58]);
MUX_4X1_KGP muxkgp59(bit2[59], kgp[59]);
MUX_4X1_KGP muxkgp60(bit2[60], kgp[60]);
MUX_4X1_KGP muxkgp61(bit2[61], kgp[61]);
MUX_4X1_KGP muxkgp62(bit2[62], kgp[62]);

```
MUX_4X1_KGP muxkgp63(bit2[63], kgp[63]);
```

```
// KGP modification
```

```
// Stage 1
```

```
KGP_MODIFICATION kgp_modification163(kgp[63], kgp[62], kgp1[63]);  
KGP_MODIFICATION kgp_modification162(kgp[62], kgp[61], kgp1[62]);  
KGP_MODIFICATION kgp_modification161(kgp[61], kgp[60], kgp1[61]);  
KGP_MODIFICATION kgp_modification160(kgp[60], kgp[59], kgp1[60]);  
KGP_MODIFICATION kgp_modification159(kgp[59], kgp[58], kgp1[59]);  
KGP_MODIFICATION kgp_modification158(kgp[58], kgp[57], kgp1[58]);  
KGP_MODIFICATION kgp_modification157(kgp[57], kgp[56], kgp1[57]);  
KGP_MODIFICATION kgp_modification156(kgp[56], kgp[55], kgp1[56]);  
KGP_MODIFICATION kgp_modification155(kgp[55], kgp[54], kgp1[55]);  
KGP_MODIFICATION kgp_modification154(kgp[54], kgp[53], kgp1[54]);  
KGP_MODIFICATION kgp_modification153(kgp[53], kgp[52], kgp1[53]);  
KGP_MODIFICATION kgp_modification152(kgp[52], kgp[51], kgp1[52]);  
KGP_MODIFICATION kgp_modification151(kgp[51], kgp[50], kgp1[51]);  
KGP_MODIFICATION kgp_modification150(kgp[50], kgp[49], kgp1[50]);  
KGP_MODIFICATION kgp_modification149(kgp[49], kgp[48], kgp1[49]);  
KGP_MODIFICATION kgp_modification148(kgp[48], kgp[47], kgp1[48]);  
KGP_MODIFICATION kgp_modification147(kgp[47], kgp[46], kgp1[47]);  
KGP_MODIFICATION kgp_modification146(kgp[46], kgp[45], kgp1[46]);  
KGP_MODIFICATION kgp_modification145(kgp[45], kgp[44], kgp1[45]);  
KGP_MODIFICATION kgp_modification144(kgp[44], kgp[43], kgp1[44]);  
KGP_MODIFICATION kgp_modification143(kgp[43], kgp[42], kgp1[43]);  
KGP_MODIFICATION kgp_modification142(kgp[42], kgp[41], kgp1[42]);  
KGP_MODIFICATION kgp_modification141(kgp[41], kgp[40], kgp1[41]);  
KGP_MODIFICATION kgp_modification140(kgp[40], kgp[39], kgp1[40]);  
KGP_MODIFICATION kgp_modification139(kgp[39], kgp[38], kgp1[39]);  
KGP_MODIFICATION kgp_modification138(kgp[38], kgp[37], kgp1[38]);
```

KGP_MODIFICATION kgp_modification137(kgp[37], kgp[36], kgp1[37]);
KGP_MODIFICATION kgp_modification136(kgp[36], kgp[35], kgp1[36]);
KGP_MODIFICATION kgp_modification135(kgp[35], kgp[34], kgp1[35]);
KGP_MODIFICATION kgp_modification134(kgp[34], kgp[33], kgp1[34]);
KGP_MODIFICATION kgp_modification133(kgp[33], kgp[32], kgp1[33]);
KGP_MODIFICATION kgp_modification132(kgp[32], kgp[31], kgp1[32]);
KGP_MODIFICATION kgp_modification131(kgp[31], kgp[30], kgp1[31]);
KGP_MODIFICATION kgp_modification130(kgp[30], kgp[29], kgp1[30]);
KGP_MODIFICATION kgp_modification129(kgp[29], kgp[28], kgp1[29]);
KGP_MODIFICATION kgp_modification128(kgp[28], kgp[27], kgp1[28]);
KGP_MODIFICATION kgp_modification127(kgp[27], kgp[26], kgp1[27]);
KGP_MODIFICATION kgp_modification126(kgp[26], kgp[25], kgp1[26]);
KGP_MODIFICATION kgp_modification125(kgp[25], kgp[24], kgp1[25]);
KGP_MODIFICATION kgp_modification124(kgp[24], kgp[23], kgp1[24]);
KGP_MODIFICATION kgp_modification123(kgp[23], kgp[22], kgp1[23]);
KGP_MODIFICATION kgp_modification122(kgp[22], kgp[21], kgp1[22]);
KGP_MODIFICATION kgp_modification121(kgp[21], kgp[20], kgp1[21]);
KGP_MODIFICATION kgp_modification120(kgp[20], kgp[19], kgp1[20]);
KGP_MODIFICATION kgp_modification119(kgp[19], kgp[18], kgp1[19]);
KGP_MODIFICATION kgp_modification118(kgp[18], kgp[17], kgp1[18]);
KGP_MODIFICATION kgp_modification117(kgp[17], kgp[16], kgp1[17]);
KGP_MODIFICATION kgp_modification116(kgp[16], kgp[15], kgp1[16]);
KGP_MODIFICATION kgp_modification115(kgp[15], kgp[14], kgp1[15]);
KGP_MODIFICATION kgp_modification114(kgp[14], kgp[13], kgp1[14]);
KGP_MODIFICATION kgp_modification113(kgp[13], kgp[12], kgp1[13]);
