# MODUL 7 SIMULASI RANGKAIAN MUX DAN DEMUX

# 7.1 Tujuan Praktikum Modul 7

Setelah mempraktekkan Topik ini, mahasiswa diharapkan dapat :

- 1. Dapat mengetahui dan memahami konsep dasar dari rangkaian *multiplexer* dan *demultiplexer*.
- 2. Dapat membuat rangkaian *multiplexer* dan *demultiplexer* dengan menggunakan bahasa VHDL atau Verilog.

# 7.2 Dasar Teori Praktikum Modul 7

## 7.2.1 Multiplexer

Multiplexer (MUX) adalah perangkat yang memungkinkan informasi digital dari beberapa input dapat dialihkan kedalam satu jalur output untuk menuju tujuan bersama. Dengan kata lain, rangkaian ini mengacu pada memilih satu output dari banyak input yang tersedia.

Untuk memahami *multiplexer* adalah dengan melihat *single pole multi- positioned* seperti gambar dibawah. Disini terdapat *switch* dengan banyak *input* (D0, D1, D2, dan D3) tetapi hanya memiliki satu pin *output* (*out*). Tombol kontrol digunakan untuk memilih satu dari empat data yang tersedia dan data ini akan tercermin di sisi *output*. Dengan ini, pengguna dapat memilih sinyal yang diperlukan diantara banyak sinyal yang tersedia.





Ini adalah contoh sederhana dari *multiplexer* mekanik. Tetapi, dalam sirkuit elektronik yang melibatkan perpindahan kecepatan tinggi dan transfer data mengharuskan agar dapat memilih input yang dibutuhkan dengan sangat cepat menggunakan sirkuit digital. Sinyal kontrol (S1 dan S0) melakukan hal yang sama, yaitu memilih satu *input* dari banyak yang tersedia berdasarkan dari banyak sinyal yang diberikan.

Jadi, ada tiga persyaratan minimum pada *multiplexer* apapun adalah sebagai berikut:

- **Pin** *Input*: ini adalah sinyal yang tersedia yang harus dipilih. Sinyal-sinyal ini dapat berupa sinyal digital atau analog.
- **Pin** *Output*: pin yang menyediakan keluaran sinyal dari pin *input* yang dipilih. *Multiplexer* akan selalu memiliki satu pin *input*.
- **Pin kontrol:** digunakan untuk memilih sinyal dari pin input. Jumlah kontrol pin pada multiplexer tergantung pada jumlah pin input. Misalnya *multiplexer* 4-*input* akan memiliki 2 pin sinyal.

Tabel kebenaran di bawah menggambarkan status pin kontrol (S0 dan S1) dari gambar diatas.



## 7.2.2 Tipe Multiplexer

#### 7.2.2.1 2-Channel Multiplexer

Pada 2-*channel Multiplexer* akan memiliki 2 *input* dan satu *output*. Juga hanya memiliki satu pin kontrol untuk memilih antara 2 pin *input* yang tersedia.

Gambar 7.3 Rangkaian 2-channel multiplexer



Ketika pada S0 ditetapkan logika 0 maka input D0 akan tercermin pada pin output dan jika S0 tetap logika 1 maka input D1 yang akan tercermin pada pin output. Dapat ditujukan pada tabel kebenaran dibawah ini:

<b>SO</b>	D0	D1	Out
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Tabel 7.1 Rangkaian 2-channel multiplexer

#### 7.2.2.2 4-Channel Multiplexer

Pada 4-Channel multiplexer akan memiliki 4 pin input dan 1 pin output dengan 2 pin kontrol. Dua pin kontrol ini akan membentuk 4 sinyal logika yang berbeda dan untuk setiap sinyal satu input tertentu akan dipilih. Multiplexer ini merupakan kombinasi dari 3 buah 2-channel multiplexer.



Gambar 7.4 Rangkaian 4-channel multiplexer

Pin kontrol dari dua MUX pertama dihubungkan bersama untuk membentuk pin kontrol pertama (S0) dan kemudian pin kontrol MUX ke tiga digunakan sebagai pembentuk pin kontrol kedua (S1). Dengan demikian, mendapatkan *multiplexer* dengan 4 *input* (W0, W1, W2, W3) dan hanya satu *output* (f). Tabel kebenaran untuk *multiplexer* 4:1 ditunjukan dibawah ini.

<b>SO</b>	<b>SO</b>	Out (f)
0	0	Wo
0	1	W1
1	0	W2
1	1	W <sub>3</sub>

Tabel 7.2 Tabel kebenaran 4-channel multiplexer

## 7.2.2.3 8-Channel Multiplexer

8-Channel Multiplexer merupakan kombinasi dari 2 buah 4-channel multiplexer dan 1 buah 2-channel multiplexer. Sehingga diperoleh 8 input, 3 pin kontrol, dan 1 output.





Tabel kebenaran dari 8-Channel Multiplexer seperti yang ditunjukan dibawah ini:

Se	lection Inp	Output	
S <sub>2</sub>	s <sub>1</sub>	s <sub>0</sub>	Y
0	0	0	I <sub>0</sub>
0	0	1	I1
0	1	0	I <sub>2</sub>
0	1	1	I3
1	0	0	I4
1	0	1	I <sub>5</sub>
1	1	0	I <sub>6</sub>
1	1	1	I <sub>7</sub>

Tabel 7.3 Tabel kebenaran 8-channel multiplexer

# 7.2.3 Implementasi Multiplexer

Digunakan untuk transmisi jaringan jarak jauh baik yang menggunakan kabel maupun yang menggunakan media udara seperti wireless atau radio.Sebagai contoh satu helai optic Surabaya–Jakarta bisa dipakai untuk menyalurkan ribuan percakapan pada telepon. Dan juga digunakan untuk remot TV (atau sejenisnya) dan kalkulator yang mempunyai beberapa inputan ke dalam satu outputan.

## 7.2.4 Demultiplexer

*Demultiplexer* (DEMUX) adalah rangkaian logika kombinasional yang dirancang untuk mengalihkan satu jalur input ke salah satu dari beberapa jalur output. Dengan kata lain, demultiplexer merupakan kebalikan dari *multiplexer*.

Pada *demultiplexer*, masukkan data dapat terdiri dari beberapa bit. Keluarannya terdiri dari beberapa jalur, masing-masing jalur terdiri dari satu bit atau lebih. Bit untuk masukan pada selektor (kontrol) tergantung banyaknya jalur.



#### 7.2.5 Tipe Demultiplexer

## 7.2.5.1 1 To 4 Demultiplexer

1 *to* 4 *Channel Demultiplexer* terdiri dari satu *input*, empat *output*, dan dua pin kontrol untuk membuat pilihan. Diagram dibawah menunjukkan rangkaian 1 *to* 4 *Channel Demultiplexer*.





Data D adalah bit *input* dengan 2 pin kontrol yaitu A dan B. Bit *input* D ditransmisikan ke empat bit *output* yaitu Y0, Y1, Y2, Y3. Ketika AB adalah 0, gerbang AND kedua paling atas diaktifkan sedangkan gerbang AND lainnya dinonaktifkan. Dengan demikian, hanya satu data yang dikirimkan pada Y1. Jika D rendah, maka Y1 rendah dan jika D tinggi, Y1 tinggi. Nilai Y1 tergantung pada nilai D.

Jika input kontrol berubah menjadi AB =10, semua gerbang dinonaktifkan kecuali gerbang ketiga dari atas. Kemudian D ditransmisikan ke output Y2. Berikut adalah tabel kebenaran untuk 1 to 4 Channel Demultiplexer.

Input	Select Lines	Output Lines
Ι	$S_1 S_0$	$D_0 D_1 D_2 D_3$
Ι	0 0	1000
I	0 1	0 1 0 0
I	1 0	0010
Ι	1 1	0 0 0 1

Tabel 7.4 Tabel kebenaran 1 to 4 demultiplexer

# 2.2.5.2 1 To 8 Demultiplexer

1 to 8 Demultiplexer terdiri dari satu input, 8 output, dan 3 pin kontrol.





Dibawah ini adalah tabel kebenaran untuk 1-to-8 Demultiplexer. Jika S0S1S2 = 000, maka output yang aktif adalah Y0 dan seterusnya.

Data Input	Se	lect Inp	uts Outputs								
D	<b>S</b> <sub>2</sub>	<b>S</b> <sub>1</sub>	S <sub>0</sub>	¥,	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Yo
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

Tabel 7.5 Tabel kebenaran 1 to 8 demultiplexer

# 7.2.6 Implementasi Demultiplexer

Demultiplexer Biasanya digunakan pada televisi karena cara televisi adalah menerima sinyal data yang kemudian akan dipisahkan berdasarkan chanel yang ada cara ini disebut dengan teknik demultiplexing atau lebih jelasnya adalah proses penerimaan data dan kemudian akan dipisahkan sesuai dengan channel yang ada.

# 7.3 Lembar Kegiatan Praktikum Modul 7 :

# 7.3.1 Alat dan Bahan

- a. Laptop yang telah terinstal software Quartus 18
- b. Mouse

## 7.3.2 Langkah percobaan modul 7 (4-channel multiplexer)

- 1. Hidupkan Laptop
- 2. Buka aplikasi quartus
- 3. Pilih New Project Wizard → next

	• 0 9 3 ×	ð	Home	0			IP Catalog	20
compilation Hierarchy							Device Family Cyclone IV	
		Recent Projects					<b>R</b>	×
		p1.qpf (C/Intel/FPGA) and and aC descriptions	Nto/18.1/modul77/p1.opf) Nto/18.1/modul77/p1.opf)				🛩 🞲 Installed IP	
		Proget Control Man					<ul> <li>Project Directory</li> </ul>	
		nios2demozg#10 %					V Library	sbie.
					-		> Basic Functions	
				New Project Woord	Open Project		> DSP	
							Interface Protocols	
				14 III.	0 0	-9	<ul> <li>Mensory Interfaces</li> <li>Depressors and Data</li> </ul>	and Controller
		Com	NZEERIOIS Buy Software D	scomentation. Training	Support What's New	Not fications	University Program	1
							Search for Partner IP	
s Compilation	• = 9 • •							
Task	^							
✓ ▶ Compile Design								
> 🕨 Analysis & Synthe	eis							
<ol> <li>Primer (Place &amp; Ros</li> </ol>	ere)							
> Masembler (Gener	ate programm							
> P Timing Analysis		-						
P POR NUMBER WITH	· ·	Close page after project fixed					intel)	
THE REPORT OF THE PARTY OF THE	>	Department the screen again					+ Adl.	
	Y ««Filter»»		66 Find. 66 Find Next					
a set of the set of th								

4. Menentukan directory dan nama project  $next \rightarrow next \rightarrow next$ 

What is the working directory for this pro	ject?				
C:\altera\14.1\modul7					
What is the name of this project?					
praktikum7					
What is the name of the top-level design name in the design file.	entity for this project? Th	is name is case se	nsitive and must	exactly match the	entity
praktikum7					

5. Menentukan device family dengan MAX 10(DA/DF/DC/SA/SC) dan name filter dengan 10M50DAF484C7G → klik 10M50DAF484C7G pada available device → next → next → finish

Device far	nily			Show in 'Availabl	e devices' list		
Family:	MAX 10 (DA/DF/DC/	SA/SF/SC)	-	Package:	Any		•
Devices	a: All		•	Pin count:	Any		•
Tarant davies				Core Speed grad	e: Anv		•
Auto	A base device				Name filter: 10M50DAF484C7G		
Other	: n/a						
vailable de	evices:						
vailable de	evices: Name	Core Voltage	LEs	User I/Os M	lemory Bits	Embedde	ed multiplier 9-b

6. Pilih menu file → new → pilih Block Diagram/Schematic File → ok



7. Membuat rangkaian multiplexer, klik Symbol Tool → klik tanda panah → primitives → logic → pilih gerbang logika yang ada. Dan buat inputan serta outputan dari rangkaian tersebut, klik Pin Tool → pilih input serta output sesuaikan dengan rangkaian tersebut, serta rename pin input dan output dan wiring sesuai dengan rangkaian.



8. Setelah itu, klik Compile design. Tunggu hingga berhasil

9. Pilih menu file  $\rightarrow$  new  $\rightarrow$  pilih Univertity Program VWF  $\rightarrow$  ok



10. Pilih menu Edit → Insert → Insert Node or Bus → Pilih Node Finder

×	Delete De	1 <u>2</u> X2 X	R. R. 25	(11) 明				
s	Insert	<ul> <li>Inser</li> </ul>	t Node or Bus	22.48	Name:	-		OK
	Value Grouping Reverse Group or Bus Bit Order	• .0 ns	160,0 ns	240	Type:	INPUT	•	Cancel
	Radix	•			Value type	9-Level	•	
~	Grid Size Set End Time Snap to Grid				Radix:	Binary	•	Node Finder.
	Snap to Transition	_			Bus width:	1		
162	Properties	_			Start index	0		

11. Klik List  $\rightarrow$  klik ">>"  $\rightarrow$  ok  $\rightarrow$  ok

amed:	Filter:	Pins: all	<b>.</b>	ОК
ook in: 💌			List	Cancel
odes Found:		Selected	Nodes:	
Name 6 d0 6 d1 6 d2 6 d3 6 s0 6 s1 7 x	Type Input Input Input Input Input Output	Na d0 1 d1 2 d3 d3 c m d3 c m s1 out x	me Input Input Input Input Input Input Output	Туре

- 12. Lakukan simulasi pada rangkaian di atas dengan data masukan sebagai berikut :
  - Masukan D0 : clock periode 120ms dan duty cycle 50%
  - Masukan D1 : clock periode 150ms dan duty cycle 50%
  - Masukan D2 : clock periode 180ms dan duty cycle 50%
  - Masukan D3 : clock periode 210ms dan duty cycle 50%
  - Masukan S0 : clock periode 50ms dan duty cycle 50%
  - Masukan S1 : clock periode 100ms dan duty cycle 50%

Dengan cara klik kiri pada  $D0 \rightarrow$  klik menu Edit  $\rightarrow$  Value  $\rightarrow$  Overwrite Clock  $\rightarrow$  masukan clock periode sesuai dengan masukan tersebut. lakukan langkah tersebut dari D0 sampai S1.

Edit V	ew Simulation	Help 🔊	Clock	1
aster Time E	& 안 圥 <b>프</b> ar: 0 ps	VE VE W VC VC V2 VE R A A	Base waveform on t	ime period
Name	Value at 0 ps	0 ps 80.0 ns 160.0 ns 0 ps	Period: 120	ns 🔻
_ d0	BO		Offset: 0.0	nc 🔻
. d1	B 0		onset. 0.0	112
d2	BO		Duby cycle (% 50	<b>A</b>
d3	B 0		Ducy Cycle ( 70. 50	
s0	BO			
s1	BO		OK	Cancel
1 ×	BY	622222222222222222222222222222222222222	UK	Cancer

13. Lakukan simulasi dengan cara klik pada menu Simulation → Run Timing
 Simulation → tunggu hingga selesai.



14. Setelah selesai simulasi, kembali ke halaman awal rangkaian. Pilih menu **assignments → pin planner** 

Ass	ignments	Processing	Tools	Window	Help
٠	Device				
4	Settings			Ctrl+Shi	ft+E
4	Assignme	ent Editor		Ctrl+Shi	ft+A
4	Pin Plann	er		Ctrl+Shi	ft+N
	Remove A	Assignments			

15. Berikan location pin assignment pada fisik DE10-LITE sesuai dengan datasheet → Run I/O assignment analysis.





16. Compile design. Tunggu hingga selesai

# 7.3.3 Langkah percobaan modul 7 (1 to 4 demultiplexer)

- 1. Hidupkan Laptop
- 2. Buka aplikasi quartus
- 3. Pilih New Project Wizard → next



4. Menentukan directory dan nama project  $\rightarrow$  next  $\rightarrow$  next  $\rightarrow$  next

New Project Wizard			×
Directory, Name, Top-Level B	Intity		
What is the working directory for this proje	ct?		
C:\intelFPGA_lite\18.1\modul77			
What is the name of this project?			
p1			
What is the name of the top-level design en match the entity name in the design file.	tity for this project? This r	name is case sensitive and mu	ist exactly
p1			
USE EXISTING I OJECE SECURGS			
	< Back Next >	Finish Cancel	Help

5. Menentukan device family dengan MAX 10 (DA/DF/DC/SA/SC) dan name filter dengan 10M50DAF484C7G → klik 10M50DAF484C7G pada available device → next → next → finish

Device family			Show in 'Availab	le devices' list	
Family: MAX 10 (DA/DF/D	C/SA/SF/SC)	-	Package:	Any	
Devices: All		-	Pin count:	Any	
Target device		Core Speed grade:	de: Any	Any 10M50DAF484C7G	
		Name filter:	10M50DAF		
Other: n/a	a in Available devices i	ISL	Show advan	ced devices	
Available devices					
Available devices: Name	Core Voltage	LEs	User I/Os I	Memory Bits	Embedded multiplier

6. Pilih menu **file → new →** pilih **Block Diagram/Schematic File → ok** 



7. Membuat rangkaian multiplexer, klik Symbol Tool → klik tanda panah → primitives → logic → pilih gerbang logika yang ada. Dan buat inputan serta outputan dari rangkaian tersebut, klik Pin Tool → pilih input serta output sesuaikan dengan rangkaian tersebut, serta rename pin input dan output dan wiring sesuai dengan rangkaian.



- 8. Setelah itu, klik Compile design. Tunggu hingga berhasil
- 9. Pilih menu **file → new →** pilih **Univertity Program VWF→ ok**





D.	×	Delete Del		5 XZ XB	R R R	
lad		Insert	•	Insert M	lode or Bus	22.48
100		Value	•			Lerio
		Grouping Reverse Group or Bus Bit Order	•	.0 ns	160,0 ns	240
		Radix	•			
		Grid Size Set End Time				
	~	Snap to Grid				
		Snap to Transition				
	8	Properties				

Name:	1		ОК
Type:	INPUT	•	Cancel
Value type	9-Level	•	
Radix:	Binary	•	Node Finder
Bus width:	1		
Start inde>	0		

11. Lalu klik List  $\rightarrow$  klik ">>"  $\rightarrow$  ok  $\rightarrow$  ok

Named: *	Filter	: Pins:	all	-	ОК
Look in: *			L	.ist	Cancel
Nodes Found	d:		Selected No	des:	
Name	Туре		Name	1	Гуре
A 📥	Output	>	A 🍋	Output	
🛎 B	Output	>>	🗳 B	Output	
🗳 C	Output		C C	Output	
🎬 D	Output	<	🎬 D	Output	
📙 F	Input	<<	<b>₽</b> - F	Input	
🖺 s0	Input		<b>i</b> ∎– s0	Input	
<mark>in</mark> – s1	Input		<mark>in</mark> _ s1	Input	

- 12. Lakukan simulasi pada rangkaian di atas dengan data masukan sebagai berikut :
  - Masukan S0 : clock periode 50ms dan duty cycle 50%
  - Masukan S1 : clock periode 100ms dan duty cycle 50%
  - Masukan F : clock periode 400ms dan duty cycle 50%

S Clock X	Simulation Waveform Editor - C:/intel	IFPGA_lite/18.1/modul77/p1 - p1 - [Waveform.vwf]*
Base waveform on time period	Mas Value	Del ) 语 成 成 和 國 際 , 35 0 ns Interval: 35 0 ns Start: 0 ns , 签 Forcing Unknown (X) Ctrl+Alt+X
Period: 400.0 ns 🔻	Grouping Reverse Group or Bus Bit Ord Radix	er A Forcing Low (0) Ctrl+Alt+0 A Forcing High (1) Ctrl+Alt+1 Z High Impedance (2) Ctrl+Alt+2
Offset: 0.0 ns Duty cycle (' 50	<ul> <li>Grid Size</li> <li>Set End Time</li> <li>Snap to Grid</li> <li>Snap to Transition</li> </ul>	XE         Weak Low (L)         Ctrl+Alt+L           XE         Weak High (H)         Ctrl+Alt+L           Weak High (H)         Ctrl+Alt+L           Weak New York         Ctrl+Alt+L           Weak New York         Ctrl+Alt+L           Weak New York         Ctrl+Alt+L
OK Cancel	Properties	XZ         Overwrite Clock         Ctrl+Alt+K           XZ         Arbitrary Value         Ctrl+Alt+B           XE         Random Values         Ctrl+Alt+R

Dengan cara klik kiri pada D0 → klik menu Edit → Value → Overwrite Clock → masukan clok periode sesuai dengan masukan tersebut. lakukan langkah tersebut dari D0 sampai S1.

13. Lakukan simulasi dengan cara klik pada menu Simulation → Run Timing
 Simulation → tunggu hingga selesai.



14. Setelah selesai simulasi, kembali ke halaman awal rangkaian. Pilih menu assignments → pin planner

Ass	ignments	Processing	Tools	Window	Help
•	Device				
2	Settings			Ctrl+Shi	ft+E
¢	Assignment Editor			Ctrl+Shi	ft+A
4	Pin Planner			Ctrl+Shi	ft+N
	Remove A	ssignments			

15. Berikan location pin assignment pada fisik DE10-LITE sesuai dengan datasheet → Run I/O assignment analysis.



# 16. Compile design. Tunggu hingga selesai



# 7.4 Soal Jurnal

1. Buatlah rangkaian berikut (multiplexer) pada buku jurnal kemudian analisa input serta outputnya.



4	В	С	Y
)	0	0	
)	0	1	
)	1	0	
)	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

2. Buatlah rangkaian berikut (demultiplexer) pada buku jurnal kemudian analisa input serta outputnya.



A	S	Y 0	Y1
0	0		
0	1		
1	0		
1	1	•••	

3. Tuliskan apa yang telah dilakukan pada praktikum modul 7 menggunakan Bahasa kalian sendiri!