DATA STREAM MULTIPLEXER

Design Specifications:

Inputs

clk: Internal clock signal operating at 100 MHz.

symbol_clk: External clock synchronizing incoming data streams, frequency ranging from 1 kHz to 50 MHz.

mode: 2-bit input to determine the desired mode of operation (values 00, 01, 10, or 11).

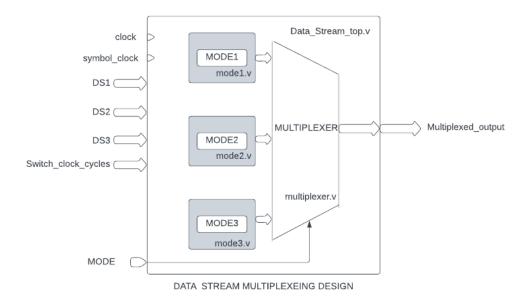
switch_clk_cycles: 32-bit input representing the number of clk cycles after which the data stream in the output should switch to the next one.

DS1, DS2, DS3: 16-bit data streams synchronized with the symbol clk.

Outputs

output_data: 16-bit output data stream representing the selected data stream based on the mode of operation.

Block Diagram



Code

DataStream top.v

`timescale 1ns / 1ps

module DataStream_top(input clk,

```
input symbol_clk,
           input [31:0]switch_clock_cycles,
           input [1:0] mode,
           input [15:0] DS1,
           input [15:0] DS2,
           input [15:0] DS3,
           output [15:0] multiplexed_data );
wire [15:0] out_1;
wire [15:0] out_2;
wire [15:0] out_3;
mode1 U1(clk,symbol_clk,switch_clock_cycles,DS1,out_1);
mode2 U2(clk,symbol_clk,switch_clock_cycles,DS1,DS2,out_2);
mode3 U3(clk,symbol_clk,switch_clock_cycles,DS1,DS2,DS3,out_3);
multiplexer U4(out_1,out_2,out_3,mode,multiplexed_data);
endmodule
Mode1.v
`timescale 1ns / 1ps
module mode1( input clk,
       input symbol_clk,
       input [31:0] switch_clk_cycles,
       input [15:0] DS1,
       output reg [15:0] out_1
       );
reg [15:0] count=1;
always@(posedge clk) begin
if (count < switch_clk_cycles) begin
out_1 <= DS1;
count <= count+1;</pre>
end
```

```
else count <= 1;
end
endmodule
Mode2.v
 `timescale 1ns / 1ps
module mode2(input clk,
       input symbol_clk,
       input [31:0] switch_clk_cycles,
       input [15:0] DS1,
       input [15:0] DS2,
       output reg [15:0] out_2 );
reg [15:0] count=1;
always @(posedge clk) begin
if (count <= (switch_clk_cycles)+1) begin</pre>
out_2<=DS1;
count<=count+1;</pre>
end
else if (count < (2*switch_clk_cycles) ) begin
out_2 <= DS2;
count<=count+1;</pre>
end
else count<=1;
end
endmodule
Mode3.v
`timescale 1ns / 1ps
module mode3(input clk,
       input symbol_clk,
```

```
input [31:0]switch_clk_cycles,
       input [15:0] DS1,
       input [15:0] DS2,
       input [15:0] DS3,
       output reg [15:0] out_3 );
reg [31:0] count=1;
always @(posedge clk) begin
if (count<(switch_clk_cycles)+1 ) begin
out_3<=DS1;
count<=count+1;
end
else if (count < (2*(switch_clk_cycles))+1) begin
out_3<=DS2;
count<=count+1;
end
else if ( count < 3*(switch_clk_cycles)+1) begin
out_3<=DS3;
count<=count+1;
end
else count<=1;
end
endmodule
Multiplexer.v
`timescale 1ns / 1ps
module multiplexer(input [15:0] out_1,
          input [15:0] out_2,
          input [15:0] out_3,
          input [1:0] mode,
          output reg [15:0] multiplexed_data
```

```
);
always @(*) begin
case(mode)
2'd1: multiplexed_data <=out_1;
2'd2:multiplexed_data <=out_2;
2'd3:multiplexed_data <=out_3;
endcase
end
endmodule
Test Bench
DataStream_top.v
`timescale 1ns / 1ps
module DataStream_tb;
reg clk;
reg symbol_clk;
reg [1:0] mode;
reg [31:0] switch_clock_cycles;
reg [15:0] DS1;
reg [15:0] DS2;
reg [15:0] DS3;
wire [15:0] multiplexed_data;
DataStream_top DUT (
  .clk(clk),
  .symbol_clk(symbol_clk),
  .mode(mode),
  .switch_clock_cycles(switch_clock_cycles),
  .DS1(DS1),
  .DS2(DS2),
  .DS3(DS3),
```

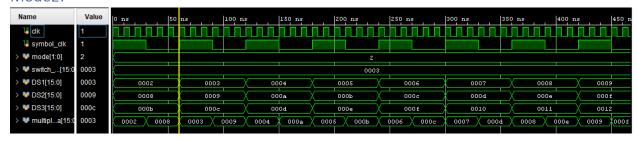
```
.multiplexed_data(multiplexed_data)
);
integer t_clk=5;
integer t_sym_clk=10;
initial begin
clk=1;
symbol_clk=1;
//switch_clock_cycles=(t_sym_clk/t_clk);//mode1
switch_clock_cycles=(t_sym_clk/t_clk)/2;//
//switch_clock_cycles=(t_sym_clk/t_clk)/3;//mode3
mode=2'b10;
DS1=16'd1;
DS2=16'd7;
DS3=16'd10;
end
always #t_clk clk=~clk;
always #t_sym_clk symbol_clk=~symbol_clk;
always @(posedge symbol_clk) begin
DS1=DS1+1;
DS2=DS2+1;
DS3=DS3+1;
end
endmodule
```

Simulation Results

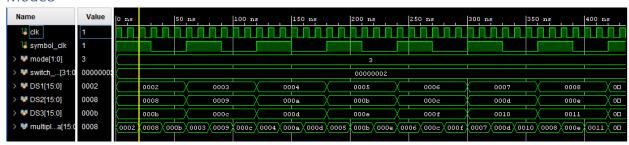
Mode1:

Name	Value	0 ns 5	0 ns 100 n	ns 150 ns	200 ns	250 ns	300 ns 35	0 ns 400 ns
₩ clk	1							
¼ symbol_clk	0							
> W mode[1:0]	1				1			
> 😻 switch[15:0	0006				0006			
> W DS1[15:0]	0012	0002	0003	0004	0005	0006	0007	X 0008
> W DS2[15:0]	0018	0008	0009	000a	000ь	000e	0004	(000e
> W DS3[15:0]	001b	000ъ	000c	0004	000e	000f	0010	0011
> W multipla[15:0	0012	0002	0003	0004	0005	0006	0007	0008

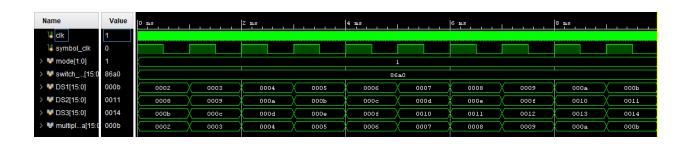
Mode2:



Mode3



Simulation results with different symbol_clk frequencies



\(\begin{align*} \begin{align*} \be	1	000 000 000)2 2 3 3 3 3 4	000 ns 1		1 0064	00 ns 3,0	00 ns 3,5	00 ns 4,00	00 ns
symbol_clk	1 1 0064 000d 0013 0016 000d	000)B	0005		0064				
■ M mode[1:0] 1 ■ M switch[15:0] 0 ■ DS1[15:0] 0 ■ DS2[15:0] 0 ■ DS3[15:0] 0 ■ multipla[15:0] 0 ■ t_clk[31:0] 0	1 0064 000d 0013 0016 000d	000)B	0005		0064				
w switch[15:0 0 DS1[15:0] 0 DS2[15:0] 0 DS3[15:0] 0 multipla[15:0 0 t_dk[31:0] 0	0064 000d 0013 0016 000d	000)B	0005		0064				
DS1[15:0] 0 DS2[15:0] 0 DS3[15:0] 0 DS3[15:0] 0 Multipla[15:0] 0 Lclk[31:0] 0	000d 0013 0016 000d 0000000!	000)B	0005						
DS2[15:0] 0 DS3[15:0] 0 Multipla[15:0] 0 Lck(31:0] 0	0013 0016 000d 0000000!	000)B	0005		0004				
 ✓ DS3[15:0] 0 ✓ multipla[15:0 0 ✓ t_clk[31:0] 0 	0016 000d 0000000:	000)b	0000	<u> </u>		X_	0005		0006
₩ multipla[15:0 0 ₩ t_clk[31:0] 0	000d 0000000!					000a		000ъ		000c
₩ t_clk[31:0] 0	0000000	000	02	0003	.	0004		000e		000f
					· X	0004		0005		0006
₩ t_sym31:0] 0	000001f4					00000005				
						000001f4				
	_									
lame	Value	0 ns	20 ns	40 ns	60 ns	80 ns	100 ns	120 ns	140 ns	160 ns
U clk 1	1				لأجرون					
	1					کید د				
	1					1				
	0002					0002				
	0034		V						V	V
		0002	0003	0004	0005	0006	0007	0008	0009	000a
	003a	0008	0009	000a	000ъ	000c	0004	000e	000f	0010
	003d	000ъ	000c	0004	000e	000f	0010	0011	0012	0013
™ multipla[15:0	0034	0002	0003	0004	0005	0006	0007	0008	0009	000a
U clk 1	Value 1	0 ns	20 ns	40 ns	60 ns	80 ns	100 ns	120 ns	140 ns	160 ns
	1					1				
₩ switch[15:0	0002					0002				
	0034	0002	0003	0004	0005	0006	0007	0008	0009	J
	003a	0008				A 3000				Y 000a
	0004	0008			000%	0000)	000a
• D35[13.0]	0034	000	0009	000a	000ъ	000c	0004	000e	000f	0010
M multipl of 15:0	003d	000Ъ	000c	0004	000e	000f	0010 0004	000e 0011	000f 0012	0010 0013
♥ multipla[15:0	003d 0034	000b 0002					0004	000e	000f	0010
		0002	000c 0003	000d	000e 0005	000 f 0006	0004 0010 0007	000e 0011 0008	000f 0012 0009	0010 0013 000a
Name	0034		000c	0004	000e	000f	0010 0004	000e 0011	000f 0012	0010 0013 000a
Name	0034 ; Value	0002	000c 0003	000d	000e 0005	000 f 0006	0004 0010 0007	000e 0011 0008	000f 0012 0009	0010 0013 000a
Name Ukulania	Value 1	0002	000c 0003	000d	000e 0005	000 £	0004 0010 0007	000e 0011 0008	000f 0012 0009	0010 0013 000a
Name U clk U symbol_clk	0034 ; Value	0002	000c 0003	000d	000e 0005	000f 0006	0004 0010 0007	000e 0011 0008	000f 0012 0009	0010 0013 000a
Name clk	Value 1 1 2 00000003	0002	000c 0003	0004 0004	000e 0005	000f 0006 4 us 2 00000032	0004 0010 0007	000e 0011 0008	000f 0012 0009	0010 0013 000a
Name clk	Value 1 1 2 00000033	0002	000c 0003	0004 0004	000e 0005	000f 0006	0004 0010 0007	000e 0011 0008	000f 001z 0009	0010 0013 000a
Name	Value 1 1 2 00000003	0002	000c 0003	0004 0004	000e 0005	000f 0006 4 us 2 00000032	0004 0010 0007	000e 0011 0008	000f 0012 0009	0010 0013 000a
Name	Value 1 1 2 00000033	0002	000c 0003	0004 0004	000e 0005	000f 0006 4 us 2 00000032	0004 0010 0007	000e 0011 0008	000f 001z 0009	0010 0013 000a 000a
Name	Value 1 1 2 0000003: 0017 001d 0020	0002	000c 0003 1 us 0003 0003 0009	0004 0004 2 us 0004 000a	000s 0005 3 us 000s 000b	000f 0006 0006 4 us 2 00000032 0006 000c	0004 0010 0007 5 us 0007 0004	000e 0011 0008 6 us 0008	000f 0012 0009 0009	0010 0013 000a 000a
Name	Value 1 1 2 0000003: 0017 001d 0020	0002 0002 0008 0000 0000	000c 0003 0003 0003 0009 000c 00003	0004 0004 0004 0004 000a 000d 9 0004 000	000s 0005 0005 000b 000e a 0005 000	000f 0006 0006 2 00000032 0006 0006 0006 000	0004 0010 0007 0007 0004 0010 c 0007 000	000e 0011 0008 0008 0008 000e 0011	000f 0012 0009 0009 0009 000f 0012	0010 0013 000a 000a 000 000 000 000
Name dk symbol_dk mode[1:0] switch[31:0] DS1[15:0] DS2[15:0] multipla[15:0] Mame	Value 1 2 00000033 0017 001d 0020 0017	0002 0 us 0002 0008 000b	000c 0003 1 us 0003 0003 0009	0004 0004 2 us 0004 000a	000s 0005 3 us 000s 000b	000f 0006 0006 2 00000032 0006 0006 0006 000	0004 0010 0007 5 us 0007 0004	000e 0011 0008 6 us 0008	000f 001z 0009 0009	0010 0013 000a 000a
Name dk	Value 1 1 2 00000033 0017 001d 0020 0017 Value	0002 0002 0008 0000 0000	000c 0003 0003 0003 0009 000c 00003	0004 0004 0004 0004 000a 000d 9 0004 000	000s 0005 0005 000b 000e a 0005 000	000f 0006 0006 2 00000032 0006 0006 0006 000	0004 0010 0007 0007 0004 0010 c 0007 000	000e 0011 0008 0008 0008 000e 0011	000f 0012 0009 0009 0009 000f 0012	0010 0013 000a 000a 000 000 000 0000

00008235

0002\0008\0008\0003\0003\0009\000c\0004\000a\000a\0003\0005\000b\000e\0006\000c\000f\0007\000a\0010\0008\000e\0001\0009

0006

0007

0010

0005

000ъ

0009

000f

0012

0008

000e

0011

0000823

0002

0008

0003

0009

000c

0004

000d

000c

0012

0015

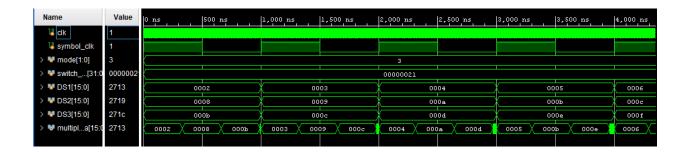
> 😽 switch_...[31:0

> W DS1[15:0]

> W DS2[15:0]

> W DS3[15:0]

₩ multipl...a[15:0 000c



Elaborated design:

