

Nama : Yasir Abdul Khalim

NIM : 2102845

Kelas : TE-01 2021

UAS Desain Sistem Digital

Buatlah Mikroarsitektur menggunakan verilog dari

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$

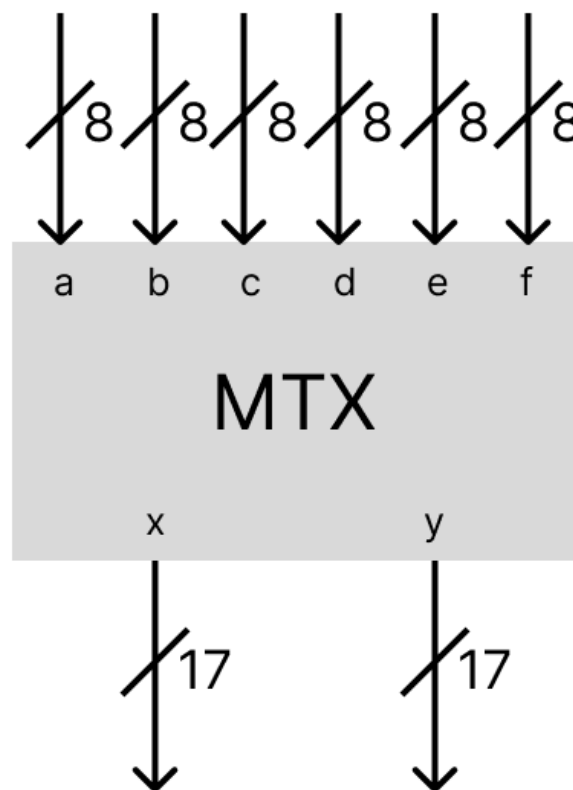
Dengan panjang data a, b, c, d, e, dan f, adalah 8 bit

Secara matematis penyelesaian matix tersebut adalah sebagai berikut.

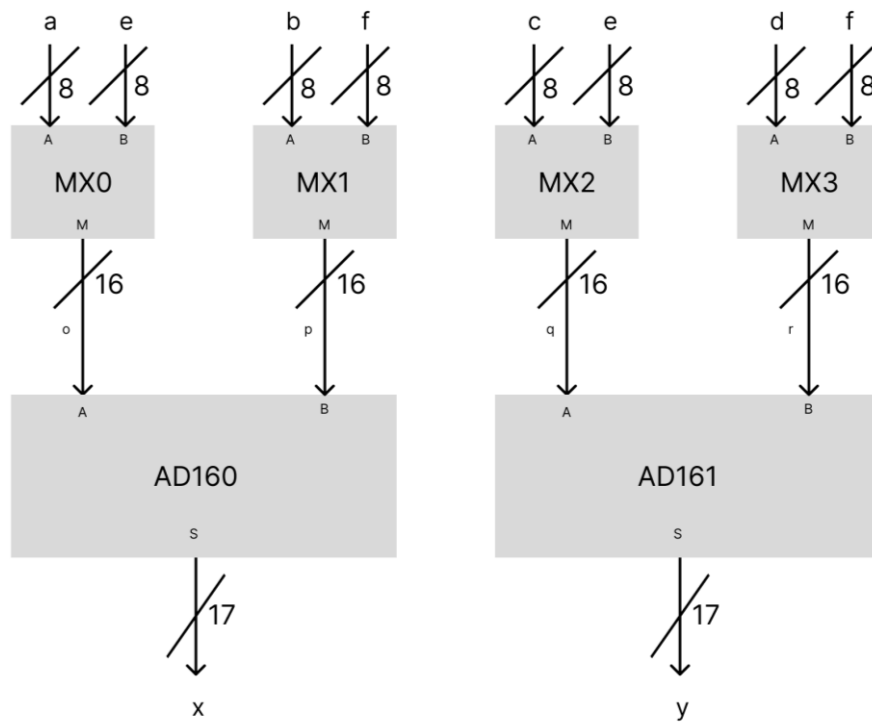
$$\begin{aligned} x &= ae + bf \\ y &= ce + df \end{aligned}$$

Maka untuk membuat arsitekturnya dalam verilog tanpa menggunakan “*” dan “+” adalah dengan menggunakan multiplier dan adder.

