# Address calculator

## Abstract

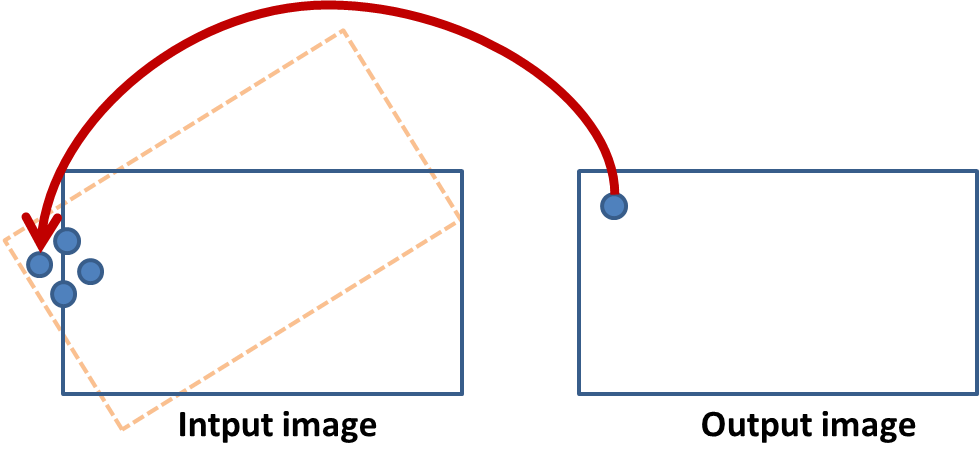
The unit calculates the source pixels address in matrix form a given current position (address, matrix form) in the output image.

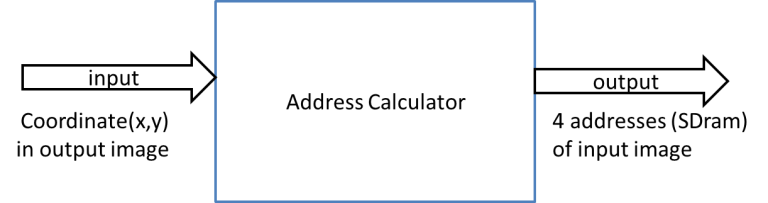
The output is addresses of 4 pixels that are required for the bilinear interpolation.

The input is address of 1 pixel of current position in output image.

**Note: X represents row indexes, Y represents column indexes.**

# Illustration

****



## Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **description** | **type** | **size** | **Recived from** |
| Zoom factor | Holds the zoom factor | signed | 8 bits | Param\_reg |
| Sin\_teta | Holds sin(teta) | signed | 8 bits | Param\_reg |
| Cos\_teta | Holds cos(teta) | signed | 8 bits | Param\_reg |
| Row\_idx\_in | Holds the current row index of the output image | signed | 11 bit | Param\_reg |
| col\_idx\_in | Holds the current column index of the output image | signed | 11 bit | Param\_reg |
| X\_crop\_start | Holds the row index of the top left pixel for crop | signed | 11 bit | Param\_reg |
| Y\_crop\_start | Holds the column index of the top left pixel for crop | signed | 11 bit | Param\_reg |
| X\_size\_in | Holds the number of rows in the input image | generic | 10 bit | Img\_man\_top (Mds\_top\_block) |
| Y\_size\_in | Holds the number of columns in the input image | generic | 10 bit | Img\_man\_top (Mds\_top\_block) |
| X\_size\_out | Holds the number of rows in the output image | generic | 10 bit | Img\_man\_top (Mds\_top\_block) |
| Y\_size\_out | Holds the number of columns in the output image | generic | 10 bit | Img\_man\_top (Mds\_top\_block) |

## Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **description** | **type** | **size** | **Destination** |
| TL\_out | Holds the top left row index in input image.  Index in SDRAM mode | Std\_logic\_vector | 23 | TBD |
| TR\_out | Holds the top right row index in input image. Index in SDRAM mode | Std\_logic\_vector | 23 | TBD |
| BL\_out | Holds the bottom left row index in input image. Index in SDRAM mode | Std\_logic\_vector | 23 | TBD |
| BR\_out | Holds the bottom right row index in input image. Index in SDRAM mode | Std\_logic\_vector | 23 | TBD |
| Data\_valid | Indicates ready data in outputs | Std\_logic\_vector | 1 | TBD |
|  |  |  |  |  |
|  |  |  |  |  |

Outputs coordinates are 23 bits because 22 bits is the SDRAM size + 1 bit for multiply calculations need.

