ECE2002 –Digital logic And Design

Embedded Lab-

Fall semester 2020~2021

Slot: L41+L42

E-Record

Experiment No. : 5

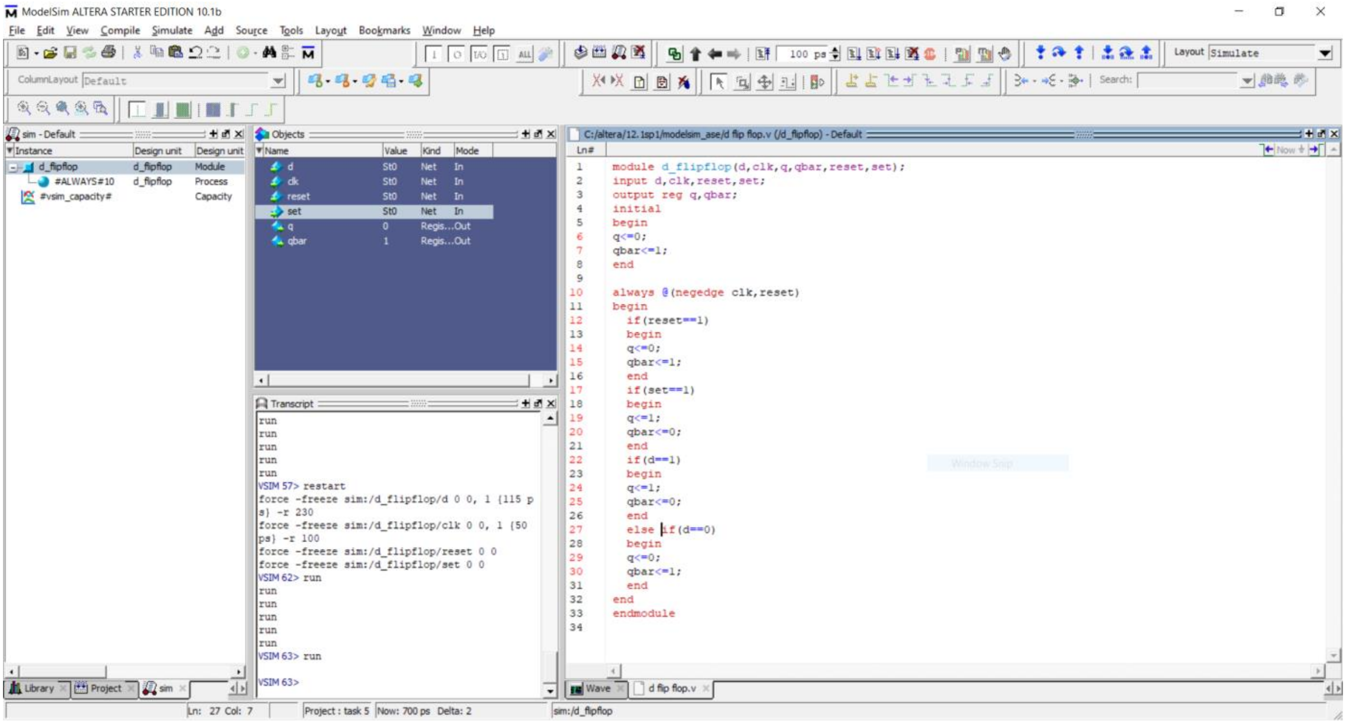
Submitted by

Name of the Student: ARPIT PATAWAT

Reg. no.: 19BEC0358

a) Write (any) one D flip flop, JK flip flop and T flip flop programs

ii) Write a Verilog program for **Synchronous Negative edge trigger "D" flip-flop with Asynchronous reset & synchronous set.**



ModelSim ALTERA STARTER EDITION 10.1b
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
Layout Simulate
ColumnLayout Default
sim - Default
Objects
Transcript
Library
Project
sim
Wave
d flip flop.v

Instance	Design unit	Design unit
d_flipflop	d_flipflop	Module
#ALWAYS#10	d_flipflop	Process
#vsim_capacity#		Capacity

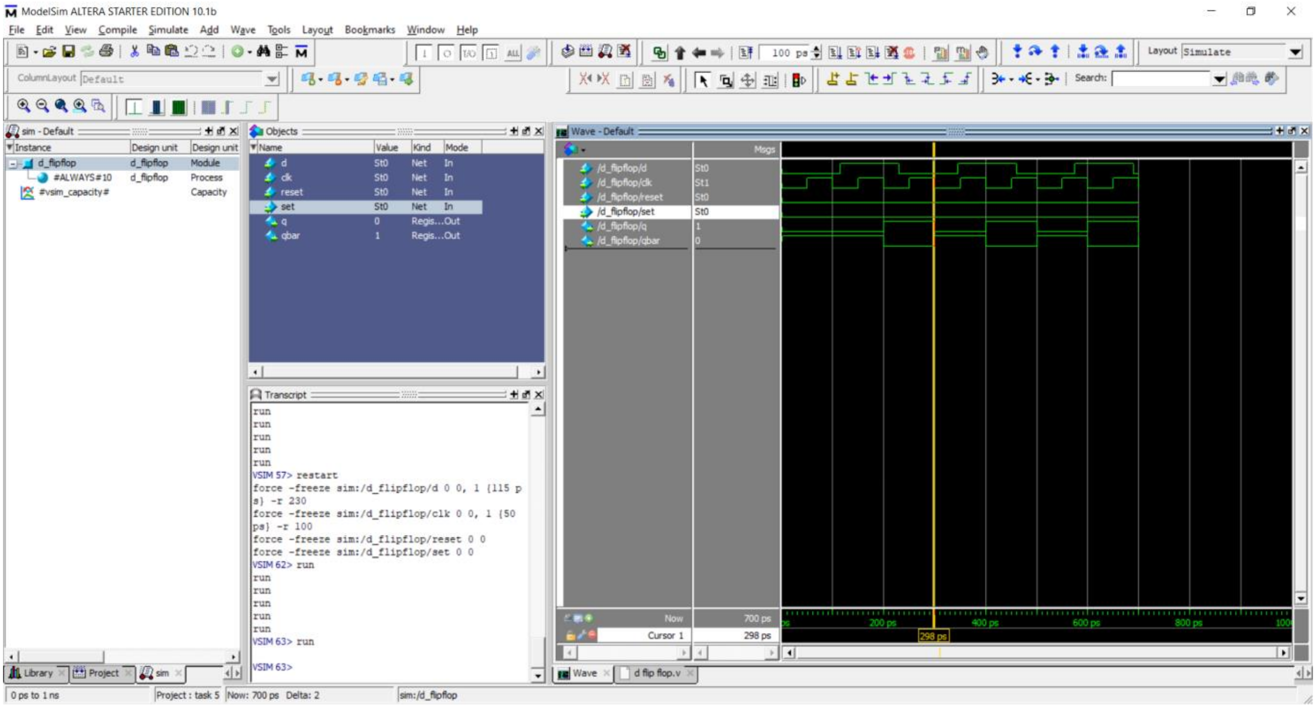
Name	Value	Kind	Mode
d	St0	Net	In
clk	St0	Net	In
reset	St0	Net	In
set	St0	Net	In
q	0	Regis...Out	
qbar	1	Regis...Out	

```
run
run
run
run
run
VSIM 57> restart
force -freeze sim:/d_flipflop/d 0 0, 1 {115 p
s} -r 230
force -freeze sim:/d_flipflop/clk 0 0, 1 {50
ps} -r 100
force -freeze sim:/d_flipflop/reset 0 0
force -freeze sim:/d_flipflop/set 0 0
VSIM 62> run
run
run
run
run
run
VSIM 63> run

VSIM 63>
```