KGP_MODIFICATION kgp_modification112(kgp[12], kgp[11], kgp1[12]);
KGP_MODIFICATION kgp_modification111(kgp[11], kgp[10], kgp1[11]);
KGP_MODIFICATION kgp_modification110(kgp[10], kgp[9], kgp1[10]);
KGP_MODIFICATION kgp_modification19(kgp[9], kgp[8], kgp1[9]);
KGP_MODIFICATION kgp_modification18(kgp[8], kgp[7], kgp1[8]);
KGP_MODIFICATION kgp_modification17(kgp[7], kgp[6], kgp1[7]);


```
KGP_MODIFICATION kgp_modification16(kgp[6], kgp[5], kgp1[6]);
KGP_MODIFICATION kgp_modification15(kgp[5], kgp[4], kgp1[5]);
KGP_MODIFICATION kgp_modification14(kgp[4], kgp[3], kgp1[4]);
KGP_MODIFICATION kgp_modification13(kgp[3], kgp[2], kgp1[3]);
KGP_MODIFICATION kgp_modification12(kgp[2], kgp[1], kgp1[2]);
KGP_MODIFICATION kgp_modification11(kgp[1], kgp[0], kgp1[1]);
assign kgp1[0] = kgp[0];
```

```
// Stage2
```

```
KGP_MODIFICATION kgp_modification263(kgp1[63], kgp1[61], kgp2[63]);
KGP_MODIFICATION kgp_modification262(kgp1[62], kgp1[60], kgp2[62]);
KGP_MODIFICATION kgp_modification261(kgp1[61], kgp1[59], kgp2[61]);
KGP_MODIFICATION kgp_modification260(kgp1[60], kgp1[58], kgp2[60]);
KGP_MODIFICATION kgp_modification259(kgp1[59], kgp1[57], kgp2[59]);
KGP_MODIFICATION kgp_modification258(kgp1[58], kgp1[56], kgp2[58]);
KGP_MODIFICATION kgp_modification257(kgp1[57], kgp1[55], kgp2[57]);
KGP_MODIFICATION kgp_modification256(kgp1[56], kgp1[54], kgp2[56]);
KGP_MODIFICATION kgp_modification255(kgp1[55], kgp1[53], kgp2[55]);
KGP_MODIFICATION kgp_modification254(kgp1[54], kgp1[52], kgp2[54]);
KGP_MODIFICATION kgp_modification253(kgp1[53], kgp1[51], kgp2[53]);
KGP_MODIFICATION kgp_modification252(kgp1[52], kgp1[50], kgp2[52]);
KGP_MODIFICATION kgp_modification251(kgp1[51], kgp1[49], kgp2[51]);
KGP_MODIFICATION kgp_modification250(kgp1[50], kgp1[48], kgp2[50]);
KGP_MODIFICATION kgp_modification249(kgp1[49], kgp1[47], kgp2[49]);
KGP_MODIFICATION kgp_modification248(kgp1[48], kgp1[46], kgp2[48]);
KGP_MODIFICATION kgp_modification247(kgp1[47], kgp1[45], kgp2[47]);
KGP_MODIFICATION kgp_modification246(kgp1[46], kgp1[44], kgp2[46]);
KGP_MODIFICATION kgp_modification245(kgp1[45], kgp1[43], kgp2[45]);
KGP_MODIFICATION kgp_modification244(kgp1[44], kgp1[42], kgp2[44]);
KGP_MODIFICATION kgp_modification243(kgp1[43], kgp1[41], kgp2[43]);
KGP_MODIFICATION kgp_modification242(kgp1[42], kgp1[40], kgp2[42]);
```

KGP_MODIFICATION kgp_modification241(kgp1[41], kgp1[39], kgp2[41]);
KGP_MODIFICATION kgp_modification240(kgp1[40], kgp1[38], kgp2[40]);
KGP_MODIFICATION kgp_modification239(kgp1[39], kgp1[37], kgp2[39]);
KGP_MODIFICATION kgp_modification238(kgp1[38], kgp1[36], kgp2[38]);
KGP_MODIFICATION kgp_modification237(kgp1[37], kgp1[35], kgp2[37]);
KGP_MODIFICATION kgp_modification236(kgp1[36], kgp1[34], kgp2[36]);
KGP_MODIFICATION kgp_modification235(kgp1[35], kgp1[33], kgp2[35]);
KGP_MODIFICATION kgp_modification234(kgp1[34], kgp1[32], kgp2[34]);
KGP_MODIFICATION kgp_modification233(kgp1[33], kgp1[31], kgp2[33]);
KGP_MODIFICATION kgp_modification232(kgp1[32], kgp1[30], kgp2[32]);
KGP_MODIFICATION kgp_modification231(kgp1[31], kgp1[29], kgp2[31]);
KGP_MODIFICATION kgp_modification230(kgp1[30], kgp1[28], kgp2[30]);
KGP_MODIFICATION kgp_modification229(kgp1[29], kgp1[27], kgp2[29]);
KGP_MODIFICATION kgp_modification228(kgp1[28], kgp1[26], kgp2[28]);
KGP_MODIFICATION kgp_modification227(kgp1[27], kgp1[25], kgp2[27]);
KGP_MODIFICATION kgp_modification226(kgp1[26], kgp1[24], kgp2[26]);
KGP_MODIFICATION kgp_modification225(kgp1[25], kgp1[23], kgp2[25]);
KGP_MODIFICATION kgp_modification224(kgp1[24], kgp1[22], kgp2[24]);
KGP_MODIFICATION kgp_modification223(kgp1[23], kgp1[21], kgp2[23]);
KGP_MODIFICATION kgp_modification222(kgp1[22], kgp1[20], kgp2[22]);
KGP_MODIFICATION kgp_modification221(kgp1[21], kgp1[19], kgp2[21]);