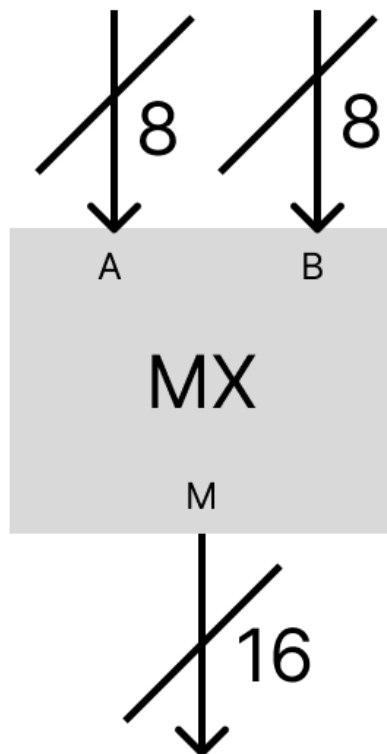
1. RTL



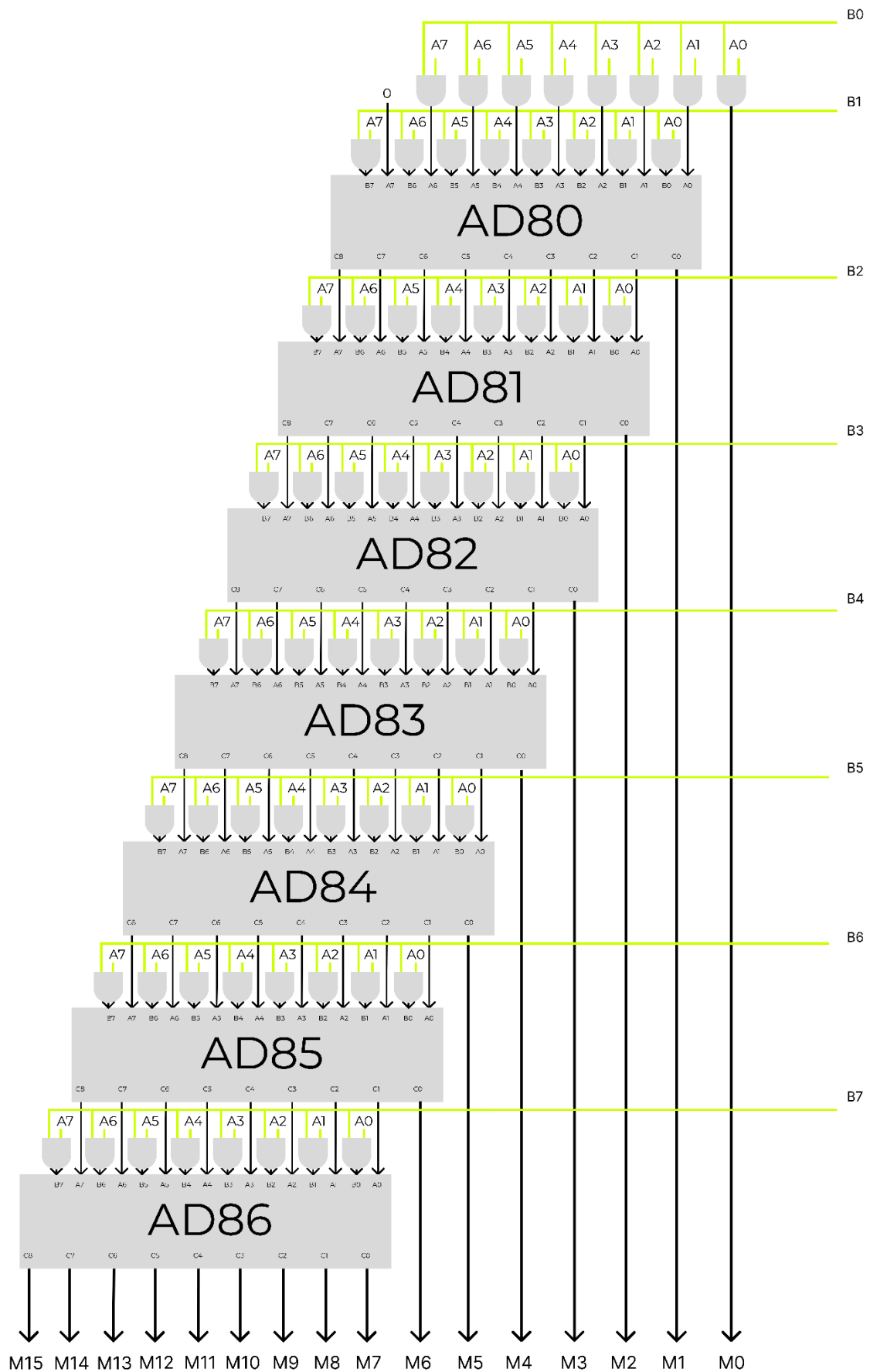
Gambar 1. Matrix



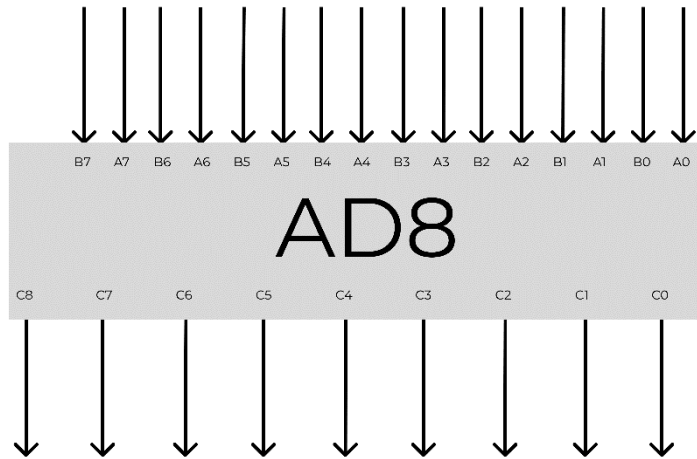
Gambar 2. Bagian dalam dari Matrix



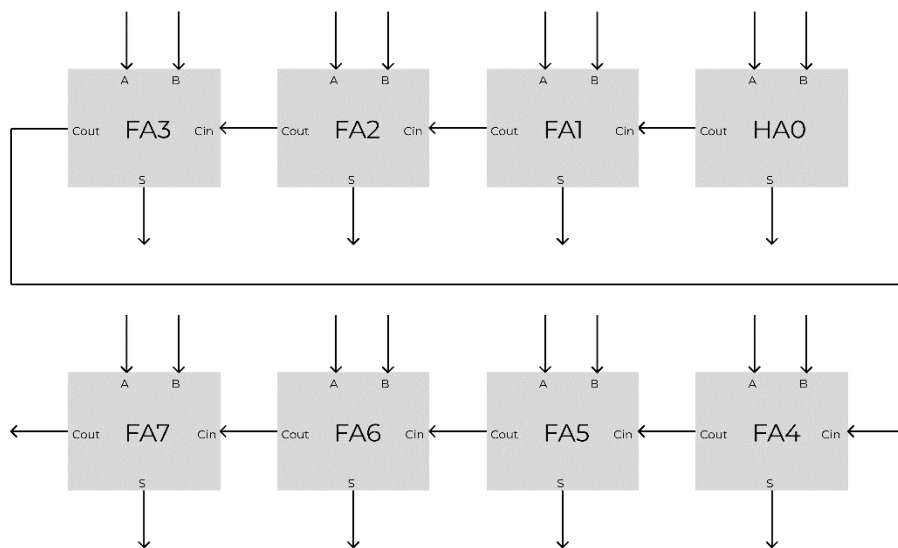
Gambar 3. Multiplier



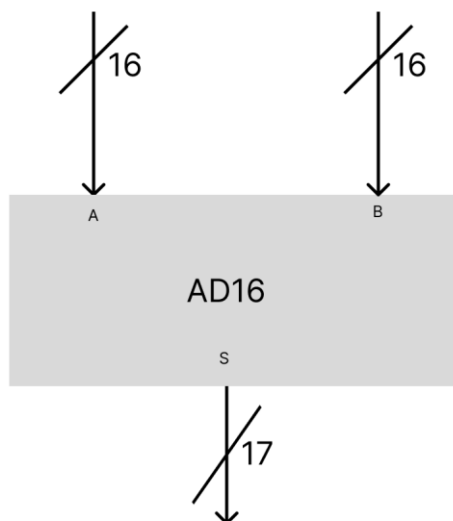
Gambar 4. Bagian dalam dari Multiplier



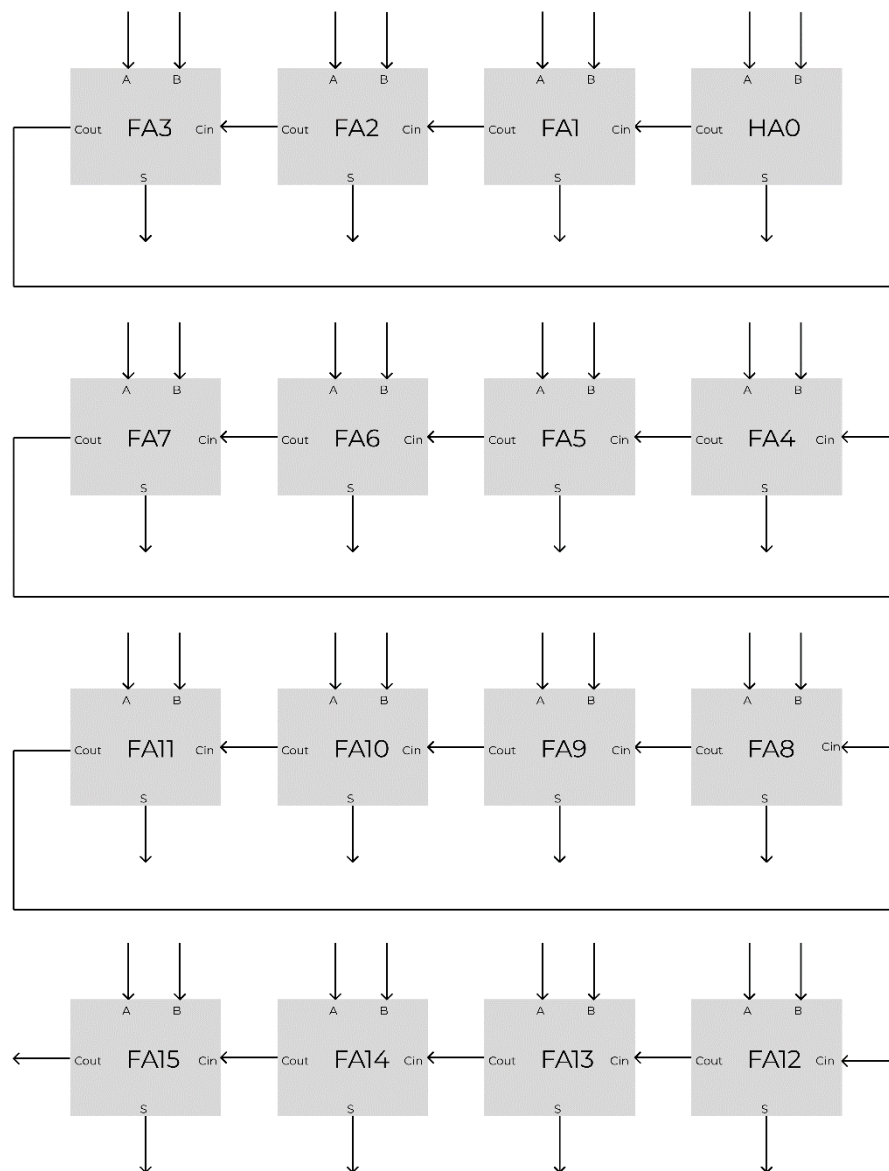
Gambar 5. Adder 8 bit



Gambar 6. Bagian dalam dari Adder 8 bit



Gambar 7. Adder 16 bit



Gambar 8. Bagian dalam dari Adder 16 bit

Penjelasan RTL:

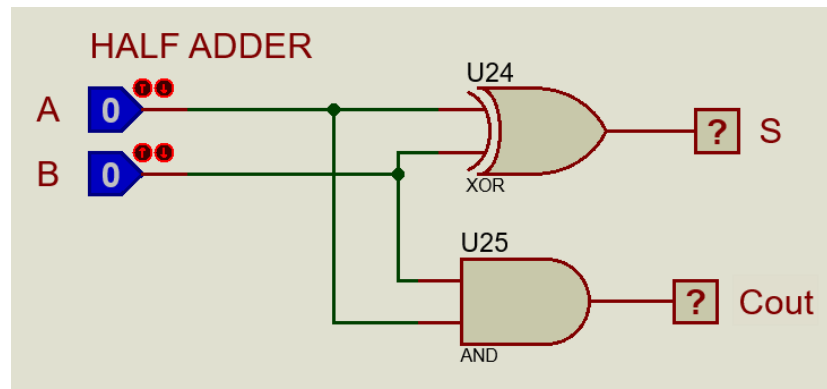
Register untuk melakukan operasi matrix dari soal yang diberikan diberi nama MTX dan di dalam MTX terdapat 4 modul multiplier atau yang dinamakan MX serta 2 module adder 16 bit atau yang diberi nama AD16. MTX memiliki 6 input yaitu a, b, c, d, e, dan f yang masing-masing inputnya memiliki panjang data 8bit dan 2 output yaitu x dan y yang memiliki panjang data 17 bit, panjang data ini diperoleh dari proses yang terjadi di dalam module matrix yaitu pada tahap pertama dilakukan penggunaan multiplier untuk mengoperasikan dua input yaitu a dikali e, b dikali f, c dikali e, dan d dikali dengan f. Karena menggunakan multipler dan panjang data masing-masing input adalah 8 bit maka output dari multiplier memiliki panjang data yaitu 16 bit, maka setelah operasi pertama didapatkan 4 hasil yaitu hasil dari ae, bf, ce, dan df kemudian hasil tersebut di inputkan pada adder 16 bit yaitu mengoperasikan ae ditambah bf dan ce ditambah df, sehingga menghasilkan 2 output dengan masing-masing panjang data sepanjang 17 bit. Untuk x merupakan hasil operasi dari ae dan bf dan y di merupakan

hasil operasi dari ce dan df. Di dalam MX (module multiplier) terdapat AND gate dan module dari adder 8 bit. Kemudian adder 8 bit tersusun oleh 1 half adder dan 7 full adder, begitu juga dengan adder 16 bit yang di dalamnya tersusun oleh half adder dan full adder namun dengan jumlah yang berbeda. Ader 16 bit tersusun dari 1 half adder dan 15 full adder. Sehingga dapat diperhatikan bahwa half adder dan full adder menjadi module dasar yang digunakan untuk membentuk module-module yang lainnya.

2. GTL

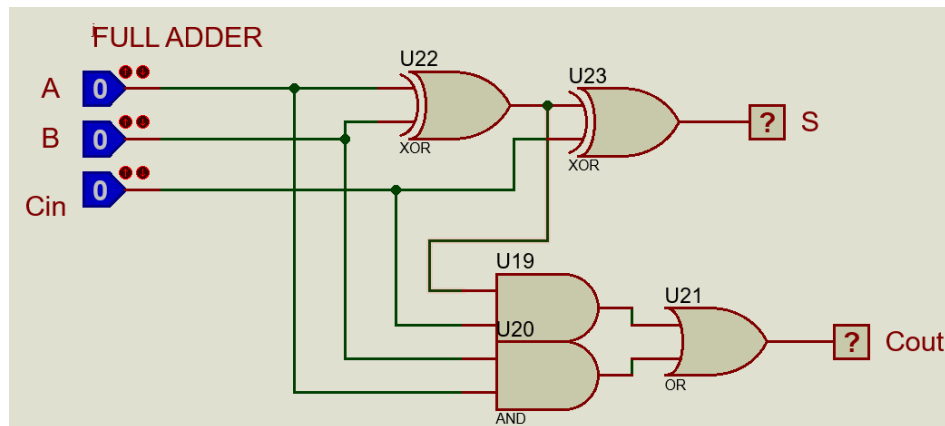
1) Half Adder

Input		Output	
A	B	Cout	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



2) Full Adder

Input			Output	
A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Penjelasan

a) Half Adder

Half adder memiliki 2 input yaitu A dan B serta 2 output yaitu S dan Cout. Half adder dibuat dengan menggunakan gerbang logika XOR untuk menghasilkan output S dan menggunakan gerbang logika AND untuk menghasilkan output Cout. Penggunaan gerbang logika ini berdasarkan truth table yang berlaku pada half adder sehingga diperoleh persamaan untuk mendapatkan output yaitu sebagai berikut.

[1] Persamaan untuk memperoleh S

$$S = \bar{A}B + A\bar{B}$$

$$S = A \oplus B$$

[2] Persamaan untuk memperoleh Cout

$$Cout = AB$$

Maka dengan persamaan tersebut maka dapat dibuat module half adder.

b) Full Adder

Full adder memiliki 3 input yaitu A, B dan Cin serta 2 output yaitu S dan Cout. Full adder dibuat dengan menggunakan gerbang logika XOR sebanyak 2 buah untuk menghasilkan output S serta menggunakan 2 gerbang logika AND dan sebuah gerbang logika OR untuk menghasilkan output Cout. Sama seperti half adder, gerbang logika yang digunakan dipilih berdasarkan persamaan yang didapat dari truth table yang berlaku pada full adder sehingga menghasilkan persamaan sebagai berikut.

[1] Persamaan untuk memperoleh S

$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

$$S = \bar{A}(\bar{B}C_{in} + B\bar{C}_{in}) + A(\bar{B}\bar{C}_{in} + BC_{in})$$

$$S = \bar{A}(B \oplus C_{in}) + A(\bar{B} \oplus \bar{C}_{in})$$

$$S = A \oplus B \oplus C_{in}$$

[2] Persamaan untuk memperoleh Cout

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB(\bar{C}_{in} + C_{in})$$

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB$$

$$C_{out} = AB + C_{in}(A\bar{B} + \bar{A}B)$$

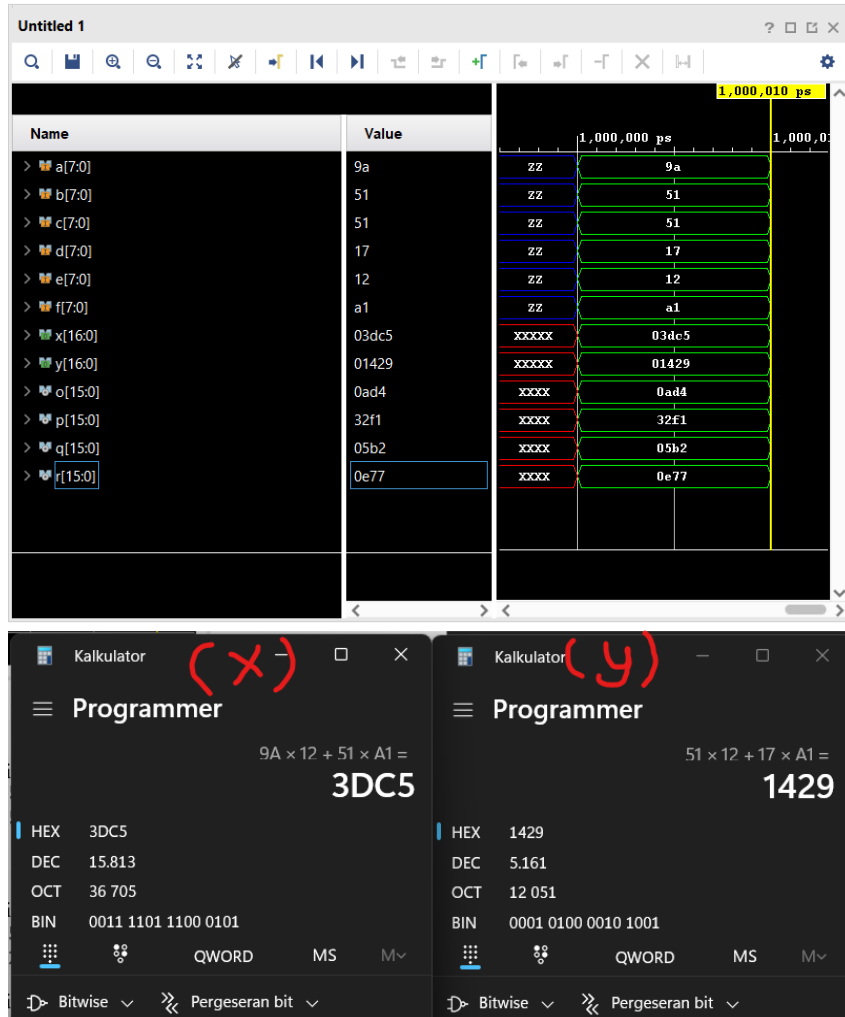
$$C_{out} = AB + (A \oplus B)C_{in}$$

3. Pembuktian

Pembuktian dengan cara memasukan nilai input dengan nilai-nilai yang telah ditentukan kemudian dibandingkan hasil dari menggunakan mikroarsitektur yang telah dibuat dengan hasil dari perhitungan menggunakan kalkulator.

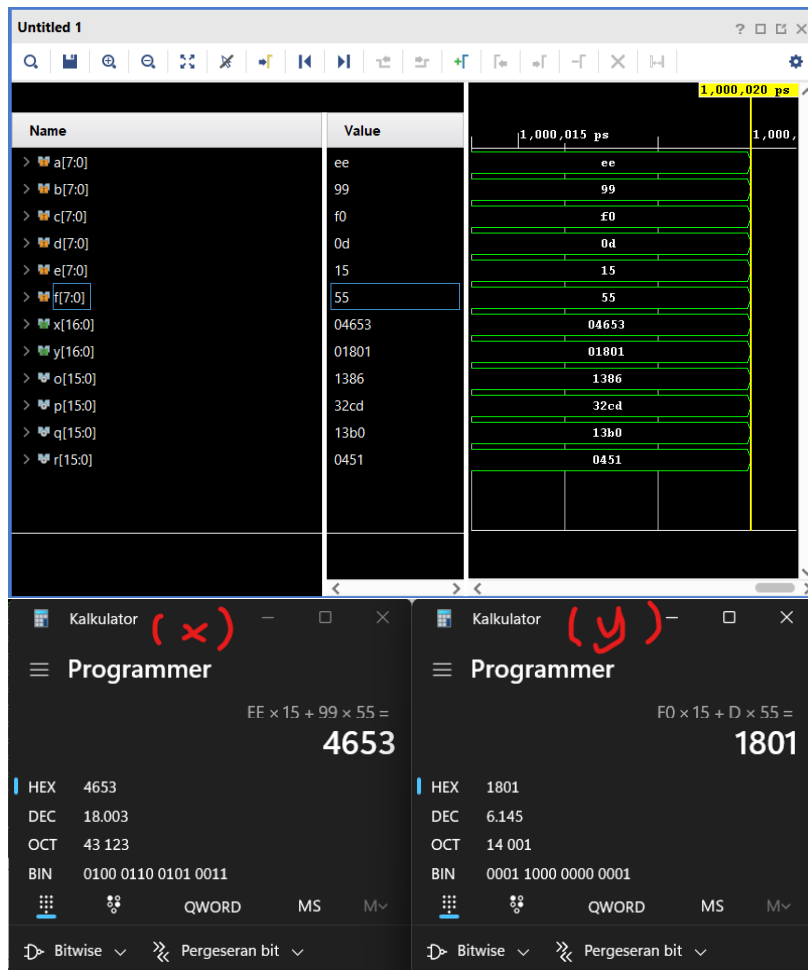
1) Input yang digunakan adalah sebagai berikut

$$\begin{pmatrix} 9A & 51 \\ 51 & 17 \end{pmatrix} \begin{pmatrix} 12 \\ A1 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$



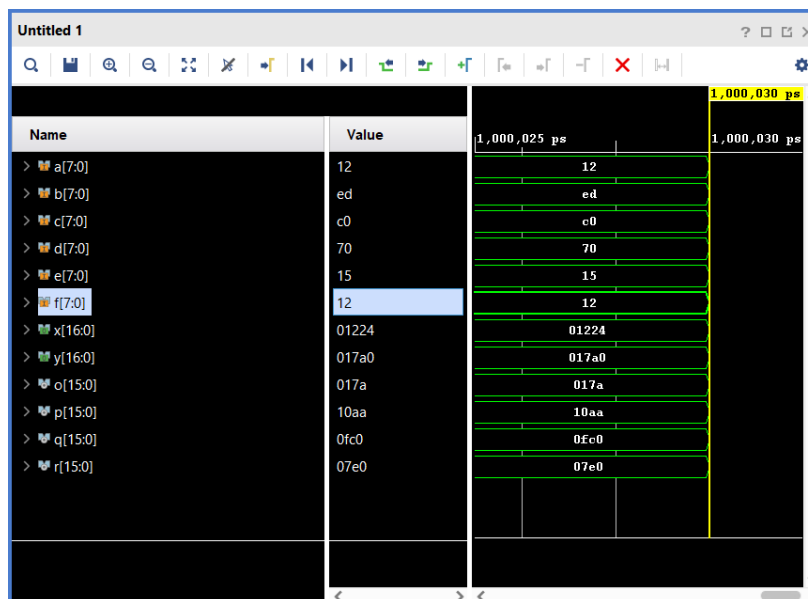
2) Input yang digunakan adalah sebagai berikut

