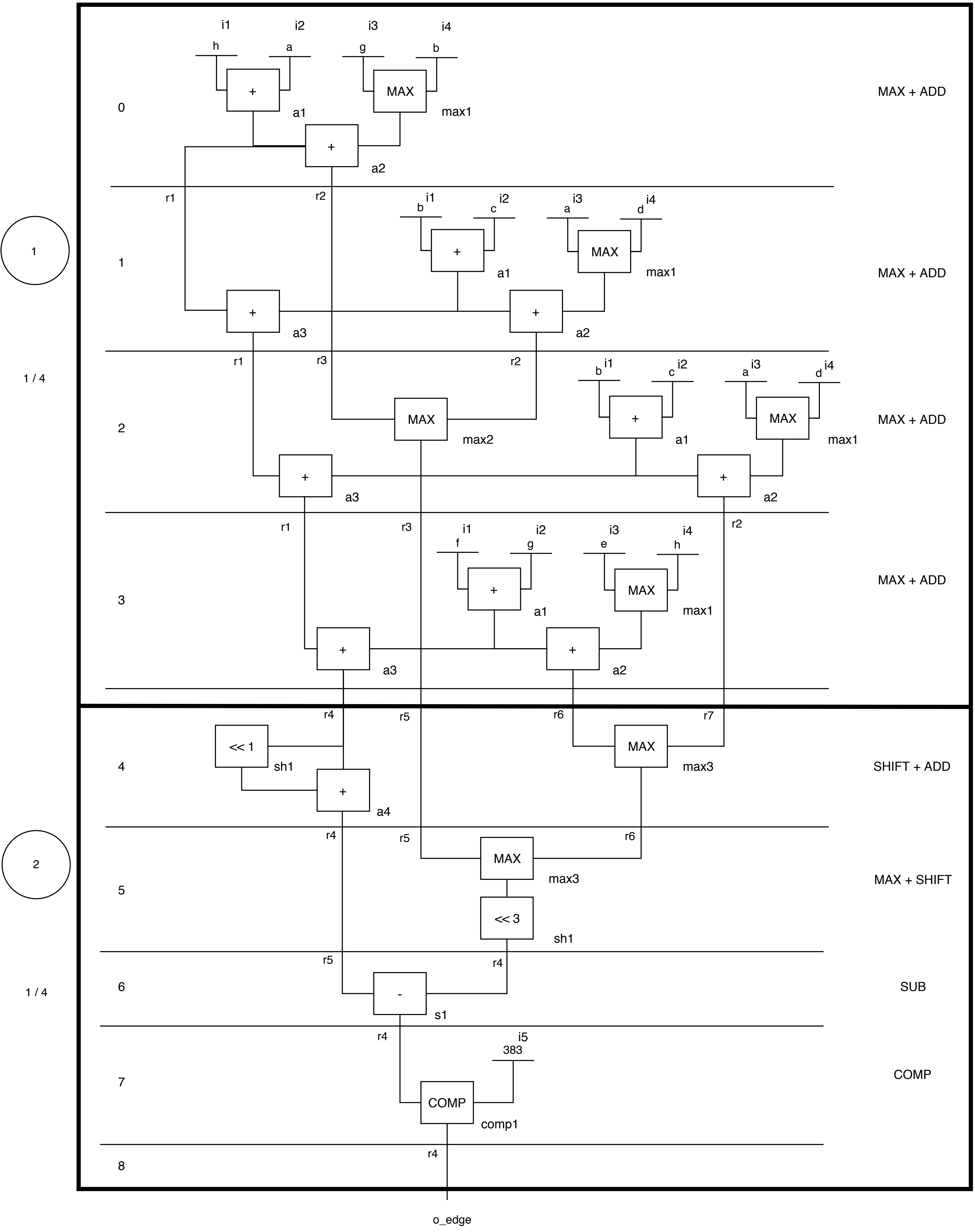

1
1 / 4
2
1 / 4
0
i1 i2 i3 i4
h a g b
+
MAX
a1
max1
+
a2
MAX + ADD
r1 r2
1
i1 i2 i3 i4
b c a d
+
MAX
a1
max1
+
a3
+
a2
MAX + ADD
r1 r3 r2
2
i1 i2 i3 i4
b c a d
MAX
max2
+
a1
MAX
max1
+
a3
+
a2
MAX + ADD
r1 r3 r2
3
i1 i2 i3 i4
f g e h
+
a1
MAX
max1
+
a3
+
a2
MAX + ADD
r4 r5 r6 r7
4
<< 1
sh1
+
a4
MAX
max3
SHIFT + ADD
r4 r5 r6
5
MAX
max3
<< 3
sh1
MAX + SHIFT
r5 r4
6
-
s1
SUB
r4
i5
383
7
COMP
comp1
COMP
r4
8
o_edge

Number of Components

ADD : 4	COMP : 1
MAX : 3	SHIFT : 1
SUB : 1	REG : 7

Throughput : min (1/4 , 1/4) = 1/4

Latency : 8

Equation:

max (ALL) = 8*(max (max(((h + a) + max(g,b)), ((b + c) + max(a,d))), max(((d + e) + max(f,c)), (f + g) + max(e,h)))) - 3*z

We obtained the optimized equation with the following steps:

1. Applied the "5a – 3b = 8a – 3(a + b)" optimization to all equations (of each direction)

2. Applied the " max(a – c, b – c) = max(a, b) – c" optimization to pairs of equations e.g N, NW

3. Applied the " max(a + b, b + c) = b + max(a, c)" to the same pairs of equations as step 2 e.g N, NW

4. Then found the max of the first 4 directions,
   max (W,NW,N,NE) = 8 * max(((h + a) + max(g,b)), ((b+c) + max(a ,d))) - 3*z

5. Then found the max of the remaining 4 directions,
   max (E,SE,S,SW) = 8*max(((d + e) + max(f,c)), (f + g) + max(e,h))) - 3*z

6. Then combined equations from steps 4 and 5 to obtain
   max (ALL) = 8*(max (max(((h + a) + max(g,b)), ((b + c) + max(a,d))), max(((d + e) + max(f,c)), (f + g) + max(e,h)))) - 3*z

The final step would then be to check if max is greater than 383, and then return o_edge from the result

ClockPeriod = #Components_maxClockPeriod * TotalDelay

ClockPeriod = Max * TotalDelay + Add * TotalDelay + Flop * TotalDelay

ClockPeriod = 1 * 2.55 + 1 * 2.3 + 1 * 0.86

ClockPeriod = 5.71 ns

ClockSpeed = 1/ClockPeriod = 175.13 MHz

To calculate clock speed, we assumed 9 bits and 2 muxes to get the upper bound.

Area = #ADD * #LUTs + #MAX * #LUTs + #SUB * #LUTs + #COMP * #LUTs + #SHIFT * #LUTs + #REG * #LUTs

Area = 4 * 18 + 3 * 22 + 1 * 18 + 3 * 22 + 1 * 0 + 7 * 1

Area = 229

Area ~ 250

To calculate the area, we assumed 9 bits and 2 muxes to get the upper bound. The area was then rounded up to 250 to account for reading and writing

Finally, using the values from above, the optimality of the system can be determined. The functionality is set to 1000 as we are assuming full functionality

Optimality = Functionality * ClockSpeed / Area

Optimality = 1000 * 175.13 / 250

Optimaility = 700.52

Optimality ~ 70%