# The algorithm

Basically, the algorithm calculates the desired coordinates from the origin image, from given [x,y] coordinates. The mathematical equation the algorithm uses is:

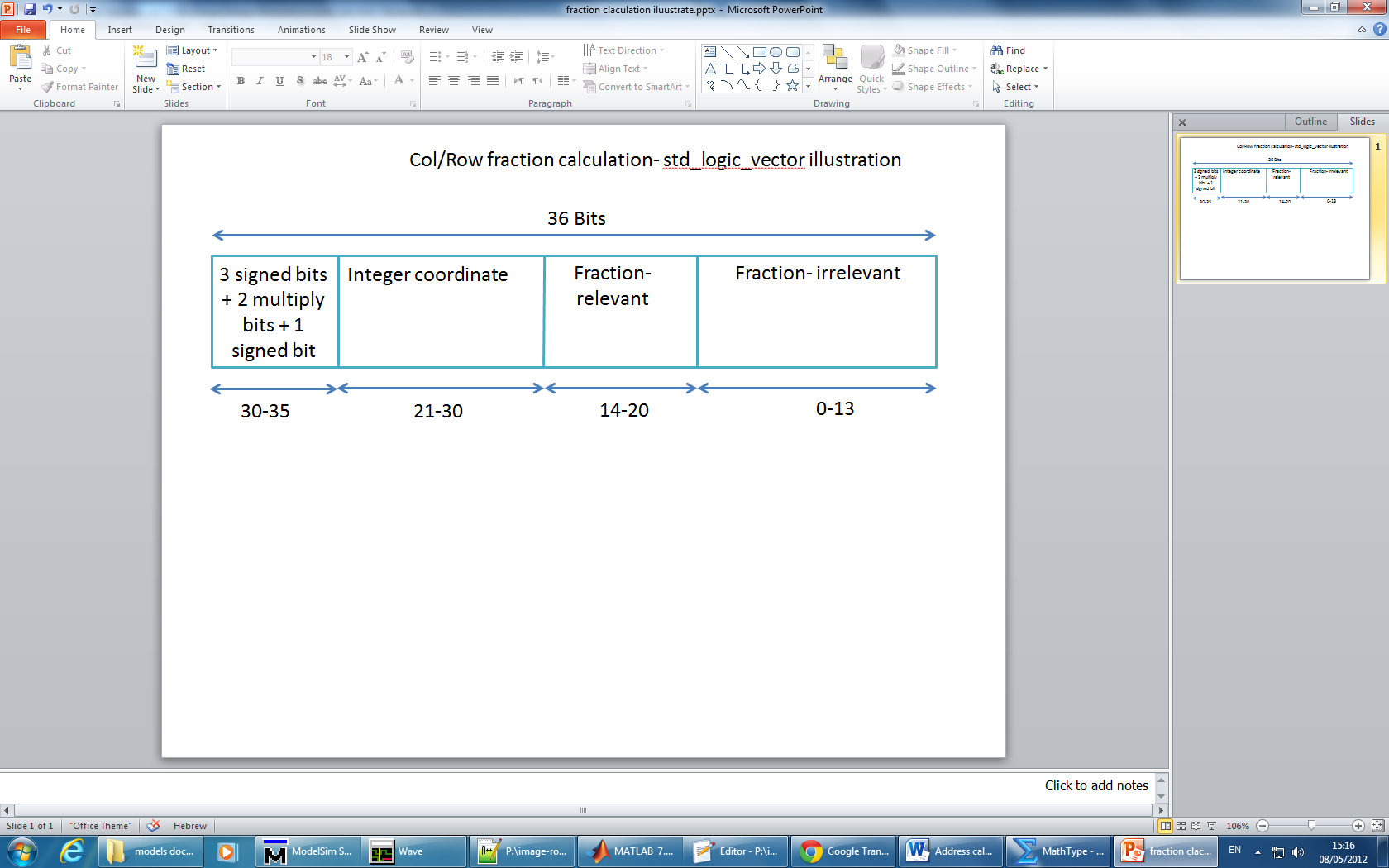


For horizontal calculation the algorithm uses the same equation after replacing the relevant parameters.

In order to reduce calculation time and improve throughput, the algorithm disassembles the equation into 3 parts (Pipeline).

Every multiply operation that should be made, 1 bit must be added to prevent overflow effects. In addition, the new size must be the size of the two