KGP_MODIFICATION kgp_modification220(kgp1[20], kgp1[18], kgp2[20]);
KGP_MODIFICATION kgp_modification219(kgp1[19], kgp1[17], kgp2[19]);
KGP_MODIFICATION kgp_modification218(kgp1[18], kgp1[16], kgp2[18]);
KGP_MODIFICATION kgp_modification217(kgp1[17], kgp1[15], kgp2[17]);
KGP_MODIFICATION kgp_modification216(kgp1[16], kgp1[14], kgp2[16]);
KGP_MODIFICATION kgp_modification215(kgp1[15], kgp1[13], kgp2[15]);
KGP_MODIFICATION kgp_modification214(kgp1[14], kgp1[12], kgp2[14]);
KGP_MODIFICATION kgp_modification213(kgp1[13], kgp1[11], kgp2[13]);
KGP_MODIFICATION kgp_modification212(kgp1[12], kgp1[10], kgp2[12]);
KGP_MODIFICATION kgp_modification211(kgp1[11], kgp1[9], kgp2[11]);

```
KGP_MODIFICATION kgp_modification210(kgp1[10], kgp1[8], kgp2[10]);
KGP_MODIFICATION kgp_modification29(kgp1[9], kgp1[7], kgp2[9]);
KGP_MODIFICATION kgp_modification28(kgp1[8], kgp1[6], kgp2[8]);
KGP_MODIFICATION kgp_modification27(kgp1[7], kgp1[5], kgp2[7]);
KGP_MODIFICATION kgp_modification26(kgp1[6], kgp1[4], kgp2[6]);
KGP_MODIFICATION kgp_modification25(kgp1[5], kgp1[3], kgp2[5]);
KGP_MODIFICATION kgp_modification24(kgp1[4], kgp1[2], kgp2[4]);
KGP_MODIFICATION kgp_modification23(kgp1[3], kgp1[1], kgp2[3]);
KGP_MODIFICATION kgp_modification22(kgp1[2], kgp1[0], kgp2[2]);
assign kgp2[1] = kgp1[1];
assign kgp2[0] = kgp1[0];
```

```
// Stage 3
```

```
KGP_MODIFICATION kgp_modification363(kgp2[63], kgp2[59], kgp3[63]);
KGP_MODIFICATION kgp_modification362(kgp2[62], kgp2[58], kgp3[62]);
KGP_MODIFICATION kgp_modification361(kgp2[61], kgp2[57], kgp3[61]);
KGP_MODIFICATION kgp_modification360(kgp2[60], kgp2[56], kgp3[60]);
KGP_MODIFICATION kgp_modification359(kgp2[59], kgp2[55], kgp3[59]);
KGP_MODIFICATION kgp_modification358(kgp2[58], kgp2[54], kgp3[58]);
KGP_MODIFICATION kgp_modification357(kgp2[57], kgp2[53], kgp3[57]);
KGP_MODIFICATION kgp_modification356(kgp2[56], kgp2[52], kgp3[56]);
KGP_MODIFICATION kgp_modification355(kgp2[55], kgp2[51], kgp3[55]);
KGP_MODIFICATION kgp_modification354(kgp2[54], kgp2[50], kgp3[54]);
KGP_MODIFICATION kgp_modification353(kgp2[53], kgp2[49], kgp3[53]);
KGP_MODIFICATION kgp_modification352(kgp2[52], kgp2[48], kgp3[52]);
KGP_MODIFICATION kgp_modification351(kgp2[51], kgp2[47], kgp3[51]);
KGP_MODIFICATION kgp_modification350(kgp2[50], kgp2[46], kgp3[50]);
KGP_MODIFICATION kgp_modification349(kgp2[49], kgp2[45], kgp3[49]);
KGP_MODIFICATION kgp_modification348(kgp2[48], kgp2[44], kgp3[48]);
KGP_MODIFICATION kgp_modification347(kgp2[47], kgp2[43], kgp3[47]);
KGP_MODIFICATION kgp_modification346(kgp2[46], kgp2[42], kgp3[46]);
```

KGP_MODIFICATION kgp_modification345(kgp2[45], kgp2[41], kgp3[45]);
KGP_MODIFICATION kgp_modification344(kgp2[44], kgp2[40], kgp3[44]);
KGP_MODIFICATION kgp_modification343(kgp2[43], kgp2[39], kgp3[43]);
KGP_MODIFICATION kgp_modification342(kgp2[42], kgp2[38], kgp3[42]);
KGP_MODIFICATION kgp_modification341(kgp2[41], kgp2[37], kgp3[41]);
KGP_MODIFICATION kgp_modification340(kgp2[40], kgp2[36], kgp3[40]);
KGP_MODIFICATION kgp_modification339(kgp2[39], kgp2[35], kgp3[39]);
KGP_MODIFICATION kgp_modification338(kgp2[38], kgp2[34], kgp3[38]);
KGP_MODIFICATION kgp_modification337(kgp2[37], kgp2[33], kgp3[37]);
KGP_MODIFICATION kgp_modification336(kgp2[36], kgp2[32], kgp3[36]);
KGP_MODIFICATION kgp_modification335(kgp2[35], kgp2[31], kgp3[35]);
KGP_MODIFICATION kgp_modification334(kgp2[34], kgp2[30], kgp3[34]);
KGP_MODIFICATION kgp_modification333(kgp2[33], kgp2[29], kgp3[33]);
KGP_MODIFICATION kgp_modification332(kgp2[32], kgp2[28], kgp3[32]);
KGP_MODIFICATION kgp_modification331(kgp2[31], kgp2[27], kgp3[31]);
KGP_MODIFICATION kgp_modification330(kgp2[30], kgp2[26], kgp3[30]);
