# Data Flow Diagram

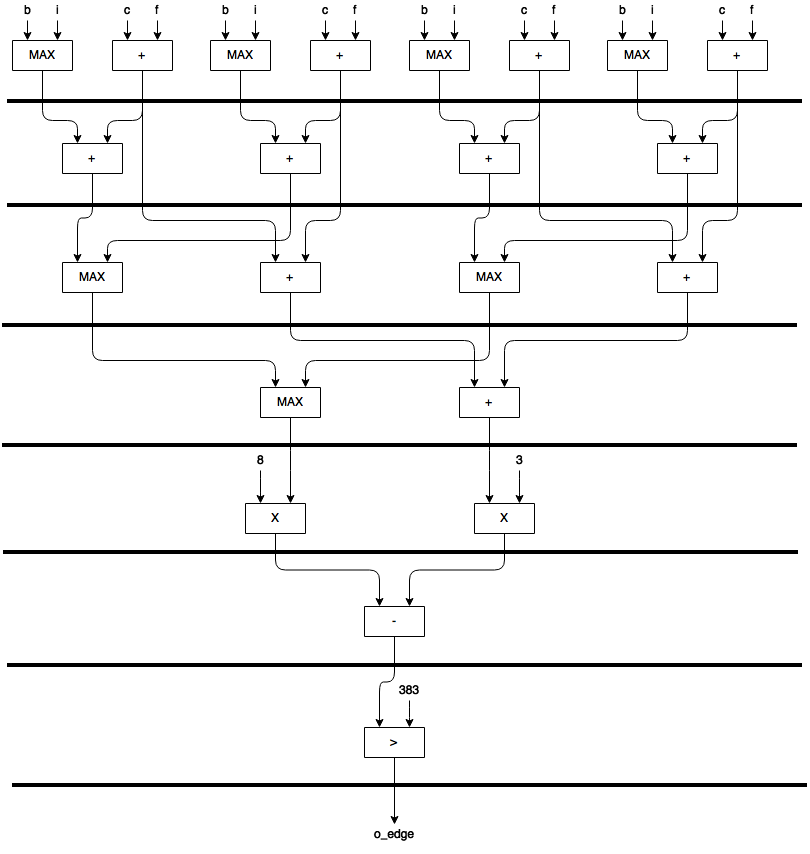
Group 7

### Anoj **Arulanantham**

### Ghanan Gowripalan

### Shayan Arman

### Sohum Rajguru



## Total Resources Used

Registers: 4 \* 8-bit register + 4 \* 9-bit register + 4\*10-bit registers + 4\*10-bit register+ 10-bit register, + 11-bit register + 2\*13-bit register + 13-bit register + 1-bit register (output)

Input: 8

Output: 1

8-bit maxer: 4

9-bit maxers: 3

9-bit adders: 4

10-bit adders: 4

11-bit adder: 1

Multiplier: 2

14-bit Subtractor: 1 (14 bit subtractor because the unsigned input to the subtractor will be 13 bits, however, since we are doing subtracting, the datatypes should be converted to signed, and when converting to signed from an unsigned datatype, we need to add an extra bit to make sure data isn’t changed)

13-bit Comparator: 1

## Latency

7 Clock Cycles

## Throughput

1/7

## Total Area Estimate

Registers: 68 1-bit registers (Maximum number of signals that cross a cycle boundary is at clock cycle 1, which is 4\*8-bit registers + 4\*9-bit registers = 69 1-bit registers)

Output: 1

Input: 8

Adders: 4

Maxers: 4

Multipliers: 2

Subtractors: 1

Comparator: 1

## Clock Period

Flop + MAX (Maxer, 11-bit Adder, Multiplier, 14-bit Subtractor, 13-bit Compator)

=Flop + 5.13ns (From the 13-bit comparator)

## Clock Speed

1/Clock Period

## Optimality

Optimality = Functionality \* Clock Speed/LE Count

## How the Calculations Were Done

We began with listing down the eight equations for each direction, and grouped them in pairs – NE and E, SE and E, SW, and W, and NW, and N. Let’s work with NE and E for example:

NE = 5 \* (b + c + f) – 3 \* (a + d + g + h + i)

E = 5 \* (c + f + i) – 3 \* (a + b + d + g + h)

We simplified these equations using the identity 5A-3B = 8A – 3(A+B).

This simplifies our equations to:

NE = 8 \*(b + c + f) – 3 \* ((b + c + f) + (a + d + g + h + i))

E = 8 \* (c + f + i) – 3 \* ((c + f + i) + (a + b + d + g + h))

We observed that the right hand side of the calculation (ie. 3 \* ((b + c + f) + (a + d + g + h + i))) stays the same for each of the eight directions, so we would not need these to determine our *EdgeMax.*

To determine which was the larger direction in each pair, we observed that in the left hand side of the calculation, (c+f) remained the same; so, we only really need the value of b in NE, or i in E to determine which is the larger direction of the two. Thus, simply max(b, i) gives us the larger of the two equations. We repeat this process for the remaining three pairs of equations.

We’ve thus narrowed our problem down to four directions (ie. the larger of each pair), and need to find the largest of these; so, we add the two common terms that each direction shared with its paired partner (c+f for the NE and E group), and have to calculate the maximum of these sums to find our *EdgeMax.*

NE and E: (Max(b, i) + (c + f))

SE and S:(Max(f, g) + (i + h))

SW and W: (Max(a, h) + (g + d))

NW and W: (Max(c, d) + (a + b))

Thus, our final equation for the value of *EdgeMax* is: 8\*Max((Max(b, i) + (c + f)), (Max(f, g) + (i + h)), (Max(a, h) + (g + d)), (Max(c, d) + (a + b))) – 3\*(a+b+c+d+f+g+h+i).