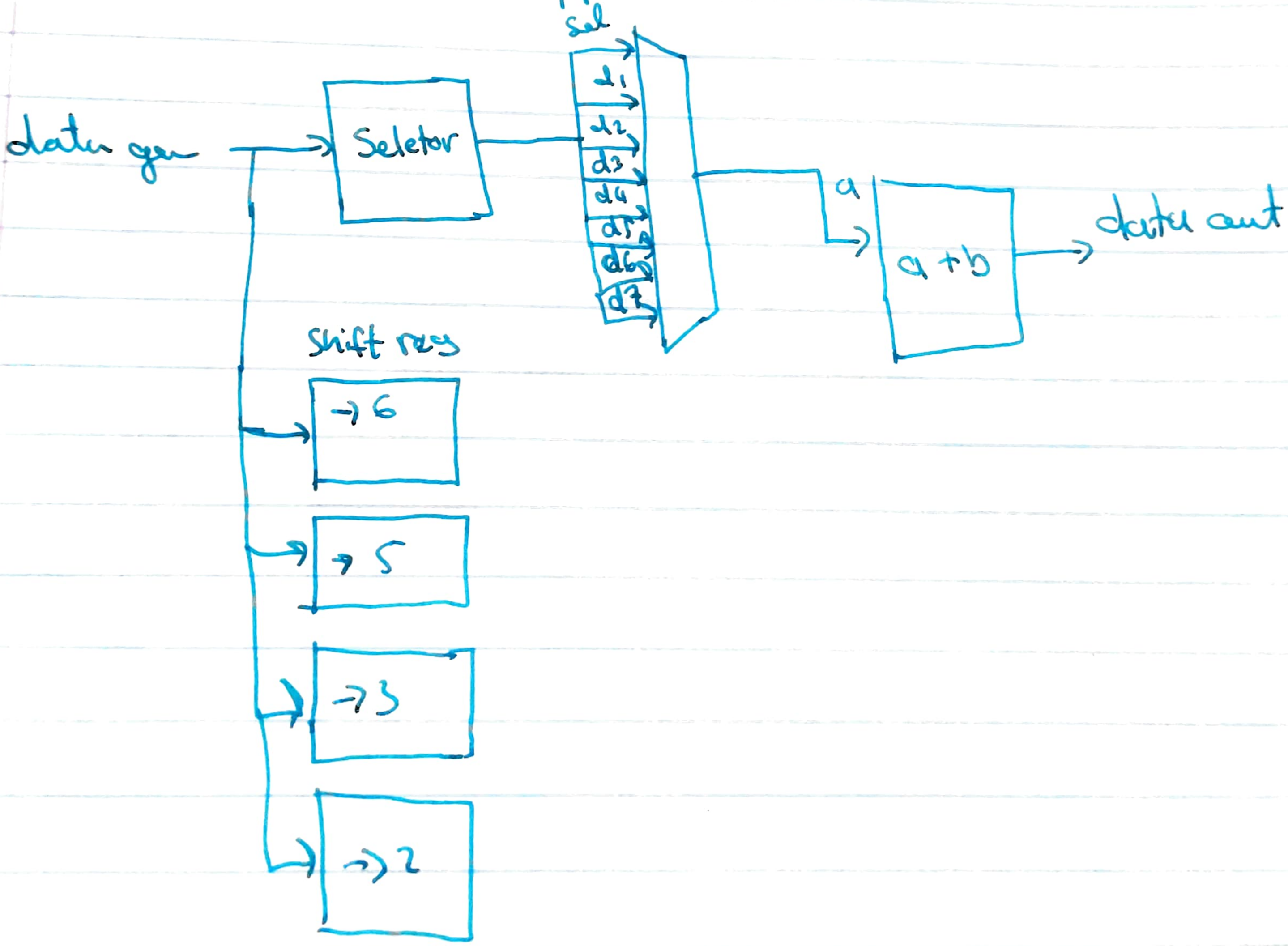
21/09/20

Aim :

The aim of the proc is to verify high level python model against a low level python and low level VHDL signal data is transmitted over serial to verify in python.

Proposed Model in Papers.



21/09/20

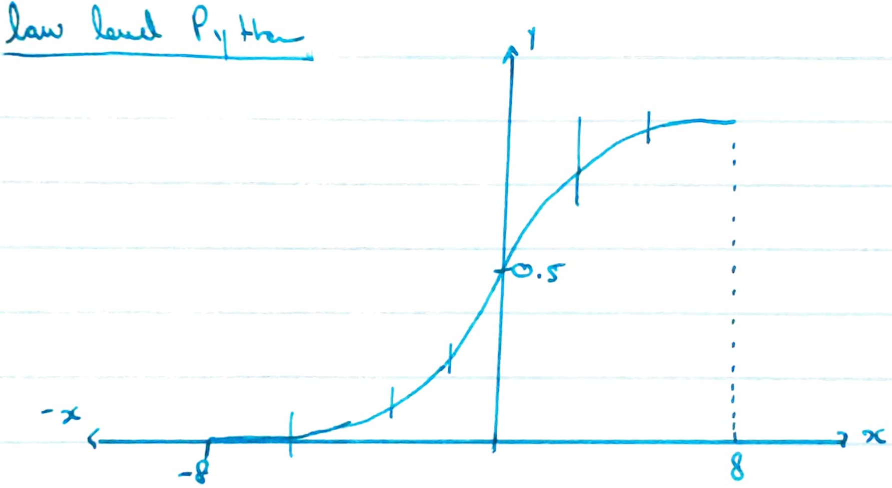
Method of gen High level

the high leve modeled is generate using a ~~m~~ python modes where the functi of the sigmoid can be expressed as