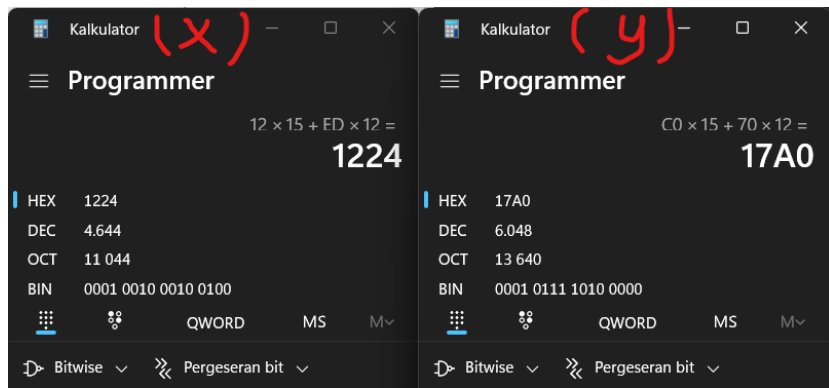
$$\begin{pmatrix} EE & 99 \\ F0 & 0D \end{pmatrix} \begin{pmatrix} 15 \\ 55 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$



3) Input yang digunakan adalah sebagai berikut

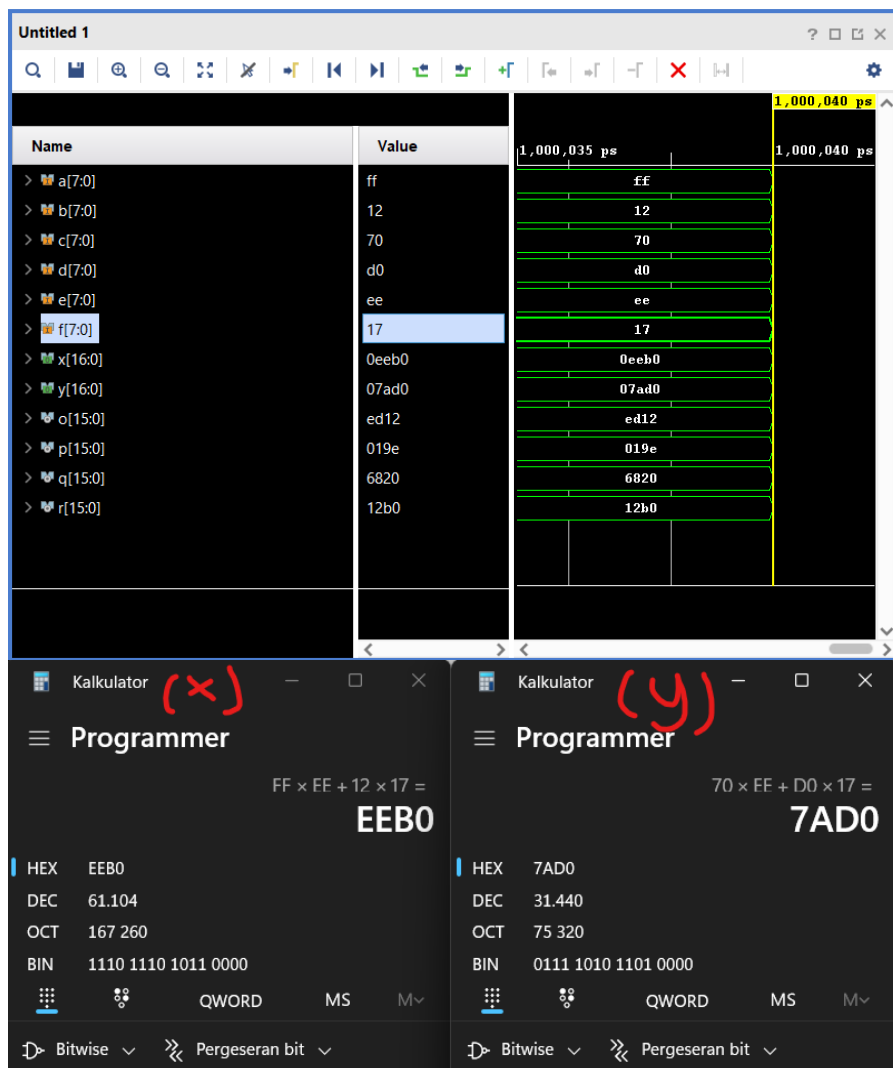
$$\begin{pmatrix} 12 & ED \\ C0 & 70 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$





4) Input yang digunakan adalah sebagai berikut

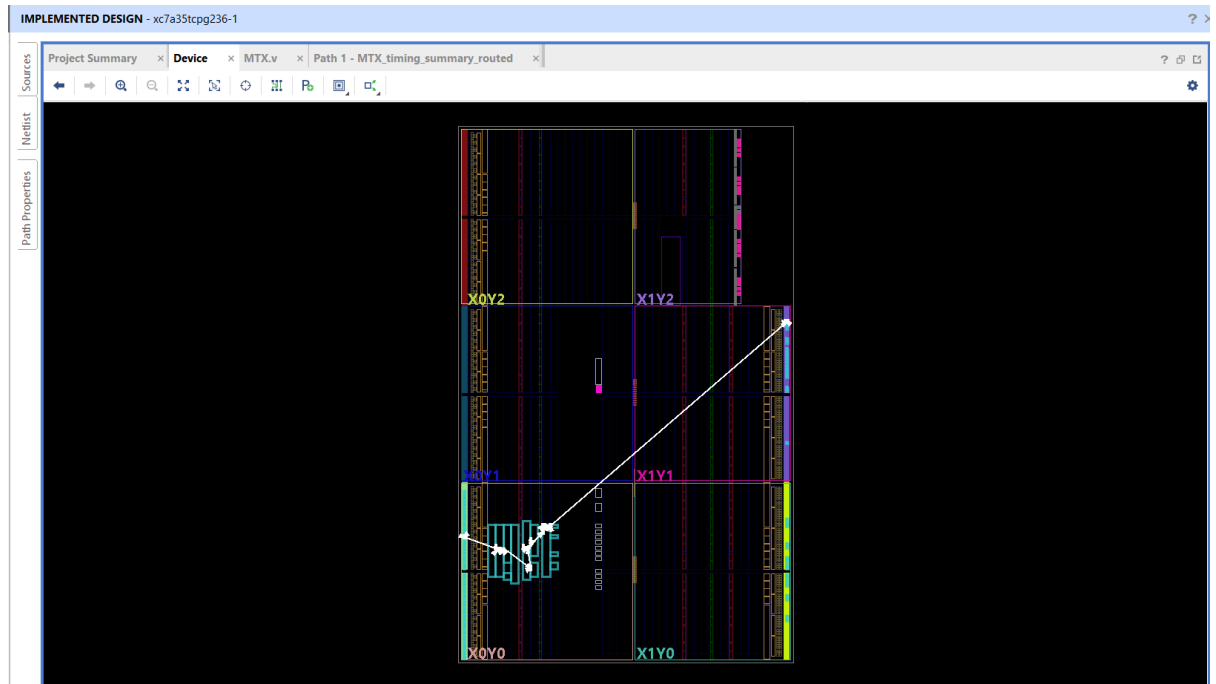
$$\begin{pmatrix} FF & 12 \\ 70 & D0 \end{pmatrix} \begin{pmatrix} EE \\ 17 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$