KGP_MODIFICATION kgp_modification329(kgp2[29], kgp2[25], kgp3[29]);
KGP_MODIFICATION kgp_modification328(kgp2[28], kgp2[24], kgp3[28]);
KGP_MODIFICATION kgp_modification327(kgp2[27], kgp2[23], kgp3[27]);
KGP_MODIFICATION kgp_modification326(kgp2[26], kgp2[22], kgp3[26]);
KGP_MODIFICATION kgp_modification325(kgp2[25], kgp2[21], kgp3[25]);
KGP_MODIFICATION kgp_modification324(kgp2[24], kgp2[20], kgp3[24]);
KGP_MODIFICATION kgp_modification323(kgp2[23], kgp2[19], kgp3[23]);
KGP_MODIFICATION kgp_modification322(kgp2[22], kgp2[18], kgp3[22]);
KGP_MODIFICATION kgp_modification321(kgp2[21], kgp2[17], kgp3[21]);
KGP_MODIFICATION kgp_modification320(kgp2[20], kgp2[16], kgp3[20]);
KGP_MODIFICATION kgp_modification319(kgp2[19], kgp2[15], kgp3[19]);
KGP_MODIFICATION kgp_modification318(kgp2[18], kgp2[14], kgp3[18]);
KGP_MODIFICATION kgp_modification317(kgp2[17], kgp2[13], kgp3[17]);
KGP_MODIFICATION kgp_modification316(kgp2[16], kgp2[12], kgp3[16]);
KGP_MODIFICATION kgp_modification315(kgp2[15], kgp2[11], kgp3[15]);

```
KGP_MODIFICATION kgp_modification314(kgp2[14], kgp2[10], kgp3[14]);
KGP_MODIFICATION kgp_modification313(kgp2[13], kgp2[9], kgp3[13]);
KGP_MODIFICATION kgp_modification312(kgp2[12], kgp2[8], kgp3[12]);
KGP_MODIFICATION kgp_modification311(kgp2[11], kgp2[7], kgp3[11]);
KGP_MODIFICATION kgp_modification310(kgp2[10], kgp2[6], kgp3[10]);
KGP_MODIFICATION kgp_modification39(kgp2[9], kgp2[5], kgp3[9]);
KGP_MODIFICATION kgp_modification38(kgp2[8], kgp2[4], kgp3[8]);
KGP_MODIFICATION kgp_modification37(kgp2[7], kgp2[3], kgp3[7]);
KGP_MODIFICATION kgp_modification36(kgp2[6], kgp2[2], kgp3[6]);
KGP_MODIFICATION kgp_modification35(kgp2[5], kgp2[1], kgp3[5]);
KGP_MODIFICATION kgp_modification34(kgp2[4], kgp2[0], kgp3[4]);
assign kgp3[3] = kgp2[3];
assign kgp3[2] = kgp2[2];
assign kgp3[1] = kgp2[1];
assign kgp3[0] = kgp2[0];
```

// Stage 4

```
KGP_MODIFICATION kgp_modification463(kgp3[63], kgp3[55], kgp4[63]);
KGP_MODIFICATION kgp_modification462(kgp3[62], kgp3[54], kgp4[62]);
KGP_MODIFICATION kgp_modification461(kgp3[61], kgp3[53], kgp4[61]);
KGP_MODIFICATION kgp_modification460(kgp3[60], kgp3[52], kgp4[60]);
KGP_MODIFICATION kgp_modification459(kgp3[59], kgp3[51], kgp4[59]);
KGP_MODIFICATION kgp_modification458(kgp3[58], kgp3[50], kgp4[58]);
KGP_MODIFICATION kgp_modification457(kgp3[57], kgp3[49], kgp4[57]);
KGP_MODIFICATION kgp_modification456(kgp3[56], kgp3[48], kgp4[56]);
KGP_MODIFICATION kgp_modification455(kgp3[55], kgp3[47], kgp4[55]);
KGP_MODIFICATION kgp_modification454(kgp3[54], kgp3[46], kgp4[54]);
KGP_MODIFICATION kgp_modification453(kgp3[53], kgp3[45], kgp4[53]);
KGP_MODIFICATION kgp_modification452(kgp3[52], kgp3[44], kgp4[52]);
KGP_MODIFICATION kgp_modification451(kgp3[51], kgp3[43], kgp4[51]);
KGP_MODIFICATION kgp_modification450(kgp3[50], kgp3[42], kgp4[50]);
```

KGP_MODIFICATION kgp_modification449(kgp3[49], kgp3[41], kgp4[49]);
KGP_MODIFICATION kgp_modification448(kgp3[48], kgp3[40], kgp4[48]);
KGP_MODIFICATION kgp_modification447(kgp3[47], kgp3[39], kgp4[47]);
KGP_MODIFICATION kgp_modification446(kgp3[46], kgp3[38], kgp4[46]);
KGP_MODIFICATION kgp_modification445(kgp3[45], kgp3[37], kgp4[45]);
KGP_MODIFICATION kgp_modification444(kgp3[44], kgp3[36], kgp4[44]);
KGP_MODIFICATION kgp_modification443(kgp3[43], kgp3[35], kgp4[43]);
KGP_MODIFICATION kgp_modification442(kgp3[42], kgp3[34], kgp4[42]);
KGP_MODIFICATION kgp_modification441(kgp3[41], kgp3[33], kgp4[41]);
KGP_MODIFICATION kgp_modification440(kgp3[40], kgp3[32], kgp4[40]);
KGP_MODIFICATION kgp_modification439(kgp3[39], kgp3[31], kgp4[39]);
KGP_MODIFICATION kgp_modification438(kgp3[38], kgp3[30], kgp4[38]);
KGP_MODIFICATION kgp_modification437(kgp3[37], kgp3[29], kgp4[37]);