C:/altera/12.1sp1/modelsim_ase/d flip flop.v (/d_flipflop) - Default

```
module d_flipflop(d,clk,q,qbar,reset,set);
input d,clk,reset,set;
output reg q,qbar;
initial
begin
q<=0;
qbar<=1;
end

always @(negedge clk,reset)
begin
  if(reset==1)
  begin
  q<=0;
  qbar<=1;
  end
  if(set==1)
  begin
  q<=1;
  qbar<=0;
  end
  if(d==1)
  begin
  q<=1;
  qbar<=0;
  end
  else if(d==0)
  begin
  q<=0;
  qbar<=1;
  end
end
endmodule
```

Ln: 27 Col: 7 Project : task 5 Now: 700 ps Delta: 2 sim:/d_flipflop



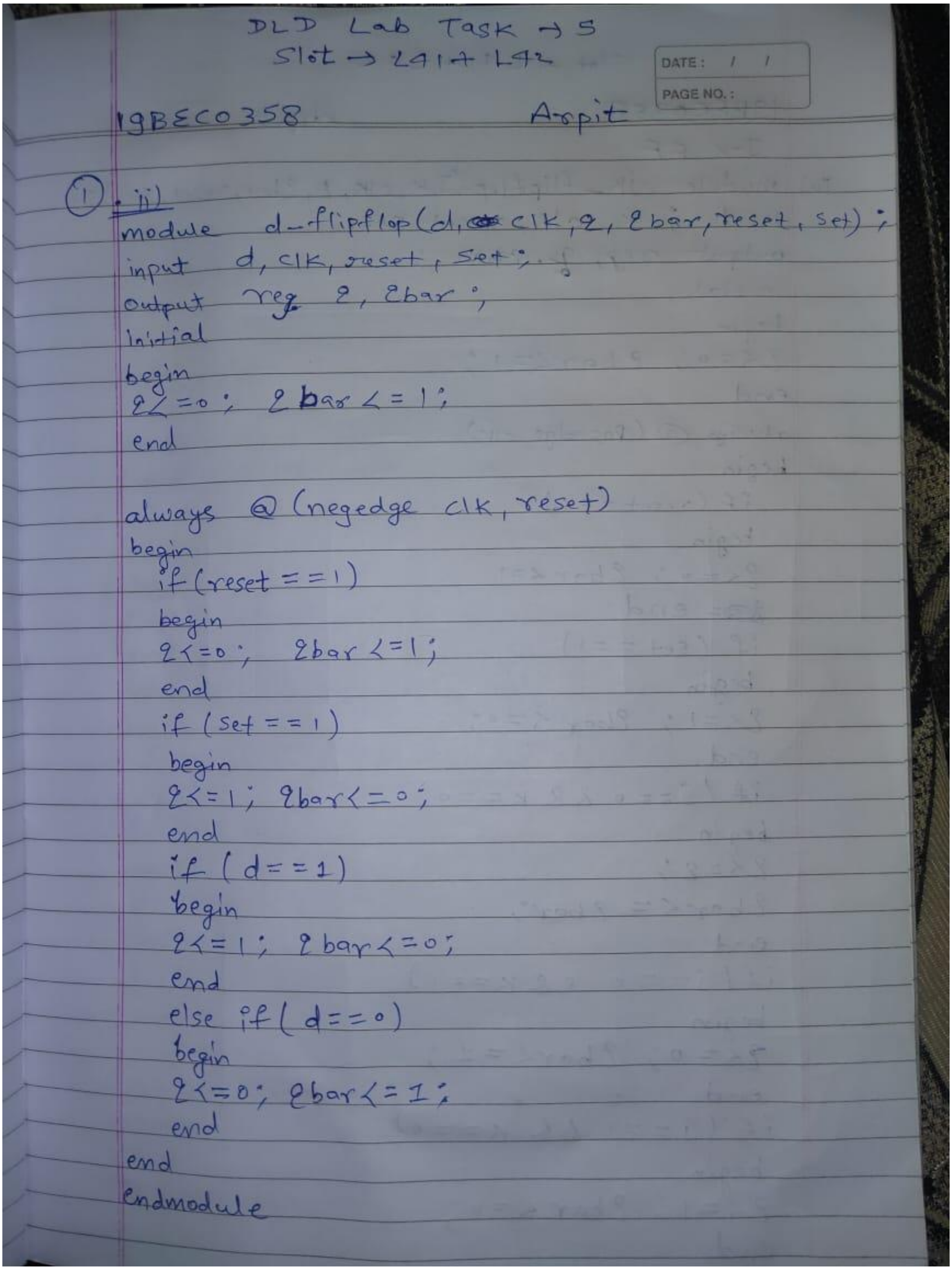
Instance	Design unit	Design unit
d_flipflop	d_flipflop	Module
#ALWAYS#10	d_flipflop	Process
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
d	St0	Net	In
clk	St0	Net	In
reset	St0	Net	In
set	St0	Net	In
q	0	Regis...Out	
qbar	1	Regis...Out	

```
run
run
run
run
run
VSIM 57> restart
force -freeze sim:/d_flipflop/d 0 0, 1 {115 p
s} -r 230
force -freeze sim:/d_flipflop/clk 0 0, 1 {50
ps} -r 100
force -freeze sim:/d_flipflop/reset 0 0
force -freeze sim:/d_flipflop/set 0 0
VSIM 62> run
run
run
run
run
run
VSIM 63> run

VSIM 63>
```