$$f(x) = \frac{1}{1+e^{-x}}$$

the function is bound from -8 to 8 on the x-axis and 0 to 1 on the y-axis

low level Python



the sigmoid can be split into 7 region and the gradient of each section is calculated the region are

~~$-8 \to -4$: 1~~ 1) $-8 < x < -4$

~~$-2 \to -1$: 2~~ 2) $-4 < x < -2$

~~$-1 \to 1$: 3~~ 3) $-2 < x < -1$

~~$+1 \to 2$: 4~~ 4) $-1 < x < 1$

~~$2 \to 4$: 5~~ 5) $1 < x < 2$

$4 \to 8$: 6 6) $2 < x < 4$

7) $4 < x < 8$

25/09/20

~~the gr~~ Calculating Gradients

	Gradients
1) $-8 < x < -4$	0.01563
2) $-4 < x < -2$	0.02875
3) $-2 < x < -1$	0.13
4) $-1 < x < 1$	0.25
5) $1 < x < 2$	~~[illegible]~~ 0.12
6) $2 < x < 4$	0.0335
7) $4 < x < 8$	0.01575

The gradient will be difficult to employ in VHDL due to the value being floats this can be circumvented by using shift register.

0.01563 → 2^{-6} shift right by 6

0.02875 → 2^{-5} by → 6

0.13 → 2^{-4}

0.25 → 2^{-2}

0.12 → 2^{-3}

0.0335 → 2^{-4}

0.01575 → 2^{-6}

25/04/20

Working with floating point data

The two common ways of working with decimal point data is either to convert to a very large value as employed in my method or make them very small.

top asymptote at +1 was converted to

$1 \times 2^{16} = 65536$

$-8 \times 2^{16} = -524288$

$8 \times 2^{16} = 524288$

In doing this all the data is convert to large integers.

Pseudo code for low level python

~~set gradient $m_1 \rightarrow m_2$~~

~~Create a y list and a x list~~

~~y= [ ] - 0 → 1048576 (2x 5242288)~~

~~x= [ ] - " "~~

~~Populat Piece wise function~~

~~$-8 < x < -4$~~

~~$y[i] = 1/64\, x[i] + 8192$~~

~~$-4 <$~~

26/09/20

Psuedo code for low level python

create the ~~empt~~ empty lists

```
x[ ] → 0 → 1048576 $(8 \times 2^{16} \times 2)$
Y[ ] → "              "

step for loop (0, 1048576)
    $-8 < x < -4$ :
  y[i] = $\frac{1}{64}$x[i] + 8192
  $-4 < x < -2$ :
  y[i] = $\frac{1}{32}$x[i] + 11468
    $-2 < x < -1$
  y[i] = $\frac{1}{8}$x[i] + 24576
    $-1 < x < 1$
  y[i] = $\frac{1}{4}$x[i] + 32440
   $1 < x < 2$
  y[i] = $\frac{1}{8}$x[i] + 40960
   $2 < x < 4$
  y[i] = $\frac{1}{32}$x[1] + 52101
   $4 < x < 8$
  y[i] = $\frac{1}{64}$x[i] + 56688
```

29/09/20

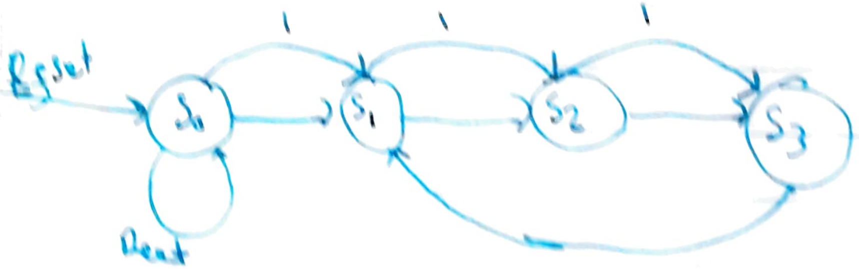
VHDL low level model.

S0 → Startup Initialise

S1 → ~~Stan~~ Start counting up from -524288
total steps ~~1048826~~ 1048576

S2 → Perfom ~~Prec~~ Piece wise function

S3 → Save to textfile.



24/08/20

Improved model.

The 7 Step were increse to 14 piecewise funtion will improve error as demonstrated in the video.