Berdasarkan 4 pengujian yang telah dilakukan dapat disimpulkan bahwa mikroarsitektur yang telah dibuat telah berfungsi sebagaimana fungsi yang seharusnya.

4. Critical Path

Critical path adalah jalan terjauh atau proses yang paling lama dari sistem yang telah dibuat. Pada mikroarsitektur operasi dari matrix ini, critical path dapat dijelaskan yaitu sebagai berikut. Yaitu dengan menggunakan fitur run implementation sehingga didapat beberapa data berikut.



Gambar Device dari mikroarsitektur MTX(Perkalian Matrix)

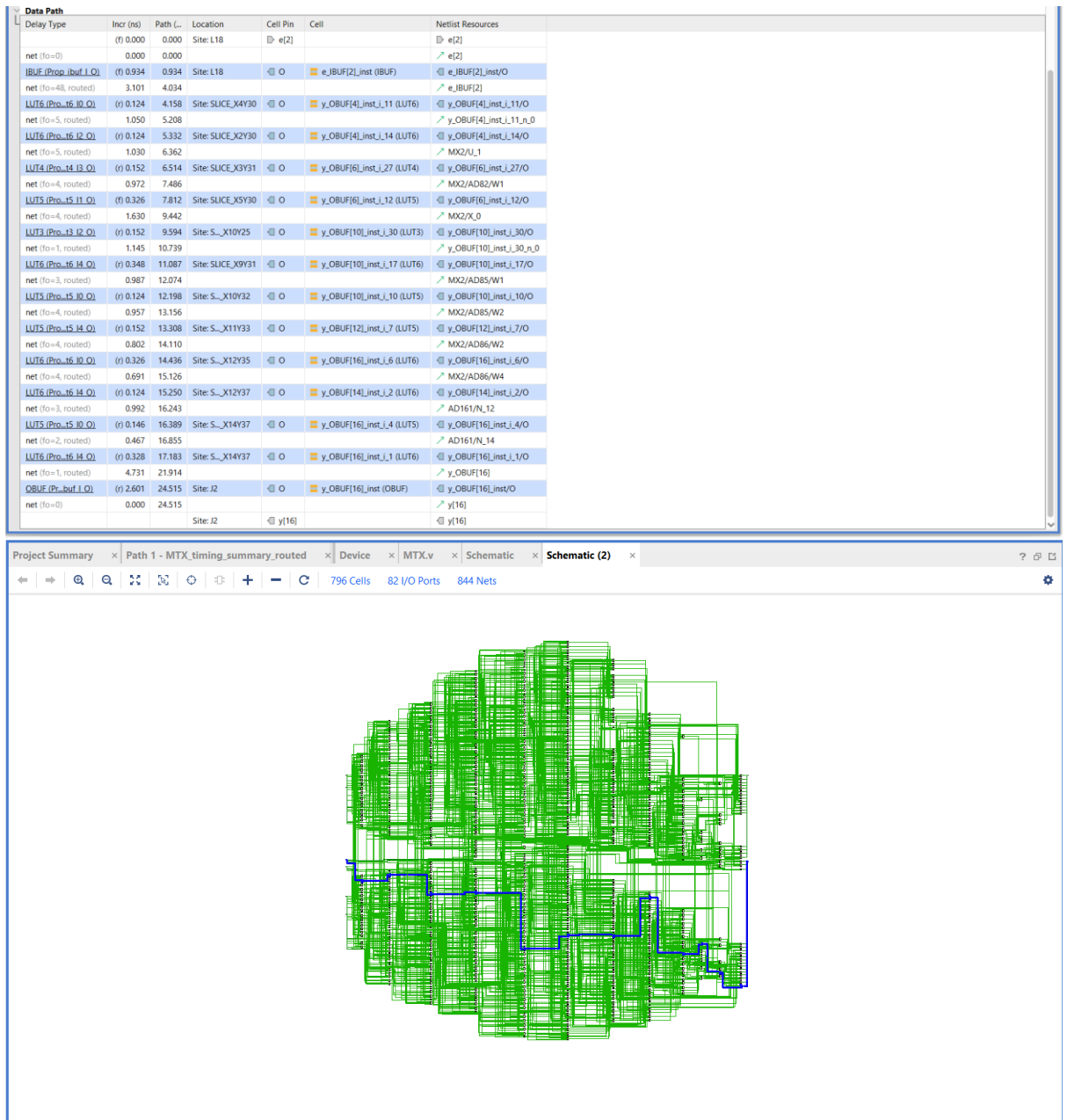
Timing													
Unconstrained Paths - NONE - NONE - Setup													
General Information	Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Clock Uncertainty
Timer Settings	Path 1	∞	14	48	e[2]	y[16]	24.515	5.961	18.555	∞	input port clock		0.000
Design Timing Summary	Path 2	∞	14	48	e[2]	y[15]	24.056	5.957	18.099	∞	input port clock		0.000
Methodology Summary	Path 3	∞	13	48	e[2]	y[13]	23.981	5.609	18.372	∞	input port clock		0.000
Check Timing (0)	Path 4	∞	13	48	e[2]	y[14]	23.669	5.608	18.061	∞	input port clock		0.000
Intra-Clock Paths	Path 5	∞	13	48	e[2]	y[12]	23.463	5.847	17.617	∞	input port clock		0.000
Inter-Clock Paths	Path 6	∞	12	48	e[2]	y[10]	22.972	5.482	17.490	∞	input port clock		0.000
Other Path Groups	Path 7	∞	14	44	e[0]	x[16]	22.953	5.488	17.465	∞	input port clock		0.000
User Ignored Paths	Path 8	∞	14	44	e[0]	x[15]	22.943	5.473	17.471	∞	input port clock		0.000
Unconstrained Paths	Path 9	∞	12	48	e[2]	y[11]	22.533	5.270	17.263	∞	input port clock		0.000
NONE to NONE	Path 10	∞	13	44	e[0]	x[14]	22.193	5.136	17.057	∞	input port clock		0.000
Setup (10)													

