Lab Report 4

Goal:

The objective of this lab is to design a complex arithmetic logic block using Verilog.

Steps:

- Made a pseudo Verilog code for log multiplier shown in the prelab.
- Made $\log_{10}$ look up table using case statements in Verilog.
- Used Cadence to verify the functionality of the log multiplier.
- Used log table to get the log of A [4:0] and B [4:0] and added the output of those to get the multiplication as $\log_{10} A * \log_{10} B = \log_{10} A + \log_{10} B$. The output of summation is stored in sum.
- Using the output of sum for all the combinations, a ROM look up table was prepared to get the output for antilog of a number and is stored in output [20:0].
- Simulated the timing diagram on Quartus.
- After simulating the design on the Quartus, to implement it on the Altera FPGA we assign pins according to the Alter specifications.
- Then loaded the code on the Altera board, and verified the output of the log multiplier.

Code:

```
module logmul(A,B,O);
input [4:0] A;
input [4:0] B;
output [20:0] O;
reg [20:0] O;
reg [8:0] log10a, log10b;
reg [9:0] sum;
always@(A or B)
begin
case(A)
5'b00001: log10a = 9'b000000000;
5'b00010: log10a = 9'b000011110;
5'b00011: log10a = 9'b000110000;
5'b00100: log10a = 9'b000111100;
5'b00101: log10a = 9'b001000110;
5'b00110: log10a = 9'b001001110;
5'b00111: log10a = 9'b001010101;
5'b01000: log10a = 9'b001011010;
5'b01001: log10a = 9'b001011111;
5'b01010: log10a = 9'b001100100;
5'b01011: log10a = 9'b001101000;
5'b01100: log10a = 9'b001101100;
5'b01101: log10a = 9'b001101111;
5'b01110: log10a = 9'b001110011;
5'b01111: log10a = 9'b001110110;
5'b10000: log10a = 9'b001111000;
```

```
5'b10001: log10a = 9'b001111011;
5'b10010: log10a = 9'b001111101;
5'b10011: log10a = 9'b010000000;
5'b10100: log10a = 9'b010000010;
5'b10101: log10a = 9'b010000100;
5'b10110: log10a = 9'b010000110;
5'b10111: log10a = 9'b010001000;
5'b11000: log10a = 9'b010001010;
5'b11001: log10a = 9'b010001100;
5'b11010: log10a = 9'b010001110;
5'b11011: log10a = 9'b010001111;
5'b11100: log10a = 9'b010010001;
5'b11101: log10a = 9'b010010010;
5'b11110: log10a = 9'b010010100;
5'b11111: log10a = 9'b010010101;
endcase
case(B)
5'b00001: log10b = 9'b000000000;
5'b00010: log10b = 9'b000011110;
5'b00011: log10b = 9'b000110000;
5'b00100: log10b = 9'b000111100;
5'b00101: log10b = 9'b001000110;
5'b00110: log10b = 9'b001001110;
5'b00111: log10b = 9'b001010101;
5'b01000: log10b = 9'b001011010;
5'b01001: log10b = 9'b001011111;
5'b01010: log10b = 9'b001100100;
5'b01011: log10b = 9'b001101000;
5'b01100: log10b = 9'b001101100;
5'b01101: log10b = 9'b001101111;
5'b01110: log10b = 9'b001110011;
5'b01111: log10b = 9'b001110110;
5'b10000: log10b = 9'b001111000;
5'b10001: log10b = 9'b001111011;
5'b10010: log10b = 9'b001111101;
5'b10011: log10b = 9'b010000000;
5'b10100: log10b = 9'b010000010;
5'b10101: log10b = 9'b010000100;
5'b10110: log10b = 9'b010000110;
5'b10111: log10b = 9'b010001000;
5'b11000: log10b = 9'b010001010;
5'b11001: log10b = 9'b010001100;
5'b11010: log10b = 9'b010001110;
5'b11011: log10b = 9'b010001111;
5'b11100: log10b = 9'b010010001;
5'b11101: log10b = 9'b010010010;
5'b11110: log10b = 9'b010010100;
5'b11111: log10b = 9'b010010101;
endcase
sum = (log10a + log10b);
end
always@(log10a or log10b)
begin
case(sum)
```