	Msgs
/d_flipflop/d	St0
/d_flipflop/clk	St1
/d_flipflop/reset	St0
/d_flipflop/set	St0
/d_flipflop/q	1
/d_flipflop/qbar	0
Now	700 ps
Cursor 1	298 ps

0 ps to 1 ns | Project : task 5 | Now: 700 ps Delta: 2 | sim:/d_flipflop



DLD Lab Task → 5
Slot → L41 + L42

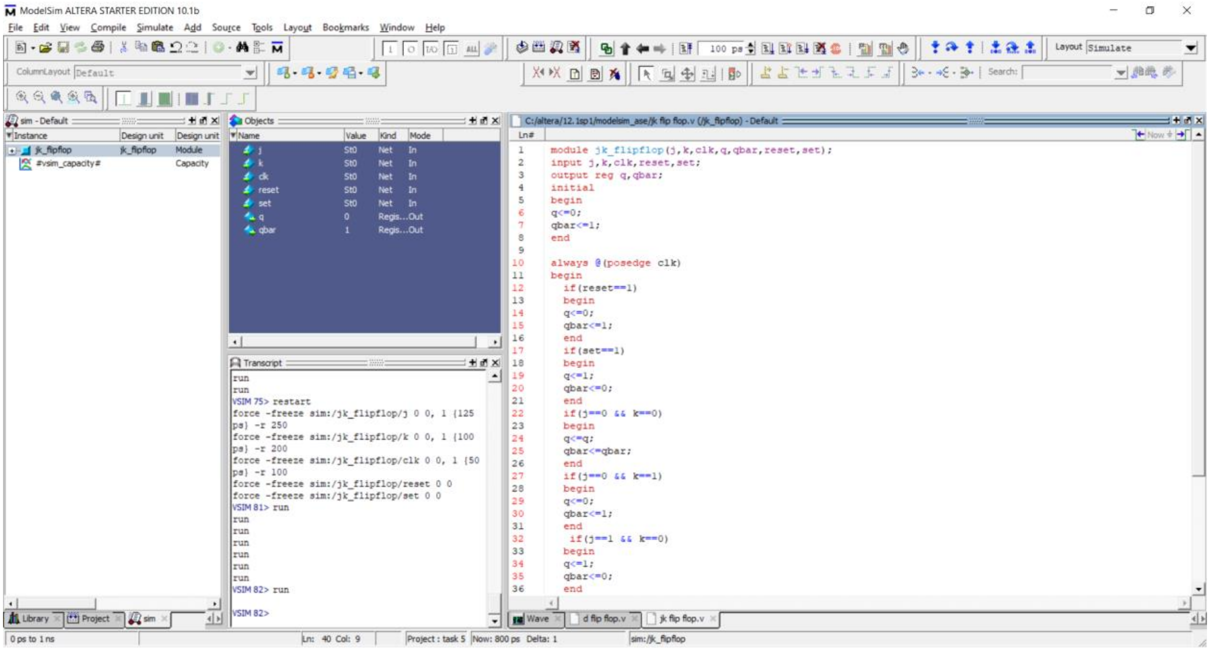
DATE: / /
PAGE NO.:

19BEC0358 Arpit

1 . ii)

```
module d_flipflop(d, clk, q, qbar, reset, set);
input d, clk, reset, set;
output reg q, qbar;
initial
begin
q<=0; qbar<=1;
end

always @(negedge clk, reset)
begin
  if(reset==1)
  begin
  q<=0;  qbar<=1;
  end
  if (set==1)
  begin
  q<=1; qbar<=0;
  end
  if (d==1)
  begin
  q<=1; qbar<=0;
  end
  else if(d==0)
  begin
  q<=0; qbar<=1;
  end
end
endmodule
```

iv) Write a Verilog program for **Synchronous Positive edge trigger "JK" flip-flop with Synchronous reset & synchronous set.**



ModelSim ALTERA STARTER EDITION 10.1b
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
sim - Default
Objects
Transcript
Wave
d flip flop.v
jk flip flop.v

Instance	Design unit	Design unit
jk_flipflop	jk_flipflop	Module
#vsim_capacity#		Capacity

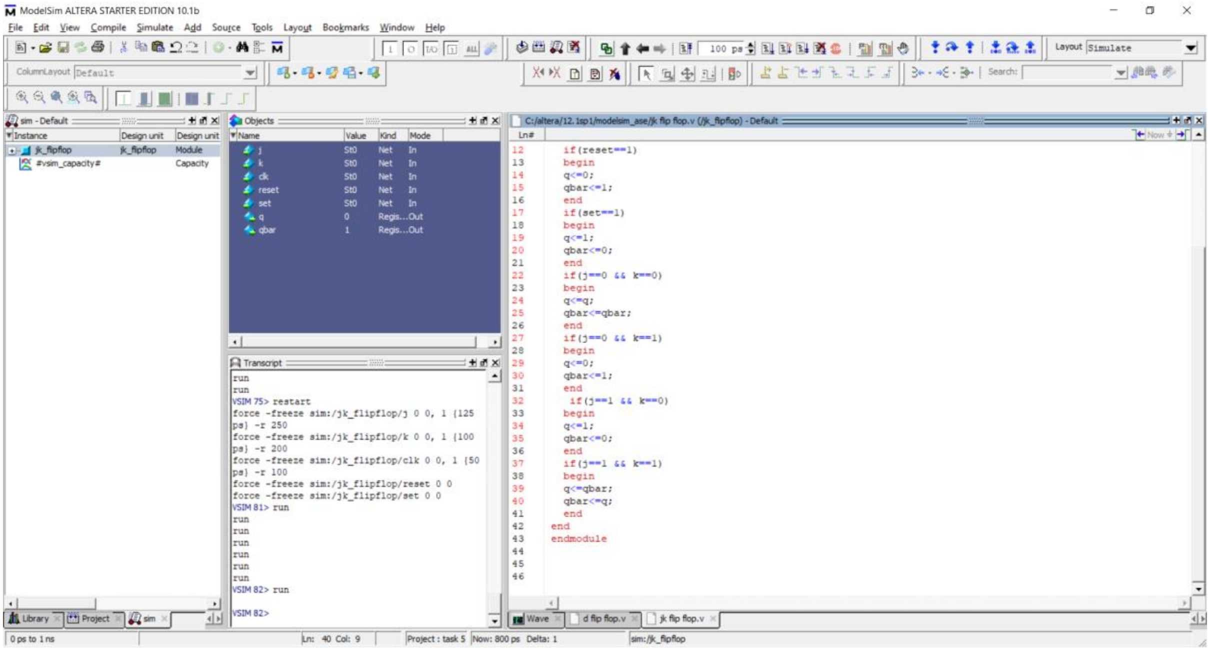
Name	Value	Kind	Mode
j	St0	Net	In
k	St0	Net	In
clk	St0	Net	In
reset	St0	Net	In
set	St0	Net	In
q	0	Regis...	Out
qbar	1	Regis...	Out

C:/altera/12.1sp1/modelsim_ase/jk flip flop.v (/jk_flipflop) - Default

```verilog
module jk_flipflop(j,k,clk,q,qbar,reset,set);
input j,k,clk,reset,set;
output reg q,qbar;
initial
begin
q<=0;
qbar<=1;
end

always @(posedge clk)
begin
  if(reset==1)
  begin
  q<=0;
  qbar<=1;
  end
  if(set==1)
  begin
  q<=1;
  qbar<=0;
  end
  if(j==0 && k==0)
  begin
  q<=q;
  qbar<=qbar;
  end
  if(j==0 && k==1)
  begin
  q<=0;
  qbar<=1;
  end
   if(j==1 && k==0)
  begin
  q<=1;
  qbar<=0;
  end
```

```
run
run
VSIM 75> restart
force -freeze sim:/jk_flipflop/j 0 0, 1 {125
ps} -r 250
force -freeze sim:/jk_flipflop/k 0 0, 1 {100
ps} -r 200
force -freeze sim:/jk_flipflop/clk 0 0, 1 {50
ps} -r 100
force -freeze sim:/jk_flipflop/reset 0 0
force -freeze sim:/jk_flipflop/set 0 0
VSIM 81> run
run
run
run
run
run
run
VSIM 82> run

VSIM 82>
```

0 ps to 1 ns | Ln: 40 Col: 9 | Project : task 5 | Now: 800 ps Delta: 1 | sim:/jk_flipflop

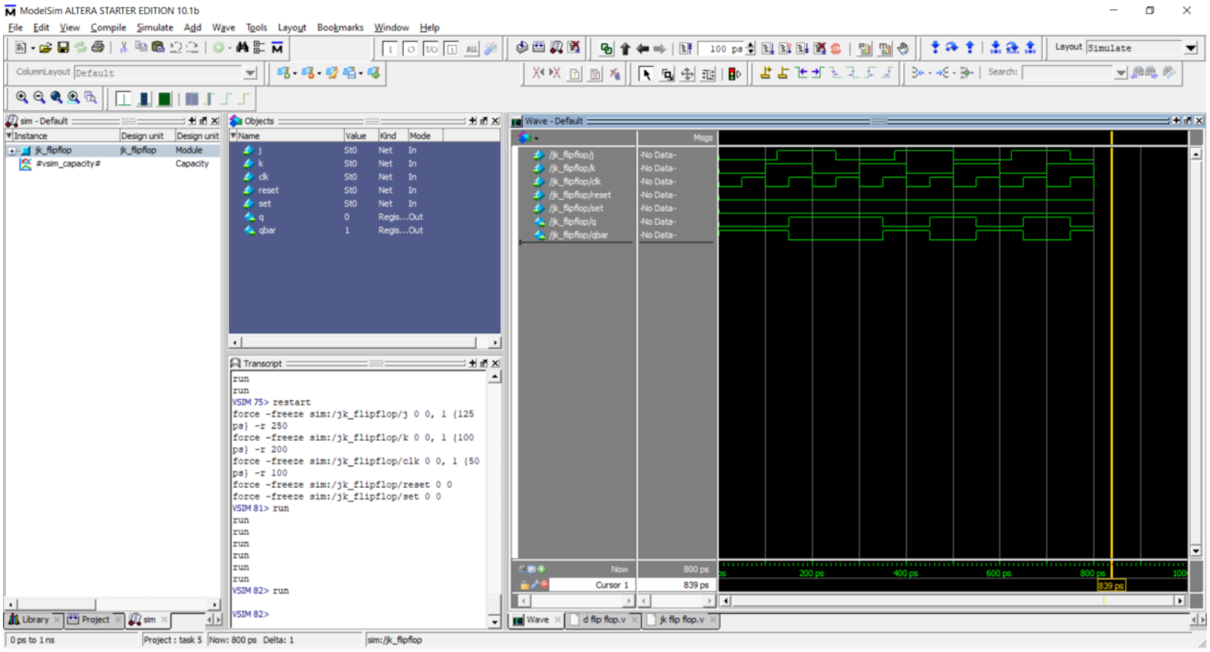

ModelSim ALTERA STARTER EDITION 10.1b
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
Layout Simulate
ColumnLayout Default
sim - Default
Objects
C:/altera/12.1sp1/modelsim_ase/jk flip flop.v (/jk_flipflop) - Default
Transcript
Library Project sim
Wave d flip flop.v jk flip flop.v
0 ps to 1 ns
Ln: 40 Col: 9
Project : task 5
Now: 800 ps Delta: 1
sim:/jk_flipflop

Instance	Design unit	Design unit
jk_flipflop	jk_flipflop	Module
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
j	St0	Net	In
k	St0	Net	In
clk	St0	Net	In
reset	St0	Net	In
set	St0	Net	In
q	0	Regis...	Out
qbar	1	Regis...	Out

```
run
run
VSIM 75> restart
force -freeze sim:/jk_flipflop/j 0 0, 1 {125
ps} -r 250
force -freeze sim:/jk_flipflop/k 0 0, 1 {100
ps} -r 200
force -freeze sim:/jk_flipflop/clk 0 0, 1 {50
ps} -r 100
force -freeze sim:/jk_flipflop/reset 0 0
force -freeze sim:/jk_flipflop/set 0 0
VSIM 81> run
run
run
run
run
run
run
VSIM 82> run

VSIM 82>
```

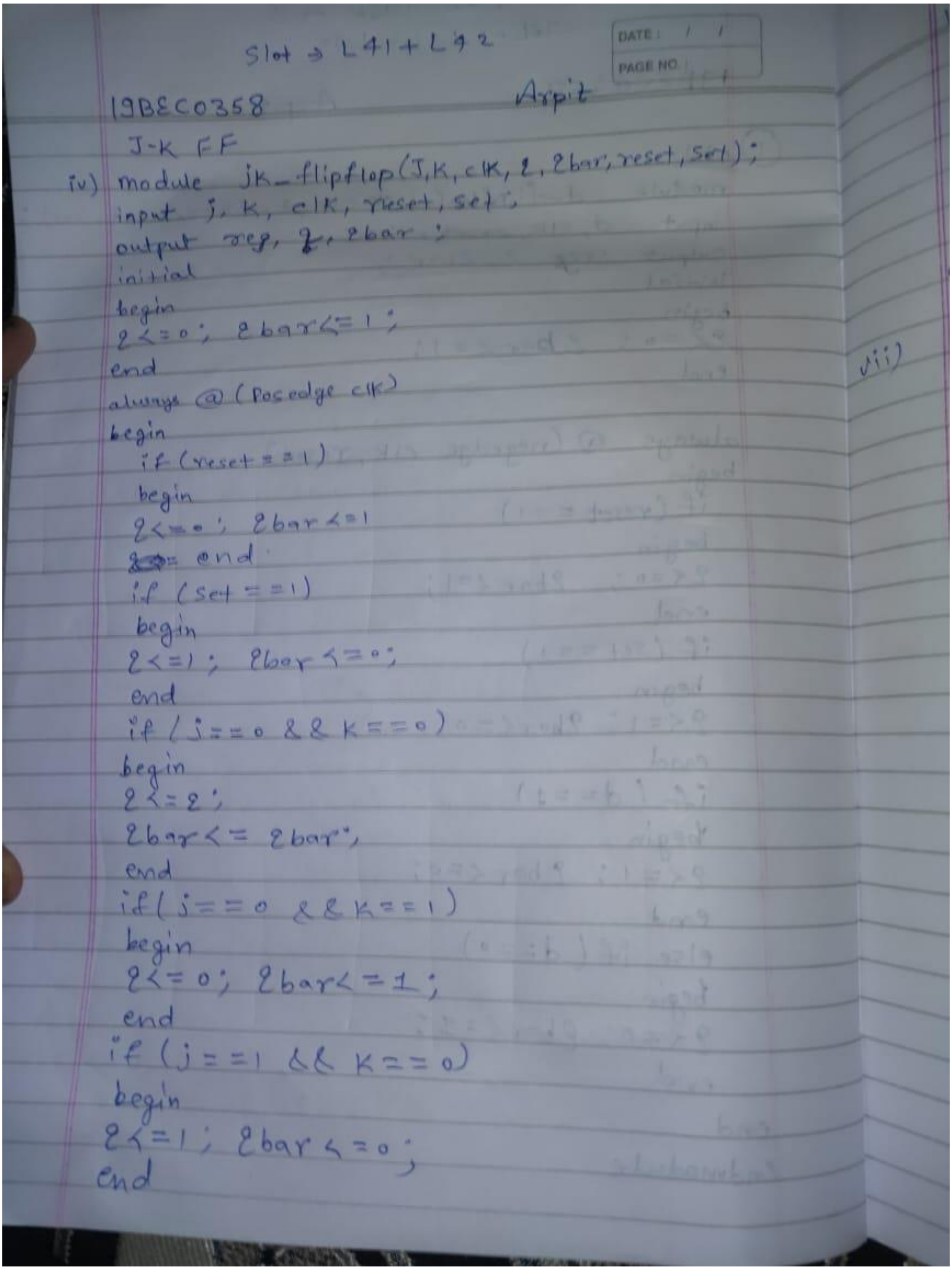
```
    if(reset==1)
    begin
    q<=0;
    qbar<=1;
    end
    if(set==1)
    begin
    q<=1;
    qbar<=0;
    end
    if(j==0 && k==0)
    begin
    q<=q;
    qbar<=qbar;
    end
    if(j==0 && k==1)
    begin
    q<=0;
    qbar<=1;
    end
     if(j==1 && k==0)
    begin
    q<=1;
    qbar<=0;
    end
    if(j==1 && k==1)
    begin
    q<=qbar;
    qbar<=q;
    end
  end
  endmodule
```



Instance	Design unit	Design unit
jk_flipflop	jk_flipflop	Module
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
j	St0	Net	In
k	St0	Net	In
clk	St0	Net	In
reset	St0	Net	In
set	St0	Net	In
q	0	Regis...	Out
qbar	1	Regis...	Out

```
run
run
VSIM 75> restart
force -freeze sim:/jk_flipflop/j 0 0, 1 {125
ps} -r 250
force -freeze sim:/jk_flipflop/k 0 0, 1 {100
ps} -r 200
force -freeze sim:/jk_flipflop/clk 0 0, 1 {50
ps} -r 100
force -freeze sim:/jk_flipflop/reset 0 0
force -freeze sim:/jk_flipflop/set 0 0
VSIM 81> run
run
run
run
run
run
run
VSIM 82> run

VSIM 82>
```



Slot → L41+L42

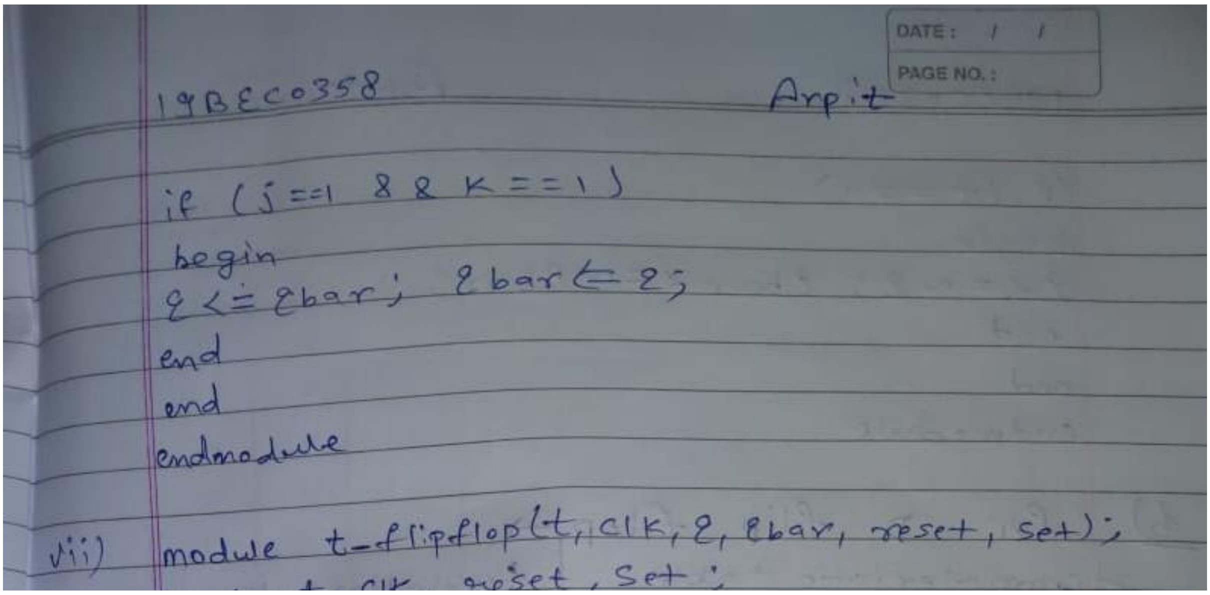
DATE: / /
PAGE NO.

19BEC0358 Arpit

J-K FF

iv)
```
module jk_flipflop(J,K,clk,q,qbar,reset,set);
input j,k,clk,reset,set;
output reg,q,qbar;
initial
begin
q<=0; qbar<=1;
end
always @(posedge clk)
begin
  if(reset==1)
  begin
  q<=0; qbar<=1
  end
  if (set==1)
  begin
  q<=1; qbar<=0;
  end
  if(j==0 && k==0)
  begin
  q<=q;
  qbar<=qbar;
  end
  if(j==0 && k==1)
  begin
  q<=0; qbar<=1;
  end
 if(j==1 && k==0)
 begin
q<=1; qbar<=0;
end
```

vii)



19BEC0358 Arpit

DATE : / /

PAGE NO. :

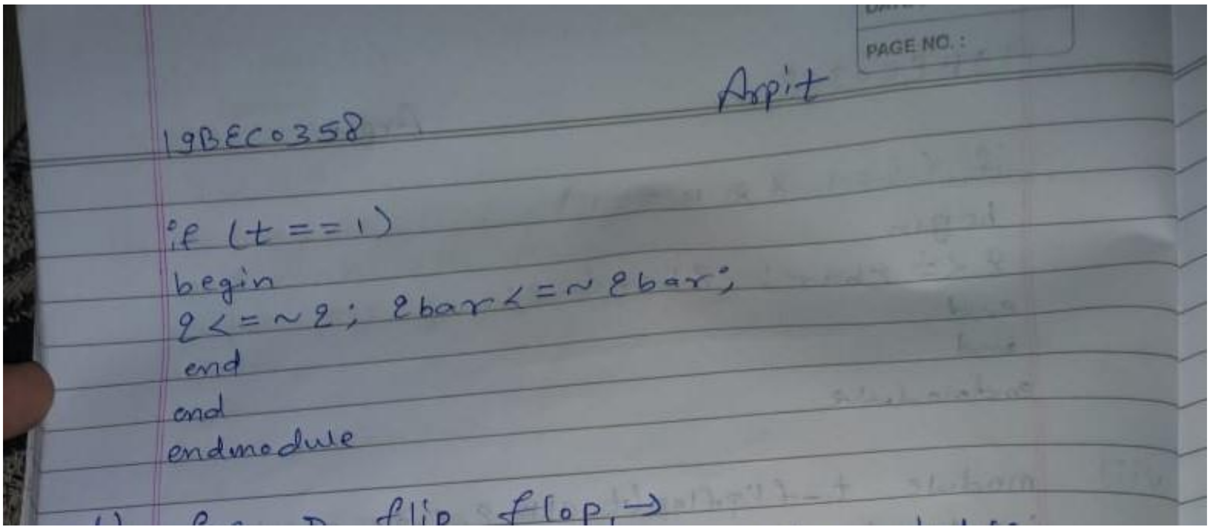
```
if (j==1 && k==1)
 begin
q <= qbar; qbar <= q;
end
end
endmodule

vii) module t_flipflop(t, clk, q, qbar, reset, set);
      t, clk, reset, Set ;
```

vii) Write a Verilog program for **Synchronous Positive edge trigger "T" flip-flop with Synchronous reset & set.**

```
if (j==1 && k==1)
begin
q<=qbar; qbar<=q;
end
end
endmodule
```

vii)

```
module t_flipflop(t, clk, q, qbar, reset, set);
input t, clk, reset, set;
output reg, q, qbar;
initial
begin
q<=0; qbar<=1;
end
always @(posedge clk)
begin
if (reset==1)
begin
q<=0; qbar<=1;
end
if (set==1)
begin
q<=1; qbar<=0;
end
if (t==0)
begin
q<=q; qbar<=qbar;
end
~~if (t==1)~~
```



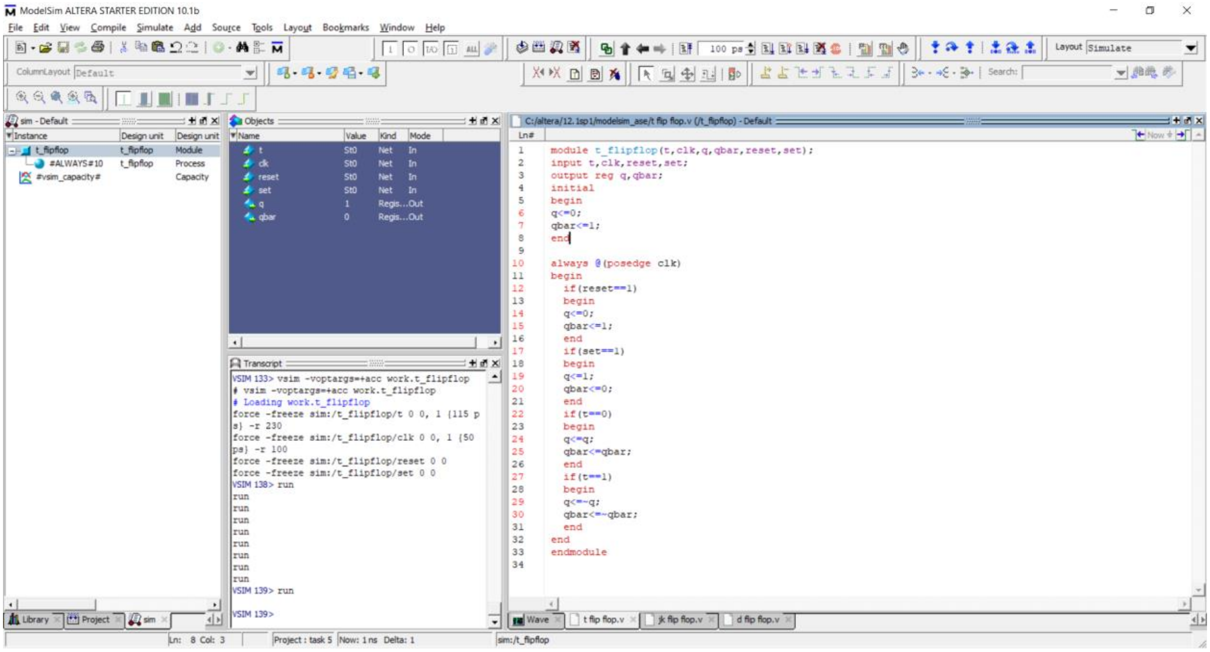
PAGE NO. :

19BEC0358

Arpit

```
if (t==1)
begin
q<=~q; qbar<=~qbar;
end
end
endmodule
```

D flip flop →

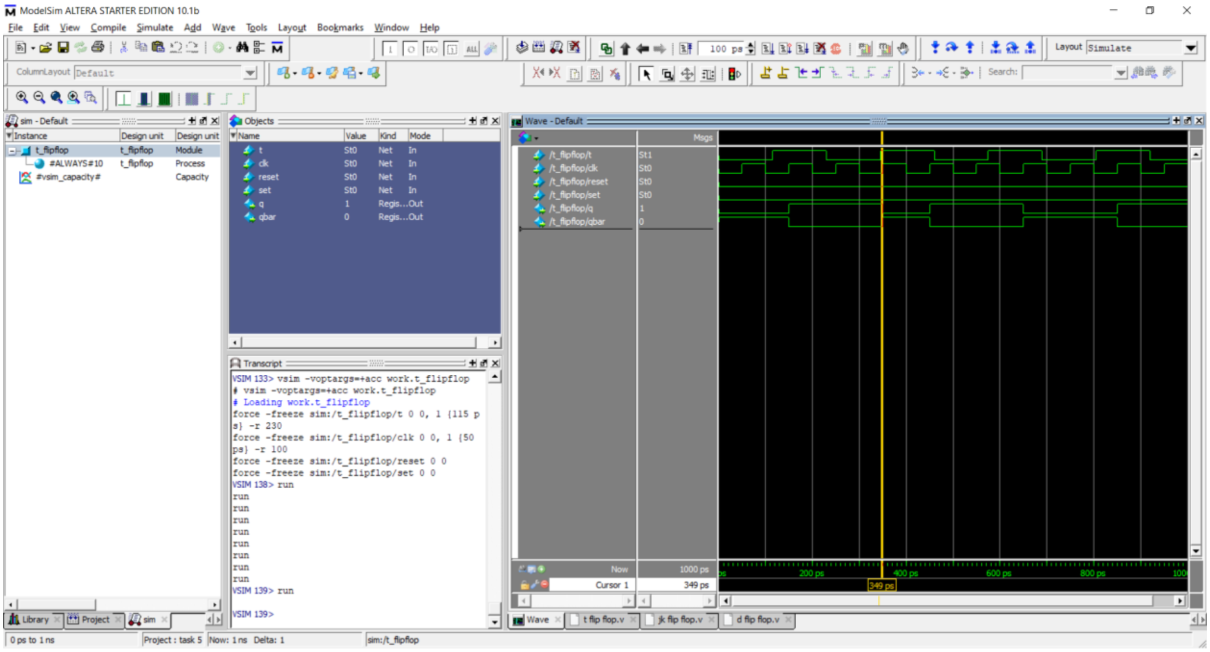


Instance	Design unit	Design unit
t_flipflop	t_flipflop	Module
#ALWAYS#10	t_flipflop	Process
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
t	St0	Net	In
clk	St0	Net	In
reset	St0	Net	In
set	St0	Net	In
q	1	Regis...	Out
qbar	0	Regis...	Out

```
module t_flipflop(t,clk,q,qbar,reset,set);
input t,clk,reset,set;
output reg q,qbar;
initial
begin
q<=0;
qbar<=1;
end

always @(posedge clk)
begin
  if(reset==1)
  begin
  q<=0;
  qbar<=1;
  end
  if(set==1)
  begin
  q<=1;
  qbar<=0;
  end
  if(t==0)
  begin
  q<=q;
  qbar<=qbar;
  end
  if(t==1)
  begin
  q<=~q;
  qbar<=~qbar;
  end
end
endmodule
```

```
VSIM 133> vsim -voptargs=+acc work.t_flipflop
# vsim -voptargs=+acc work.t_flipflop
# Loading work.t_flipflop
force -freeze sim:/t_flipflop/t 0 0, 1 {115 p
s} -r 230
force -freeze sim:/t_flipflop/clk 0 0, 1 {50
ps} -r 100
force -freeze sim:/t_flipflop/reset 0 0
force -freeze sim:/t_flipflop/set 0 0
VSIM 138> run
run
run
run
run
run
run
run
run
VSIM 139> run

VSIM 139>
```