arguments together. Finally, 1 bit of signed type is added (mid calculations might include negative results). Thus, the sizes of output ports were determined.

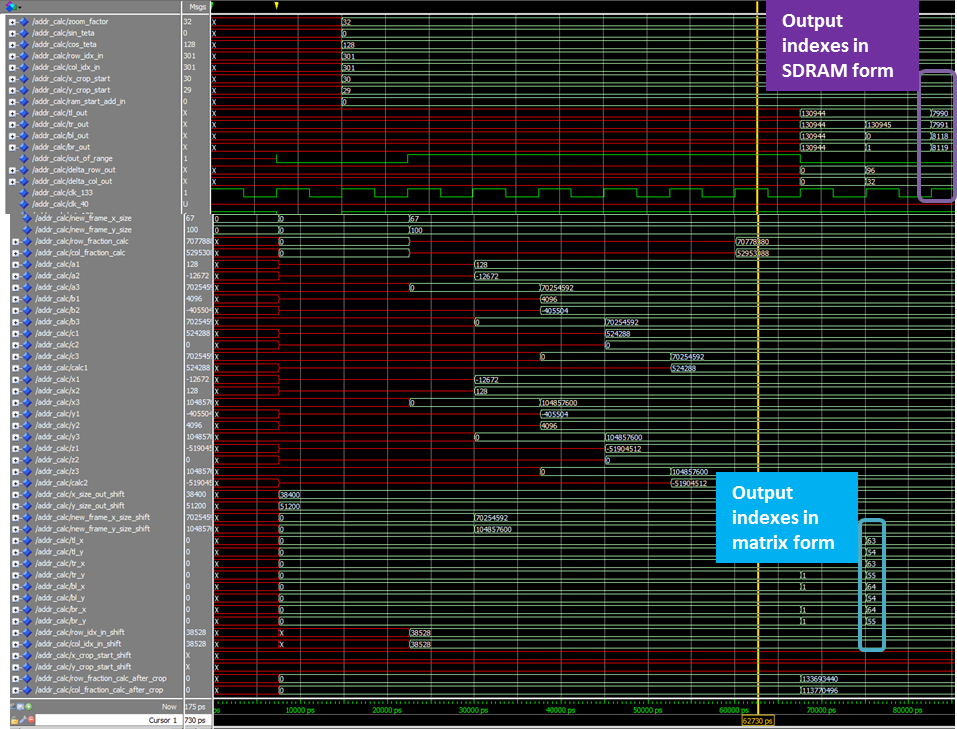
The following image may explain the final output form:



This std\_logic\_vector represent the desired output index in matrix form.

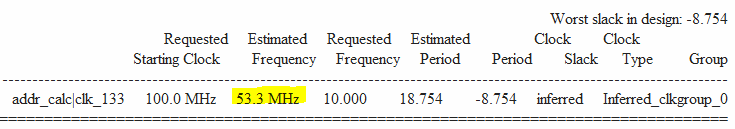
In order to convert this index to SDRAM form, the algorithm uses another simple equation and creates the output index port (23 bits).

# Simulation results



# Code improvement

The synthesis before improvement:



The diagram: