# Data Flow Diagram

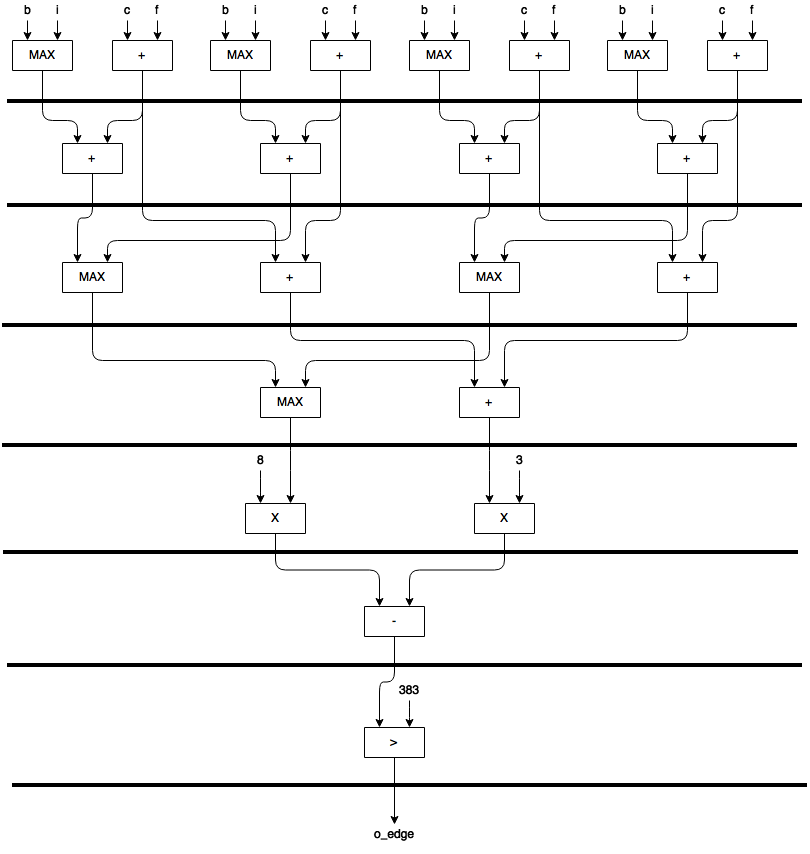
Group 7

### Anoj **Arulanantham**

### Ghanan Gowripalan

### Shayan Arman

### Sohum Rajguru



## Total Resources Used

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

Inputs: 8

Outputs: 1

8-bit Maxers: 4

9-bit Maxers: 3

9-bit Adders: 4

10-bit Adders: 4

11-bit Adder: 1

Multipliers: 2

14-bit Subtractor: 1 (the unsigned input to the Subtractor will be 13 bits – to convert this to a signed datatype, we need to account for an extra bit)

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: 4\*8-bit registers + 4\*9-bit registers = 68 1-bit registers)

Output: 1

Input: 8

Adders: 4

Maxers: 4

Multipliers: 2

Subtractors: 1

Comparator: 1

## Clock Period

Flop + Maximum (Maxer, 11-bit Adder, Multiplier, 14-bit Subtractor, 13-bit Comparator)

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

## Optimality

**LE Count**: (not accounting for Maxers and Multipliers)

Cycle1: (9-bit adder + register LUT) \* 4 + (8-bit register) \* 4 = 68 LEs

Cycle 2: 4 \* (10-bit adder + register) = 40 LEs

Cycle 3: 2 \* (10-bit adder + register) + 2 \* (10-bit register) = 40 LEs

Cycle 4: (11-bit adder + register) = 6 LEs

Cycle 6: 14-bit Subtractor = 14 LEs

Cycle 7: 13-bit Comparator = 6 LEs

Therefore, total LE = 68 + 40 + 40 + 6 + 14 + 6 = 174

**Clock Speed** = 1/Clock Period = 194.9 MHz

**Functionality**: We will assume our system is perfect, with a functionality score of 1000.

Therefore,

**Optimality** = Functionality \* Clock Speed/LE Count

= 1000 \* (194.9MHz/174)

= 1,120.1

## How the Calculations Were Done

|  |  |  |
| --- | --- | --- |
| A | B | C |
| D | E | F |
| G | H | I |

3x3 table

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).