KGP_MODIFICATION kgp_modification436(kgp3[36], kgp3[28], kgp4[36]);
KGP_MODIFICATION kgp_modification435(kgp3[35], kgp3[27], kgp4[35]);
KGP_MODIFICATION kgp_modification434(kgp3[34], kgp3[26], kgp4[34]);
KGP_MODIFICATION kgp_modification433(kgp3[33], kgp3[25], kgp4[33]);
KGP_MODIFICATION kgp_modification432(kgp3[32], kgp3[24], kgp4[32]);
KGP_MODIFICATION kgp_modification431(kgp3[31], kgp3[23], kgp4[31]);
KGP_MODIFICATION kgp_modification430(kgp3[30], kgp3[22], kgp4[30]);
KGP_MODIFICATION kgp_modification429(kgp3[29], kgp3[21], kgp4[29]);
KGP_MODIFICATION kgp_modification428(kgp3[28], kgp3[20], kgp4[28]);
KGP_MODIFICATION kgp_modification427(kgp3[27], kgp3[19], kgp4[27]);
KGP_MODIFICATION kgp_modification426(kgp3[26], kgp3[18], kgp4[26]);
KGP_MODIFICATION kgp_modification425(kgp3[25], kgp3[17], kgp4[25]);
KGP_MODIFICATION kgp_modification424(kgp3[24], kgp3[16], kgp4[24]);
KGP_MODIFICATION kgp_modification423(kgp3[23], kgp3[15], kgp4[23]);
KGP_MODIFICATION kgp_modification422(kgp3[22], kgp3[14], kgp4[22]);
KGP_MODIFICATION kgp_modification421(kgp3[21], kgp3[13], kgp4[21]);
KGP_MODIFICATION kgp_modification420(kgp3[20], kgp3[12], kgp4[20]);
KGP_MODIFICATION kgp_modification419(kgp3[19], kgp3[11], kgp4[19]);

```
KGP_MODIFICATION kgp_modification418(kgp3[18], kgp3[10], kgp4[18]);
KGP_MODIFICATION kgp_modification417(kgp3[17], kgp3[9], kgp4[17]);
KGP_MODIFICATION kgp_modification416(kgp3[16], kgp3[8], kgp4[16]);
KGP_MODIFICATION kgp_modification415(kgp3[15], kgp3[7], kgp4[15]);
KGP_MODIFICATION kgp_modification414(kgp3[14], kgp3[6], kgp4[14]);
KGP_MODIFICATION kgp_modification413(kgp3[13], kgp3[5], kgp4[13]);
KGP_MODIFICATION kgp_modification412(kgp3[12], kgp3[4], kgp4[12]);
KGP_MODIFICATION kgp_modification411(kgp3[11], kgp3[3], kgp4[11]);
KGP_MODIFICATION kgp_modification410(kgp3[10], kgp3[2], kgp4[10]);
KGP_MODIFICATION kgp_modification49(kgp3[9], kgp3[1], kgp4[9]);
KGP_MODIFICATION kgp_modification48(kgp3[8], kgp3[0], kgp4[8]);
assign kgp4[7] = kgp3[7];
assign kgp4[6] = kgp3[6];
assign kgp4[5] = kgp3[5];
assign kgp4[4] = kgp3[4];
assign kgp4[3] = kgp3[3];
assign kgp4[2] = kgp3[2];
assign kgp4[1] = kgp3[1];
assign kgp4[0] = kgp3[0];
```

// Stage 5

```
KGP_MODIFICATION kgp_modification563(kgp4[63], kgp4[47], kgp5[63]);
KGP_MODIFICATION kgp_modification562(kgp4[62], kgp4[46], kgp5[62]);
KGP_MODIFICATION kgp_modification561(kgp4[61], kgp4[45], kgp5[61]);
KGP_MODIFICATION kgp_modification560(kgp4[60], kgp4[44], kgp5[60]);
KGP_MODIFICATION kgp_modification559(kgp4[59], kgp4[43], kgp5[59]);
KGP_MODIFICATION kgp_modification558(kgp4[58], kgp4[42], kgp5[58]);
KGP_MODIFICATION kgp_modification557(kgp4[57], kgp4[41], kgp5[57]);
KGP_MODIFICATION kgp_modification556(kgp4[56], kgp4[40], kgp5[56]);
KGP_MODIFICATION kgp_modification555(kgp4[55], kgp4[39], kgp5[55]);
KGP_MODIFICATION kgp_modification554(kgp4[54], kgp4[38], kgp5[54]);
```

KGP_MODIFICATION kgp_modification553(kgp4[53], kgp4[37], kgp5[53]);
KGP_MODIFICATION kgp_modification552(kgp4[52], kgp4[36], kgp5[52]);
KGP_MODIFICATION kgp_modification551(kgp4[51], kgp4[35], kgp5[51]);
KGP_MODIFICATION kgp_modification550(kgp4[50], kgp4[34], kgp5[50]);
KGP_MODIFICATION kgp_modification549(kgp4[49], kgp4[33], kgp5[49]);
KGP_MODIFICATION kgp_modification548(kgp4[48], kgp4[32], kgp5[48]);
KGP_MODIFICATION kgp_modification547(kgp4[47], kgp4[31], kgp5[47]);
KGP_MODIFICATION kgp_modification546(kgp4[46], kgp4[30], kgp5[46]);
KGP_MODIFICATION kgp_modification545(kgp4[45], kgp4[29], kgp5[45]);
KGP_MODIFICATION kgp_modification544(kgp4[44], kgp4[28], kgp5[44]);
KGP_MODIFICATION kgp_modification543(kgp4[43], kgp4[27], kgp5[43]);
KGP_MODIFICATION kgp_modification542(kgp4[42], kgp4[26], kgp5[42]);
KGP_MODIFICATION kgp_modification541(kgp4[41], kgp4[25], kgp5[41]);