9'd000:O=21'b100000010000001111001;
9'd001:O=21'b100000010000001111001;
9'd002:O=21'b100000010000001111001;
9'd003:O=21'b100000010000001111001;
9'd004:O=21'b100000010000001111001;
9'd005:O=21'b100000010000001111001;
9'd006:O=21'b100000010000001111001;
9'd007:O=21'b100000010000001111001;
9'd008:O=21'b100000010000001111001;
9'd009:O=21'b100000010000001111001;
9'd010:O=21'b100000010000001111001;
9'd011:O=21'b100000010000001111001;
9'd012:O=21'b100000010000001111001;
9'd013:O=21'b100000010000001111001;
9'd014:O=21'b100000010000001111001;
9'd015:O=21'b100000010000001111001;
9'd016:O=21'b100000010000001111001;
9'd017:O=21'b100000010000001111001;
9'd018:O=21'b100000010000001111001;
9'd019:O=21'b100000010000001111001;
9'd020:O=21'b100000010000001111001;
9'd021:O=21'b100000010000001111001;
9'd022:O=21'b100000010000001111001;
9'd023:O=21'b100000010000001111001;
9'd024:O=21'b100000010000001111001;
9'd025:O=21'b100000010000001111001;
9'd026:O=21'b100000010000001111001;
9'd027:O=21'b100000010000001111001;
9'd028:O=21'b100000010000001111001;
9'd029:O=21'b100000010000001111001;
9'd030:O=21'b100000010000000100100;
9'd031:O=21'b100000010000000100100;
9'd032:O=21'b100000010000000100100;
9'd033:O=21'b100000010000000100100;
9'd034:O=21'b100000010000000100100;
9'd035:O=21'b100000010000000100100;
9'd036:O=21'b100000010000000100100;
9'd037:O=21'b100000010000000100100;
9'd038:O=21'b100000010000000100100;
9'd039:O=21'b100000010000000100100;
9'd040:O=21'b100000010000000100100;
9'd041:O=21'b100000010000000100100;
9'd042:O=21'b100000010000000100100;
9'd043:O=21'b100000010000000100100;
9'd044:O=21'b100000010000000100100;
9'd045:O=21'b100000010000000100100;
9'd046:O=21'b100000010000000100100;
9'd047:O=21'b100000010000000100100;
9'd048:O=21'b100000010000000110000;
9'd049:O=21'b100000010000000110000;
9'd050:O=21'b100000010000000110000;
9'd051:O=21'b100000010000000110000;
9'd052:O=21'b100000010000000110000;
9'd053:O=21'b100000010000000110000;

9'd054:O=21'b100000010000000110000;
9'd055:O=21'b100000010000000110000;
9'd056:O=21'b100000010000000110000;
9'd057:O=21'b100000010000000110000;
9'd058:O=21'b100000010000000110000;
9'd059:O=21'b100000010000000110000;
9'd060:O=21'b100000010000000011001;
9'd061:O=21'b100000010000000011001;
9'd062:O=21'b100000010000000011001;
9'd063:O=21'b100000010000000011001;
9'd064:O=21'b100000010000000011001;
9'd065:O=21'b100000010000000011001;
9'd066:O=21'b100000010000000011001;
9'd067:O=21'b100000010000000011001;
9'd068:O=21'b100000010000000011001;
9'd069:O=21'b100000010000000011001;
9'd070:O=21'b100000010000000010010;
9'd071:O=21'b100000010000000010010;
9'd072:O=21'b100000010000000010010;
9'd073:O=21'b100000010000000010010;
9'd074:O=21'b100000010000000010010;
9'd075:O=21'b100000010000000010010;
9'd076:O=21'b100000010000000010010;
9'd077:O=21'b100000010000000010010;
9'd078:O=21'b100000010000000000010;
9'd079:O=21'b100000010000000000010;
9'd080:O=21'b100000010000000000010;
9'd081:O=21'b100000010000000000010;
9'd082:O=21'b100000010000000000010;
9'd083:O=21'b100000010000000000010;
9'd084:O=21'b100000010000000000010;
9'd085:O=21'b100000010000001111000;
9'd086:O=21'b100000010000001111000;
9'd087:O=21'b100000010000001111000;
9'd088:O=21'b100000010000001111000;
9'd089:O=21'b100000010000001111000;
9'd090:O=21'b100000010000000000000;
9'd091:O=21'b100000010000000000000;
9'd092:O=21'b100000010000000000000;
9'd093:O=21'b100000010000000000000;
9'd094:O=21'b100000010000000000000;
9'd095:O=21'b100000010000000010000;
9'd096:O=21'b100000010000000010000;
9'd097:O=21'b100000010000000010000;
9'd098:O=21'b100000010000000010000;
9'd099:O=21'b100000010000000010000;
9'd100:O=21'b100000011111001000000;
9'd101:O=21'b100000011111001000000;
9'd102:O=21'b100000011111001000000;
9'd103:O=21'b100000011111001000000;
9'd104:O=21'b100000011110011111001;
9'd105:O=21'b100000011110011111001;
9'd106:O=21'b100000011110011111001;
9'd106:O=21'b100000011110011111001;

9'd107:O=21'b100000011110011111001;
9'd108:O=21'b100000011110010100100;
9'd109:O=21'b100000011110010100100;
9'd110:O=21'b100000011110010100100;
9'd111:O=21'b100000011110010110000;
9'd112:O=21'b100000011110010110000;
9'd113:O=21'b100000011110010110000;
9'd114:O=21'b100000011110010110000;
9'd115:O=21'b100000011110010011001;
9'd116:O=21'b100000011110010011001;
9'd117:O=21'b100000011110010011001;
9'd118:O=21'b100000011110010010010;
9'd119:O=21'b100000011110010010010;
9'd120:O=21'b100000011110010000010;
9'd121:O=21'b100000011110010000010;
9'd122:O=21'b100000011110010000010;
9'd123:O=21'b100000011110011111000;
9'd124:O=21'b100000011110011111000;
9'd125:O=21'b100000011110011111000;
9'd126:O=21'b100000011110010000000;
9'd127:O=21'b100000011110010000000;
9'd128:O=21'b100000011110010010000;
9'd129:O=21'b100000011110010010000;
9'd130:O=21'b100000011110011000000;
9'd131:O=21'b100000011110011000000;
9'd132:O=21'b100000011110010100100;
9'd133:O=21'b100000011110010100100;
9'd134:O=21'b100000011110011111001;
9'd135:O=21'b100000011110011111001;
9'd136:O=21'b100000011110010110000;
9'd137:O=21'b100000011110010110000;
9'd138:O=21'b100000011110010011001;
9'd139:O=21'b100000011110010011001;
9'd140:O=21'b100000011110010010010;
9'd141:O=21'b100000011110010000010;
9'd142:O=21'b100000011110010000010;
9'd143:O=21'b100000011110011111000;
9'd144:O=21'b100000011110011111000;
9'd145:O=21'b100000011110010000000;
9'd146:O=21'b100000011110010010000;
9'd147:O=21'b100000011110010010000;
9'd148:O=21'b100000001100001000000;
9'd149:O=21'b100000001100001111001;
9'd150:O=21'b100000001100001111001;
9'd151:O=21'b100000001100000100100;
9'd152:O=21'b100000001100000110000;
9'd153:O=21'b100000001100000011001;
9'd154:O=21'b100000001100000010010;
9'd155:O=21'b100000001100000010010;
9'd156:O=21'b100000001100000000010;
9'd157:O=21'b100000001100001111000;
9'd158:O=21'b100000001100000000000;
9'd159:O=21'b100000001100000010000;
9'd160:O=21'b100000000110011000000;

9'd161:O=21'b100000000110011111001;
9'd162:O=21'b100000000110010100100;
9'd163:O=21'b100000000110010110000;
9'd164:O=21'b100000000110010011001;
9'd165:O=21'b100000000110010010010;
9'd166:O=21'b100000000110010000010;
9'd167:O=21'b100000000110011111000;
9'd168:O=21'b100000000110010000000;
9'd169:O=21'b100000000110010010000;
9'd170:O=21'b100000000100101000000;
9'd171:O=21'b100000000100101111001;
9'd172:O=21'b100000000100100100100;
9'd173:O=21'b100000000100100110000;
9'd174:O=21'b100000000100100010010;
9'd175:O=21'b100000000100100000010;
9'd176:O=21'b100000000100101111000;
9'd177:O=21'b100000000100100010000;
9'd178:O=21'b100000000000101000000;
9'd179:O=21'b100000000000101111001;
9'd180:O=21'b100000000000100110000;
9'd181:O=21'b100000000000100011001;
9'd182:O=21'b100000000000100000010;
9'd183:O=21'b100000000000101111000;
9'd184:O=21'b100000000000100010000;
9'd185:O=21'b100000011110001000000;
9'd186:O=21'b100000011110000100100;
9'd187:O=21'b100000011110000011001;
9'd188:O=21'b100000011110000010010;
9'd189:O=21'b100000011110001111000;
9'd190:O=21'b100000011110000010000;
9'd191:O=21'b100000000000001111001;
9'd192:O=21'b100000000000000110000;
9'd193:O=21'b100000000000000010010;
9'd194:O=21'b100000000000001111000;
9'd195:O=21'b100000000000000010000;
9'd196:O=21'b100000000100001111001;
9'd197:O=21'b100000000100000110000;
9'd198:O=21'b100000000100000010010;
9'd199:O=21'b100000000100000000000;
9'd200:O=21'b111100110000001000000;
9'd201:O=21'b111100110000000100100;
9'd202:O=21'b111100110000000011001;
9'd203:O=21'b111100110000000000010;
9'd204:O=21'b111100110000000010000;
9'd205:O=21'b111100111110011111001;
9'd206:O=21'b111100111110010011001;
9'd207:O=21'b111100111110011111000;
9'd208:O=21'b111100111110010010000;
9'd209:O=21'b111100101001000100100;
9'd210:O=21'b111100101001000010010;
9'd211:O=21'b111100101001000000000;
9'd212:O=21'b111100101100001111001;
9'd213:O=21'b111100101100000011001;
9'd214:O=21'b111100101100001111000;

9'd215:O=21'b111100100110011000000;
9'd216:O=21'b111100100110010110000;
9'd217:O=21'b111100100110011111000;
9'd218:O=21'b111100100100101111001;
9'd219:O=21'b111100100100100011001;
9'd220:O=21'b111100100100101111000;
9'd221:O=21'b111100100000101111001;
9'd222:O=21'b111100100000100010010;
9'd223:O=21'b111100100000100000000;
9'd224:O=21'b111100111110000100100;
9'd225:O=21'b111100111110000000010;
9'd226:O=21'b111100100000001000000;
9'd227:O=21'b111100100000000010010;
9'd228:O=21'b111100100000000010000;
9'd229:O=21'b111100100100000110000;
9'd230:O=21'b111100100100000000000;
9'd231:O=21'b010010010000000100100;
9'd232:O=21'b010010010000001111000;
9'd233:O=21'b010010011110010100100;
9'd234:O=21'b010010011110011111000;
9'd235:O=21'b010010001001000100100;
9'd236:O=21'b010010001001001111000;
9'd237:O=21'b010010001100000100100;
9'd238:O=21'b010010001100000000000;
9'd239:O=21'b010010000110010110000;
9'd240:O=21'b010010000110010010000;
9'd241:O=21'b010010000100100010010;
9'd242:O=21'b010010000000101111001;
9'd243:O=21'b010010000000101111000;
9'd244:O=21'b010010011110000110000;
9'd245:O=21'b010010011110000010000;
9'd246:O=21'b010010000000000000010;
9'd247:O=21'b010010000100000100100;
9'd248:O=21'b010010000100000010000;
9'd249:O=21'b011000010000000000010;
9'd250:O=21'b011000011110010110000;
9'd251:O=21'b011000001001001000000;
9'd252:O=21'b011000001001000000000;
9'd253:O=21'b011000001100000010010;
9'd254:O=21'b011000000110010110000;
9'd255:O=21'b011000000100101111001;
9'd256:O=21'b011000000100100010000;
9'd257:O=21'b011000000000100000000;
9'd258:O=21'b011000011110000000010;
9'd259:O=21'b011000000000000010010;
9'd260:O=21'b011000000100000011001;
9'd261:O=21'b001100110000000110000;
9'd262:O=21'b001100111110010110000;
9'd263:O=21'b001100101001000100100;
9'd264:O=21'b001100101100000100100;
9'd265:O=21'b001100100110010100100;
9'd266:O=21'b001100100100100100100;
9'd267:O=21'b001100100000100110000;
9'd268:O=21'b001100111110000011001;

```
9'd269:O=21'b001100100000000010010;
9'd270:O=21'b001100100100000000010;
9'd271:O=21'b001001010000001111000;
9'd272:O=21'b001001011110010010000;
9'd273:O=21'b001001001100001111001;
9'd274:O=21'b001001000110010011001;
9'd275:O=21'b001001000100100000010;
9'd276:O=21'b001001000000100010000;
9'd277:O=21'b001001000000000110000;
9'd278:O=21'b001001000100000000010;
9'd279:O=21'b000001011110011000000;
9'd280:O=21'b000001001001000011001;
9'd281:O=21'b000001001100000010000;
9'd282:O=21'b000001000100100011001;
9'd283:O=21'b000001000000100010000;
9'd284:O=21'b000001000000000011001;
9'd285:O=21'b111100010000001000000;
9'd286:O=21'b111100011110011111000;
9'd287:O=21'b111100001100000110000;
9'd288:O=21'b111100000100101000000;
9'd289:O=21'b111100000000100000000;
9'd290:O=21'b111100000000000000010;
9'd291:O=21'b000000010000000011001;
9'd292:O=21'b000000001001000110000;
9'd293:O=21'b000000000110010100100;
9'd294:O=21'b000000000000101111001;
9'd295:O=21'b000000000000000100100;
9'd296:O=21'b001000010000000100100;
9'd297:O=21'b001000001001000110000;
9'd298:O=21'b001000000100100010010;
default:O=21'b100000010000001000000;
endcase
end
endmodule
```

Conclusion:

We learnt how to make a complex arithmetic logic block i.e. logarithmic multiplier using Verilog. In this lab we learnt structural and behavioral description of circuit is in Verilog HDL.

- In structural modeling we give the description of the entire circuit by specifying all the components used to build it. We also, define the interconnections between the components. After the 'begin' statement in the architecture, the function of the component in the circuit is defined using the inputs, signals and outputs. This model allows a hierarchical design. It is a textual replacement for any schematic.
- On the other hand, in behavioral model we use similar semantics and constructs like we use in C programming. In this we can use statements like if-else, for and while loops, case statements etc.