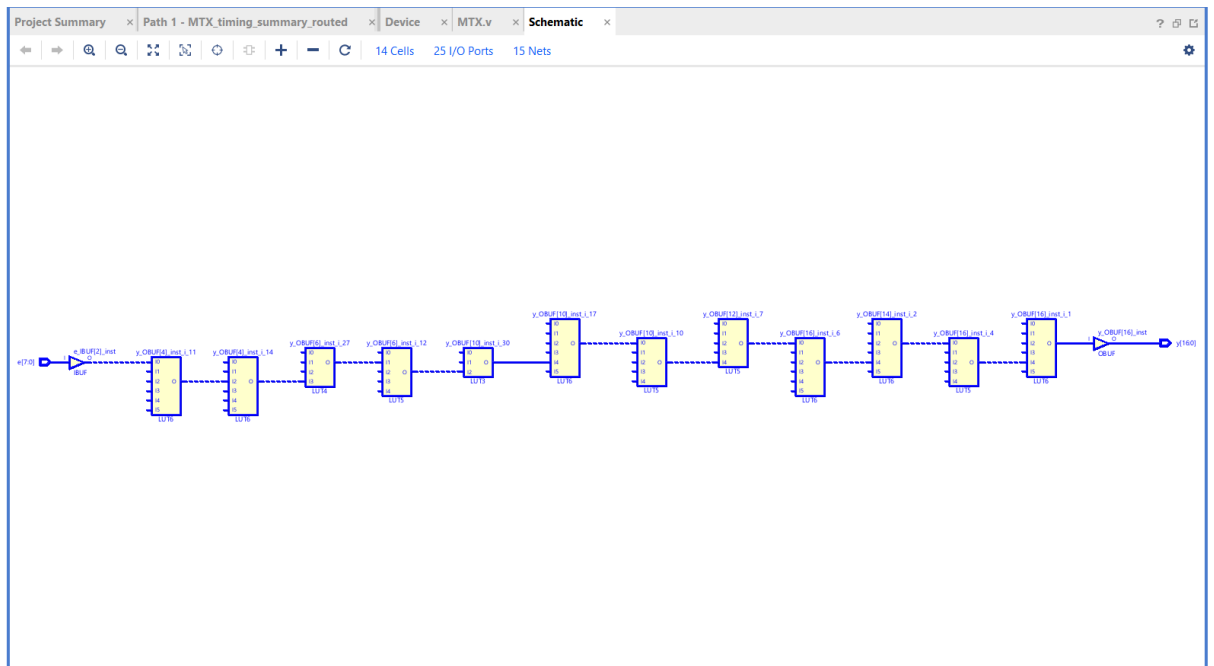
Combinational Delays					
From Port	To Port	M a	Max Process Corner	Min Delay	Min Process Corner
e[2]	y[16]	24.515	SLOW	5.760	FAST
e[2]	y[15]	24.056	SLOW	5.671	FAST
e[2]	y[13]	23.981	SLOW	5.782	FAST
e[2]	y[14]	23.669	SLOW	5.656	FAST
e[2]	y[12]	23.463	SLOW	5.767	FAST
c[2]	y[16]	23.420	SLOW	5.381	FAST
c[0]	y[16]	23.411	SLOW	5.892	FAST
f[2]	y[16]	23.289	SLOW	5.135	FAST
f[1]	y[16]	23.280	SLOW	5.192	FAST
c[3]	y[16]	23.220	SLOW	5.146	FAST
c[1]	y[16]	23.158	SLOW	5.566	FAST
e[0]	y[16]	23.111	SLOW	5.754	FAST
e[1]	y[16]	23.002	SLOW	5.472	FAST
e[2]	y[10]	22.972	SLOW	5.679	FAST

Gambar Beberapa Path serta keterangan waktu yang diperlukan

Summary	
Name	Path 1
Slack	∞ns
Source	e[2] (input port)
Destination	y[16] (output port)
Path Group	(none)
Path Type	Max at Slow Process Corner
Requirement	∞ns
Data Path Delay	24.515ns (logic 5.961ns (24.314%) route 18.555ns (75.686%))
Logic Levels	14 (IBUF=1 LUT3=1 LUT4=1 LUT5=4 LUT6=6 OBUF=1)

Gambar Kesimpulan dari Jalur dengan delay terlama





Gambar jalur yang dilalui

Berdasarkan data-data yang telah dipaparkan pada gambar gambar diatas critical path berada pada jalur input e[2] ke output y[16] dengan memiliki total delay terlama dibandingkan dengan jalur-jalur lain yang ditempuh dengan memiliki maximal delay selama 24.515 ns. Oleh karena itu jalur input e[2] ke output y[16] merupakan critical path dari mikroarsitektur perkalian matrix ordo 2x2 dengan 2x1 atau dalam hal ini dinamakan dengan module MTX.