University of Moratuwa

Circuits And System Design

EN3030

ASSIGNMENT : WEEK 01

Team : INFOS

Yasarathna D.D.K.B.	190719V	Ex - 01
Tharindu.O.K.D	190622R	Ex - 02
S.Sanjith	190562G	Ex - 03
K.Kajhanan	190286M	Ex - 04

August 6, 2022

Contents

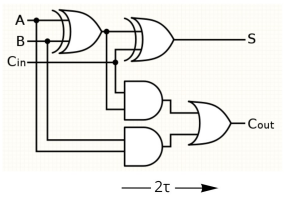
Executable codes can be found here

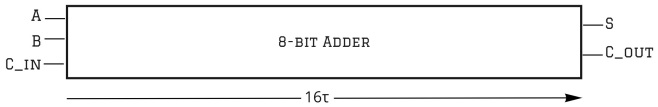
1 Ripple Carry Adder & Pipelined Architecture

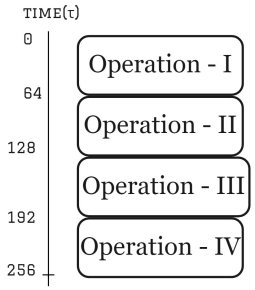
The following discussion will illustrate the improvement in the performance of the adder in adding four 32-bit numbers consecutively. Adders are formed using four separate eight-bit adders and results are based on giving inputs in two different passions in a way to evaluate the effect of pipelining over performance. For the comparison purposes the average gate delay is considered as τ.

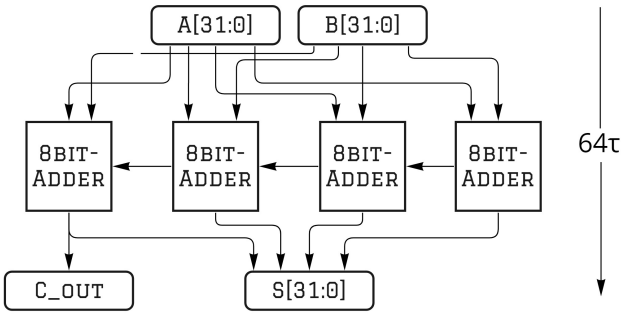
1.1 Operation in Normal Mode

A full adder takes at least 2τ from the time its inputs are available, to ensure useful results are in place. This way propagation of carry-value throughout the blocks leads to a delay of 16τ for an 8-bit adder. Hence an addition between two 32-bit numbers takes $4 \times 16\tau = 64\tau$.



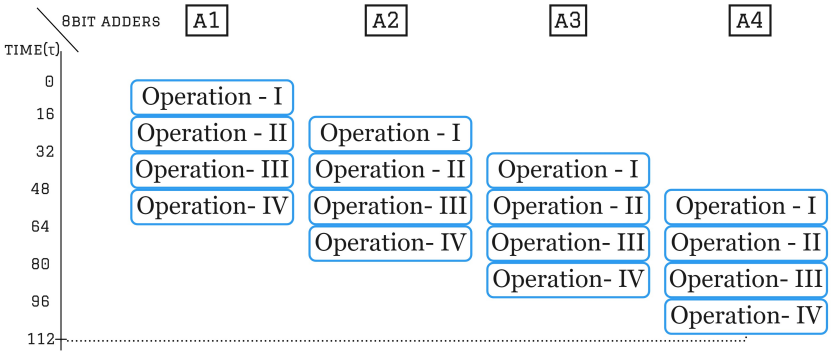






Total time taken by adder to perform four consecutive addition $= 4 \times 64\tau = 256\tau$

1.2 Pipelined Architecture



Total Time taken $= 112\tau$

The Main idea of pipelining is rather than waiting for the whole operation about one addition to complete, make use of the unutilized 8-bit adder blocks to perform the subsequent calculations.In the case of four additions, utilizing idle stages in the adder blocks seem to reduce the accumulated gate delays from 256τ low to 112τ.

Simulation Results

```
  0 A= 536849066, B=3221203626, S=3758052692
 64 A= 142352352, B= 465693463, S= 608045815
128 A= 847583573, B= 347583488, S=1195167061
192 A= 573286762, B= 235762376, S= 809049138
```

Simulation - I : Normal Architecture

```
  0 A=         x, B=         x, S=         x
 16 A= 124532445, B= 212421455, S=         x
 32 A= 142352352, B= 465693463, S=         x
 48 A= 847583573, B= 347583488, S=         X
 64 A= 573286762, B= 235762376, S= 608045868
 80 A= 573286762, B= 235762376, S=1195167223
 96 A= 573286762, B= 235762376, S= 809049173
112 A= 573286762, B= 235762376, S= 809049138
```

Simulation - II : Pipelined Architecture

2 Carry Look-Ahead Adder

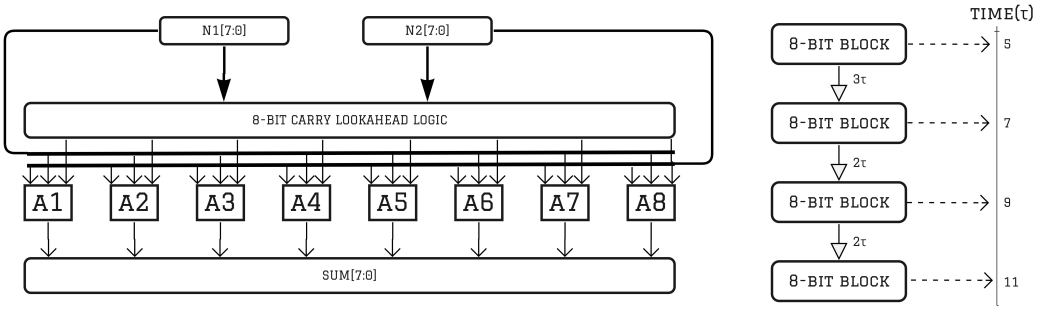
Instead of waiting for carry-bit to propagate through separate full adders, CLA implementation generates carry-bit parallelly with the price of a larger fan-in.

$$\text{Carry-Generate} \quad P_i = A_i \oplus B_i$$

$$\text{Carry-Propagate} \quad G_i = A_i \cdot B_i$$

$$\text{Carry-Bit} \quad C_n = C_0 \prod_{i=1}^{n-1} + \sum_{i=0}^{n} -1 \left\{ G_i \prod_{j=i+1}^{n-1} P_j \right\}$$

In the case of an 8-bit adder, all the carry bits can be generated in an average time of 3τ. Together with a delay of 2τ in the full adder, an 8-bit adder block takes 5τ to complete its process. Each block takes 2τ time to generate the carry-bit necessary for the next block and additionally the first stage takes τ time more to generate carry generate and propagate terms. Times add up to give a total delay of 11τ in the completion of addition between two 32-bit numbers.



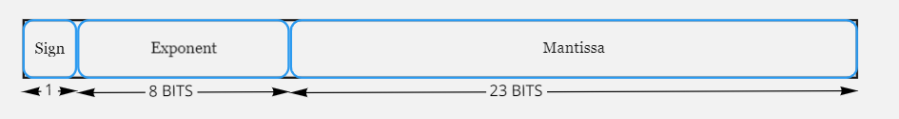
Total time taken for the addition of four numbers $= 4 \times 11\tau = 44\tau$

```
 0 A= 124532445, B= 212421455, S= 336953900
11 A= 142352352, B= 465693463, S= 608045815
22 A= 847583573, B= 347583488, S=1195167061
33 A= 573286762, B= 235762376, S= 809049138
```

Simulation - III : CLA Architecture

3 Floating Point Addition/Subtraction

3.1 IEEE-754 Single Precision Format



For the verification purposes h-schmidt, an online floatconverter is used.

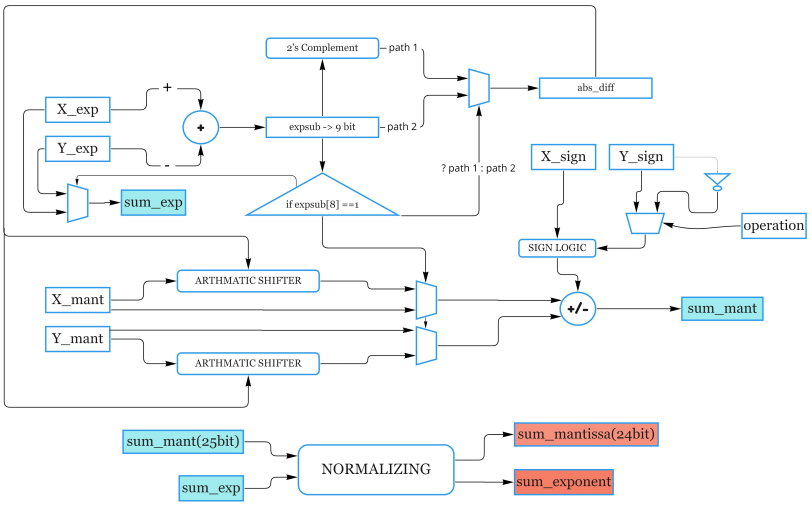
3.2 Algorithm

1. Compare the exponents of two numbers and find the absolute difference.
2. Shift the number with the smaller exponent to right by the amount of absolute difference to make the exponents of both numbers equal. Assign the prominent exponent as the exponent of the sum.
3. Check the signs of two numbers after performing sign conversion according to operation

- If both are of the same sign add two corresponding mantissa and assign it to sum_mantissa.
- Otherwise, subtract the matissa of number with 1 as sign bit from the other and assign the magnitude of it as sum_mantissa.

At this stage, sum mantissa will contain 25 bits to accomadate the overflow.

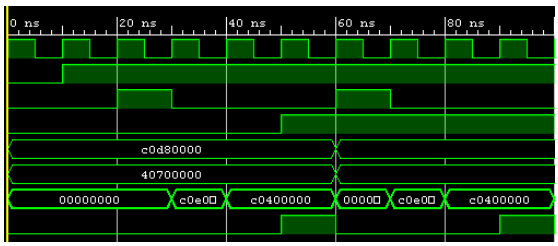
4. Assign the sum_sign according to sign of two numbers and above derived answer.
5. Shift the mantissa left until and 24th-bit position of the mantissa is one and reduce the exponent of the sum accordingly(normalization)
6. Now discard the implicit 1 and consider only the lower 23-bits of sum_mantissa.
7. Final Answer = {sum_sign, sum_exponent, sum_mantissa}



Design Flow of FP Add/Sub

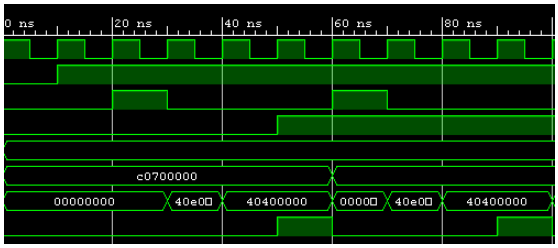
3.3 Results

Simulation Results



-6.75+3.75 = -3 & -6.75-(-3.75) = -3

```
 0 X=3235381248, Y=1081081856, sum=          0, Valid =0
30 X=3235381248, Y=1081081856, sum=3235905536, Valid =0
40 X=3235381248, Y=1081081856, sum=3225419776, Valid =0
50 X=3235381248, Y=1081081856, sum=3225419776, Valid =1
60 X=1081081856, Y=1087897600, sum=          0, Valid =0
70 X=1081081856, Y=1087897600, sum=3235905536, Valid =0
80 X=1081081856, Y=1087897600, sum=3225419776, Valid =0
90 X=1081081856, Y=1087897600, sum=3225419776, Valid =1
```