KGP_MODIFICATION kgp_modification540(kgp4[40], kgp4[24], kgp5[40]);
KGP_MODIFICATION kgp_modification539(kgp4[39], kgp4[23], kgp5[39]);
KGP_MODIFICATION kgp_modification538(kgp4[38], kgp4[22], kgp5[38]);
KGP_MODIFICATION kgp_modification537(kgp4[37], kgp4[21], kgp5[37]);
KGP_MODIFICATION kgp_modification536(kgp4[36], kgp4[20], kgp5[36]);
KGP_MODIFICATION kgp_modification535(kgp4[35], kgp4[19], kgp5[35]);
KGP_MODIFICATION kgp_modification534(kgp4[34], kgp4[18], kgp5[34]);
KGP_MODIFICATION kgp_modification533(kgp4[33], kgp4[17], kgp5[33]);
KGP_MODIFICATION kgp_modification532(kgp4[32], kgp4[16], kgp5[32]);
KGP_MODIFICATION kgp_modification531(kgp4[31], kgp4[15], kgp5[31]);
KGP_MODIFICATION kgp_modification530(kgp4[30], kgp4[14], kgp5[30]);
KGP_MODIFICATION kgp_modification529(kgp4[29], kgp4[13], kgp5[29]);
KGP_MODIFICATION kgp_modification528(kgp4[28], kgp4[12], kgp5[28]);
KGP_MODIFICATION kgp_modification527(kgp4[27], kgp4[11], kgp5[27]);
KGP_MODIFICATION kgp_modification526(kgp4[26], kgp4[10], kgp5[26]);
KGP_MODIFICATION kgp_modification525(kgp4[25], kgp4[9], kgp5[25]);
KGP_MODIFICATION kgp_modification524(kgp4[24], kgp4[8], kgp5[24]);
KGP_MODIFICATION kgp_modification523(kgp4[23], kgp4[7], kgp5[23]);


```
KGP_MODIFICATION kgp_modification522(kgp4[22], kgp4[6], kgp5[22]);
KGP_MODIFICATION kgp_modification521(kgp4[21], kgp4[5], kgp5[21]);
KGP_MODIFICATION kgp_modification520(kgp4[20], kgp4[4], kgp5[20]);
KGP_MODIFICATION kgp_modification519(kgp4[19], kgp4[3], kgp5[19]);
KGP_MODIFICATION kgp_modification518(kgp4[18], kgp4[2], kgp5[18]);
KGP_MODIFICATION kgp_modification517(kgp4[17], kgp4[1], kgp5[17]);
KGP_MODIFICATION kgp_modification516(kgp4[16], kgp4[0], kgp5[16]);

assign kgp5[15] = kgp4[15];
assign kgp5[14] = kgp4[14];
assign kgp5[13] = kgp4[13];
assign kgp5[12] = kgp4[12];
assign kgp5[11] = kgp4[11];
assign kgp5[10] = kgp4[10];
assign kgp5[9] = kgp4[9];
assign kgp5[8] = kgp4[8];
assign kgp5[7] = kgp4[7];
assign kgp5[6] = kgp4[6];
assign kgp5[5] = kgp4[5];
assign kgp5[4] = kgp4[4];
assign kgp5[3] = kgp4[3];
assign kgp5[2] = kgp4[2];
assign kgp5[1] = kgp4[1];
assign kgp5[0] = kgp4[0];
```

```
// Stage 6
```

```
KGP_MODIFICATION kgp_modification663(kgp5[63], kgp5[31], kgp6[63]);
KGP_MODIFICATION kgp_modification662(kgp5[62], kgp5[30], kgp6[62]);
KGP_MODIFICATION kgp_modification661(kgp5[61], kgp5[29], kgp6[61]);
KGP_MODIFICATION kgp_modification660(kgp5[60], kgp5[28], kgp6[60]);
KGP_MODIFICATION kgp_modification659(kgp5[59], kgp5[27], kgp6[59]);
KGP_MODIFICATION kgp_modification658(kgp5[58], kgp5[26], kgp6[58]);
```

```
KGP_MODIFICATION kgp_modification657(kgp5[57], kgp5[25], kgp6[57]);
KGP_MODIFICATION kgp_modification656(kgp5[56], kgp5[24], kgp6[56]);
KGP_MODIFICATION kgp_modification655(kgp5[55], kgp5[23], kgp6[55]);
KGP_MODIFICATION kgp_modification654(kgp5[54], kgp5[22], kgp6[54]);
KGP_MODIFICATION kgp_modification653(kgp5[53], kgp5[21], kgp6[53]);
KGP_MODIFICATION kgp_modification652(kgp5[52], kgp5[20], kgp6[52]);
KGP_MODIFICATION kgp_modification651(kgp5[51], kgp5[19], kgp6[51]);
KGP_MODIFICATION kgp_modification650(kgp5[50], kgp5[18], kgp6[50]);
KGP_MODIFICATION kgp_modification649(kgp5[49], kgp5[17], kgp6[49]);
KGP_MODIFICATION kgp_modification648(kgp5[48], kgp5[16], kgp6[48]);
KGP_MODIFICATION kgp_modification647(kgp5[47], kgp5[15], kgp6[47]);
KGP_MODIFICATION kgp_modification646(kgp5[46], kgp5[14], kgp6[46]);
KGP_MODIFICATION kgp_modification645(kgp5[45], kgp5[13], kgp6[45]);
KGP_MODIFICATION kgp_modification644(kgp5[44], kgp5[12], kgp6[44]);
KGP_MODIFICATION kgp_modification643(kgp5[43], kgp5[11], kgp6[43]);
KGP_MODIFICATION kgp_modification642(kgp5[42], kgp5[10], kgp6[42]);
KGP_MODIFICATION kgp_modification641(kgp5[41], kgp5[9], kgp6[41]);
KGP_MODIFICATION kgp_modification640(kgp5[40], kgp5[8], kgp6[40]);
KGP_MODIFICATION kgp_modification639(kgp5[39], kgp5[7], kgp6[39]);
KGP_MODIFICATION kgp_modification638(kgp5[38], kgp5[6], kgp6[38]);
KGP_MODIFICATION kgp_modification637(kgp5[37], kgp5[5], kgp6[37]);
KGP_MODIFICATION kgp_modification636(kgp5[36], kgp5[4], kgp6[36]);
KGP_MODIFICATION kgp_modification635(kgp5[35], kgp5[3], kgp6[35]);
KGP_MODIFICATION kgp_modification634(kgp5[34], kgp5[2], kgp6[34]);
KGP_MODIFICATION kgp_modification633(kgp5[33], kgp5[1], kgp6[33]);
KGP_MODIFICATION kgp_modification632(kgp5[32], kgp5[0], kgp6[32]);

assign kgp6[31] = kgp5[31];
assign kgp6[30] = kgp5[30];
assign kgp6[29] = kgp5[29];
assign kgp6[28] = kgp5[28];
assign kgp6[27] = kgp5[27];
```

```
assign kgp6[26] = kgp5[26];
assign kgp6[25] = kgp5[25];
assign kgp6[24] = kgp5[24];
assign kgp6[23] = kgp5[23];
assign kgp6[22] = kgp5[22];
assign kgp6[21] = kgp5[21];
assign kgp6[20] = kgp5[20];
assign kgp6[19] = kgp5[19];
assign kgp6[18] = kgp5[18];
assign kgp6[17] = kgp5[17];
assign kgp6[16] = kgp5[16];
assign kgp6[15] = kgp5[15];
assign kgp6[14] = kgp5[14];
assign kgp6[13] = kgp5[13];
assign kgp6[12] = kgp5[12];
assign kgp6[11] = kgp5[11];
assign kgp6[10] = kgp5[10];
assign kgp6[9] = kgp5[9];
assign kgp6[8] = kgp5[8];
assign kgp6[7] = kgp5[7];
assign kgp6[6] = kgp5[6];
assign kgp6[5] = kgp5[5];
assign kgp6[4] = kgp5[4];
assign kgp6[3] = kgp5[3];
assign kgp6[2] = kgp5[2];
assign kgp6[1] = kgp5[1];
assign kgp6[0] = kgp5[0];
```

```
generate
```

```
  for (i = 0; i<64; i=i+1) begin
```

```
    assign Cin[i] = kgp6[i][1];
```

```
end  
endgenerate
```

```
FA fa1(a[0], b[0], c_input, sum[0]);  
FA fa2(a[1], b[1], Cin[0], sum[1]);  
FA fa3(a[2], b[2], Cin[1], sum[2]);  
FA fa4(a[3], b[3], Cin[2], sum[3]);  
FA fa5(a[4], b[4], Cin[3], sum[4]);  
FA fa6(a[5], b[5], Cin[4], sum[5]);  
FA fa7(a[6], b[6], Cin[5], sum[6]);  
FA fa8(a[7], b[7], Cin[6], sum[7]);  
FA fa9(a[8], b[8], Cin[7], sum[8]);  
FA fa10(a[9], b[9], Cin[8], sum[9]);  
FA fa11(a[10], b[10], Cin[9], sum[10]);  
FA fa12(a[11], b[11], Cin[10], sum[11]);  
FA fa13(a[12], b[12], Cin[11], sum[12]);  
FA fa14(a[13], b[13], Cin[12], sum[13]);  
FA fa15(a[14], b[14], Cin[13], sum[14]);  
FA fa16(a[15], b[15], Cin[14], sum[15]);  
FA fa17(a[16], b[16], Cin[15], sum[16]);  
FA fa18(a[17], b[17], Cin[16], sum[17]);  
FA fa19(a[18], b[18], Cin[17], sum[18]);  
FA fa20(a[19], b[19], Cin[18], sum[19]);  
FA fa21(a[20], b[20], Cin[19], sum[20]);  
FA fa22(a[21], b[21], Cin[20], sum[21]);  
FA fa23(a[22], b[22], Cin[21], sum[22]);  
FA fa24(a[23], b[23], Cin[22], sum[23]);  
FA fa25(a[24], b[24], Cin[23], sum[24]);  
FA fa26(a[25], b[25], Cin[24], sum[25]);  
FA fa27(a[26], b[26], Cin[25], sum[26]);
```

FA fa28(a[27], b[27], Cin[26], sum[27]);
FA fa29(a[28], b[28], Cin[27], sum[28]);
FA fa30(a[29], b[29], Cin[28], sum[29]);
FA fa31(a[30], b[30], Cin[29], sum[30]);
FA fa32(a[31], b[31], Cin[30], sum[31]);
FA fa33(a[32], b[32], Cin[31], sum[32]);
FA fa34(a[33], b[33], Cin[32], sum[33]);
FA fa35(a[34], b[34], Cin[33], sum[34]);
FA fa36(a[35], b[35], Cin[34], sum[35]);
FA fa37(a[36], b[36], Cin[35], sum[36]);
FA fa38(a[37], b[37], Cin[36], sum[37]);
FA fa39(a[38], b[38], Cin[37], sum[38]);
FA fa40(a[39], b[39], Cin[38], sum[39]);
FA fa41(a[40], b[40], Cin[39], sum[40]);
FA fa42(a[41], b[41], Cin[40], sum[41]);
FA fa43(a[42], b[42], Cin[41], sum[42]);
FA fa44(a[43], b[43], Cin[42], sum[43]);
FA fa45(a[44], b[44], Cin[43], sum[44]);
FA fa46(a[45], b[45], Cin[44], sum[45]);
FA fa47(a[46], b[46], Cin[45], sum[46]);
FA fa48(a[47], b[47], Cin[46], sum[47]);
FA fa49(a[48], b[48], Cin[47], sum[48]);
FA fa50(a[49], b[49], Cin[48], sum[49]);
FA fa51(a[50], b[50], Cin[49], sum[50]);
FA fa52(a[51], b[51], Cin[50], sum[51]);
FA fa53(a[52], b[52], Cin[51], sum[52]);
FA fa54(a[53], b[53], Cin[52], sum[53]);
FA fa55(a[54], b[54], Cin[53], sum[54]);
FA fa56(a[55], b[55], Cin[54], sum[55]);
FA fa57(a[56], b[56], Cin[55], sum[56]);
FA fa58(a[57], b[57], Cin[56], sum[57]);

```
FA fa59(a[58], b[58], Cin[57], sum[58]);
FA fa60(a[59], b[59], Cin[58], sum[59]);
FA fa61(a[60], b[60], Cin[59], sum[60]);
FA fa62(a[61], b[61], Cin[60], sum[61]);
FA fa63(a[62], b[62], Cin[61], sum[62]);
FA fa64(a[63], b[63], Cin[62], sum[63]);
```

```
assign carry = Cin[63];
```

```
endmodule
```

```
module RCDA_TB;
```

```
reg[63:0] a, b;
```

```
wire[63:0] sum;
```

```
wire carry;
```

```
RCDA rcda1(a, b, 1'b0, sum, carry);
```

```
initial begin
```

```
#0 a=64'b0010010101010110110110111010000100010001100100010100010001011010;
b=64'b0111000110000110100110000110000111011110110111100111001110111011;
```

```
#10 a=64'b1101100111101111000101001000101001011011101100001110001100001100;
b=64'b0111010010010101010000111011110010111001110001100000000001001000;
```

```
// a=2690579312231400633 b=8180393319283454732 sum=10870972631514855445
```

```
#10 a=64'b1010010101010110110110111010000100010001100100010100010001011010;
b=64'b0111000110000110100110000110000111011110110111100111001110111011;
```

```
// a=15703793010140570380 b=8400695157779202120
```

```
#10 a=64'b0101100111101111000101001000101001011011101100001110001100001101;  
b=64'b011101001001010101000011101111001011100111000110000000001001010;
```

```
#10 a=64'b1111111111111111111111111111111111111111111111111111111111111111;  
b=64'b1111111111111111111111111111111111111111111111111111111111111111;
```

```
#10;
```

```
End
```

```
initial begin
```

```
$monitor(" in1=%b\n in2=%b\n sum=%b\n carry=%b", a, b, sum, carry);
```

```
$dumpfile("rcda.vcd");
```

```
$dumpvars(0, rcda1);
```

```
end
```

```
endmodule
```

Output:

```
C:\Users\rammo\OneDrive\Documents\CA\Lab\Lab3>iverilog -o rdca_str rdca_str.v  
C:\Users\rammo\OneDrive\Documents\CA\Lab\Lab3>vvp rdca_str  
VCD info: dumpfile rcda.vcd opened for output.  
in1=0010010101010110110110111010000100010001100100010100010001011010  
in2=0111000110000110100110000110000111011110110111100111001110111011  
sum=10010110110111010111010000000010111100000110111110111100000010101  
carry=0  
in1=1101100111101111000101001000101001011011101100001110001100001100  
in2=0111010010010101010000111011110010111001110001100000000001001000  
sum=0100111010000100010110000100011100010101011101101110001101010100  
carry=1  
in1=1010010101010110110111010000100010001100100010100010001011010  
in2=0111000110000110100110000110000111011110110111100111001110111011  
sum=00010110110111010111010000000010111100000110111110111100000010101  
carry=1  
in1=0101100111101111000101001000101001011011101100001110001100001101  
in2=0111010010010101010000111011110010111001110001100000000001001010  
sum=1100111010000100010110000100011100010101011101101110001101010111  
carry=0  
in1=1111111111111111111111111111111111111111111111111111111111111111  
in2=1111111111111111111111111111111111111111111111111111111111111111  
sum=1111111111111111111111111111111111111111111111111111111111111110  
carry=1  
C:\Users\rammo\OneDrive\Documents\CA\Lab\Lab3>gtkwave rdca.vcd
```

Wave:

Signals	Waves					
Time	9 sec		18 sec		27 sec	
a[63:0]	2556DBA11191445A	D9EF148A5BB0E30C	A556DBA11191445A	59EF148A5BB0E30D	FFFFFFFFFFFFFFFF	
b[63:0]	71869861DEDE73BB	749543BCB9C60048	71869861DEDE73BB	749543BCB9C6004A	FFFFFFFFFFFFFFFF	
sum[63:0]	96DD7402F06FB915	4E8458471576E354	16DD7402F06FB915	CE8458471576E357	FFFFFFFFFFFFFFFE	
carry						