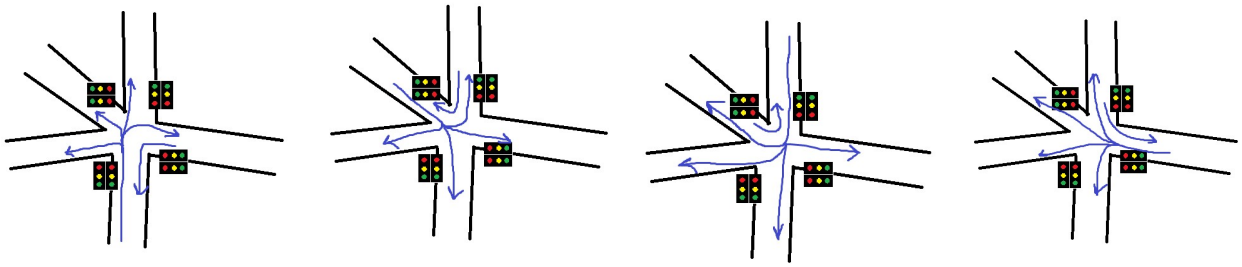
6.75-3.75 = 3 & 6.75+(-3.75) = 3

```
 0 X=1087897600, Y=3228565504, sum=          0, Valid =0
30 X=1087897600, Y=3228565504, sum=1088421888, Valid =0
40 X=1087897600, Y=3228565504, sum=1077936128, Valid =0
50 X=1087897600, Y=3228565504, sum=1077936128, Valid =1
60 X=1087897600, Y=1081081856, sum=          0, Valid =0
70 X=1087897600, Y=1081081856, sum=1088421888, Valid =0
80 X=1087897600, Y=1081081856, sum=1077936128, Valid =0
90 X=1087897600, Y=1081081856, sum=1077936128, Valid =1
```

Addition and Subtraction indicated above are choosen in a way to return the same result. During the true value of valid entry simulation returns same result for each operation.

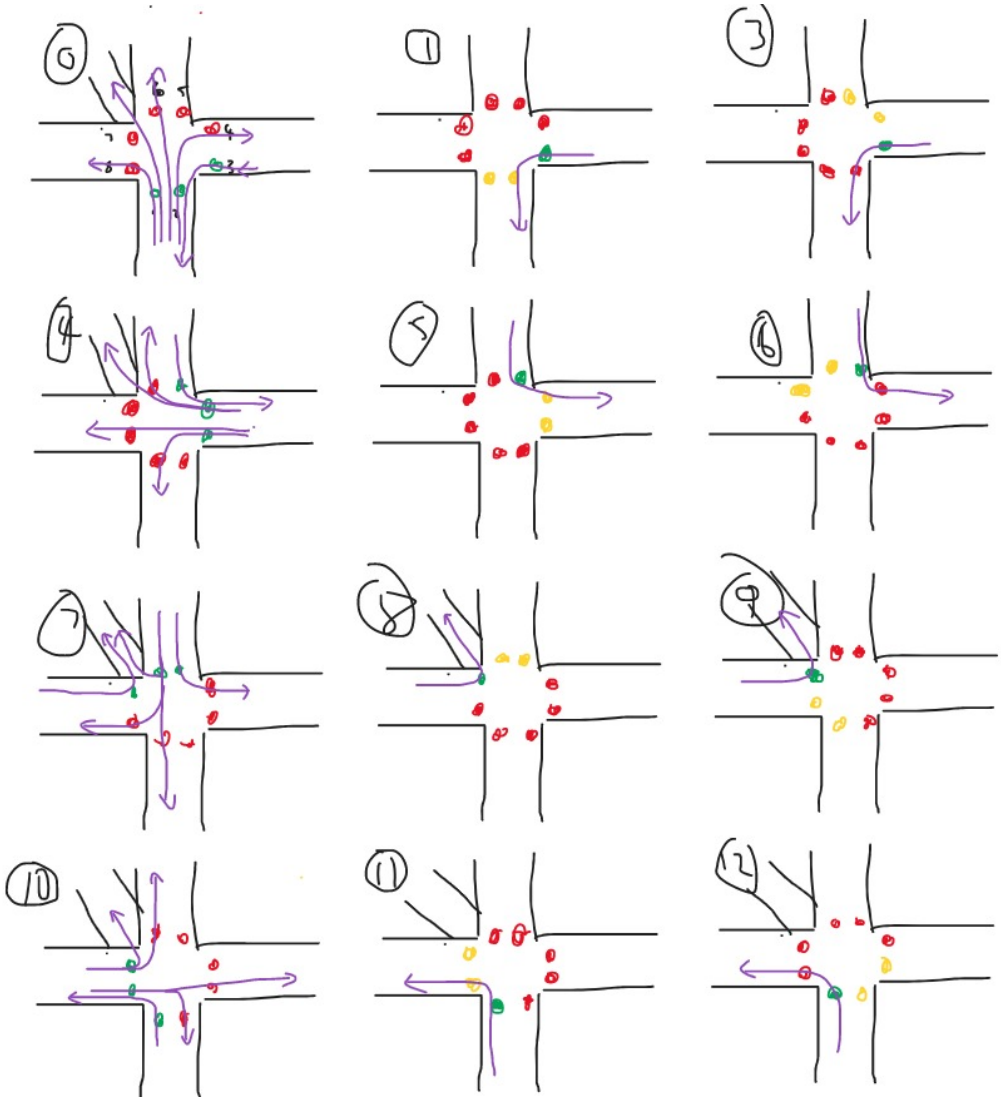
4 Traffic Light Controller System : Borella Junction

The situation can be realized as four main states as shown in the figure below.



state : 1 state : 2 state : 3 state : 4

Transition between states properly done with no intersecting pathways having green at the same time. With these there are 12 states in total. These states can be realized using four flip-flops and required combinational circuitories. State Diagram illustrating all the 12 states is as follows.



State Diagram

Pedestrain crossing is not considered as the Borella Junction has an underground crossing. A 10 bit counter is used, and throughout each full count the trasition between states happen sequentiall, and then the procedure will iterate.

	Current				Next				Outputs																			
	A	B	C	D	A	B	C	D	R1	R2	R3	R4	R5	R6	R7	R8	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	G1	G2	G3	G4	G5	G6	G7	G8
0	0	0	0	0	0	0	0	1				1	1	1	1	1									1	1	1					
1	0	0	0	1	0	0	1	0				1	1	1	1	1	1	1									1					
2	0	0	1	0	0	0	1	1	1	1				1	1	1				1	1						1					
3	0	0	1	1	0	1	0	0	1	1				1	1	1											1	1	1			
4	0	1	0	0	0	1	0	1	1	1				1	1	1			1	1									1			
5	0	1	0	1	0	1	1	0	1	1	1	1				1						1	1						1			
6	0	1	1	0	0	1	1	1	1	1	1	1				1													1	1	1	
7	0	1	1	1	1	0	0	0	1	1	1	1				1					1	1									1	
8	1	0	0	0	1	0	0	1		1	1	1	1	1			1							1							1	
9	1	0	0	1	1	0	1	0		1	1	1	1	1											1						1	1
10	1	0	1	0	1	0	1	1		1	1	1	1	1									1	1	1							
11	1	0	1	1	0	0	0	0				1	1	1	1	1		1	1						1							
12	1	1	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
13	1	1	0	1	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
14	1	1	1	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
15	1	1	1	1	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Excitation Table of Different States

Due to the larger size of variables, simulation results are not included with the report.