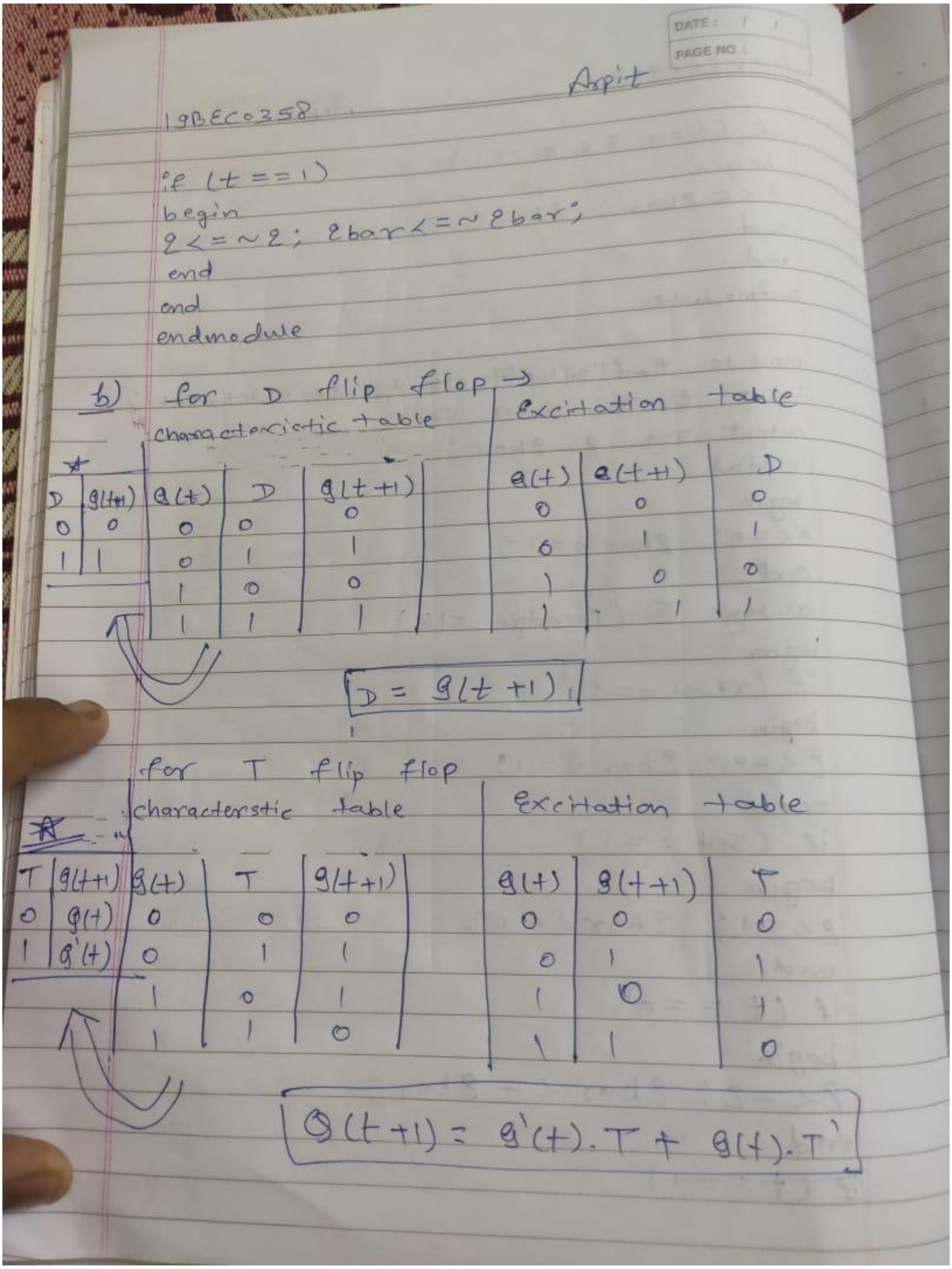


Instance	Design unit	Design unit
t_flipflop	t_flipflop	Module
#ALWAYS#10	t_flipflop	Process
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
t	St0	Net	In
clk	St0	Net	In
reset	St0	Net	In
set	St0	Net	In
q	1	Regis...	Out
qbar	0	Regis...	Out

```
VSIM 133> vsim -voptargs=+acc work.t_flipflop
# vsim -voptargs=+acc work.t_flipflop
# Loading work.t_flipflop
force -freeze sim:/t_flipflop/t 0 0, 1 {115 p
s} -r 230
force -freeze sim:/t_flipflop/clk 0 0, 1 {50
ps} -r 100
force -freeze sim:/t_flipflop/reset 0 0
force -freeze sim:/t_flipflop/set 0 0
VSIM 138> run
run
run
run
run
run
run
run
run
VSIM 139> run

VSIM 139>
```

b) Write the Characteristic table and Excitation table for “D”, “JK” and “T”



DATE: / /

PAGE NO.:

Arpit

19BEC0358

```
if (t==1)
begin
q<=~q; qbar<=~qbar;
end
end
endmodule
```

b) for D flip flop →

characteristic table

D	q(t+1)
0	0
1	1

q(t)	D	q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Excitation table

q(t)	q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

$$\boxed{D = q(t+1)}$$

for T flip flop

characterstic table

T	q(t+1)
0	q(t)
1	q'(t)

q(t)	T	q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Excitation table

q(t)	q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

$$\boxed{Q(t+1) = q'(t).T + q(t).T'}$$

19BEC0358 Arpit

for JK flip-flop =>

~~characterstic table~~

Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

characterstic table

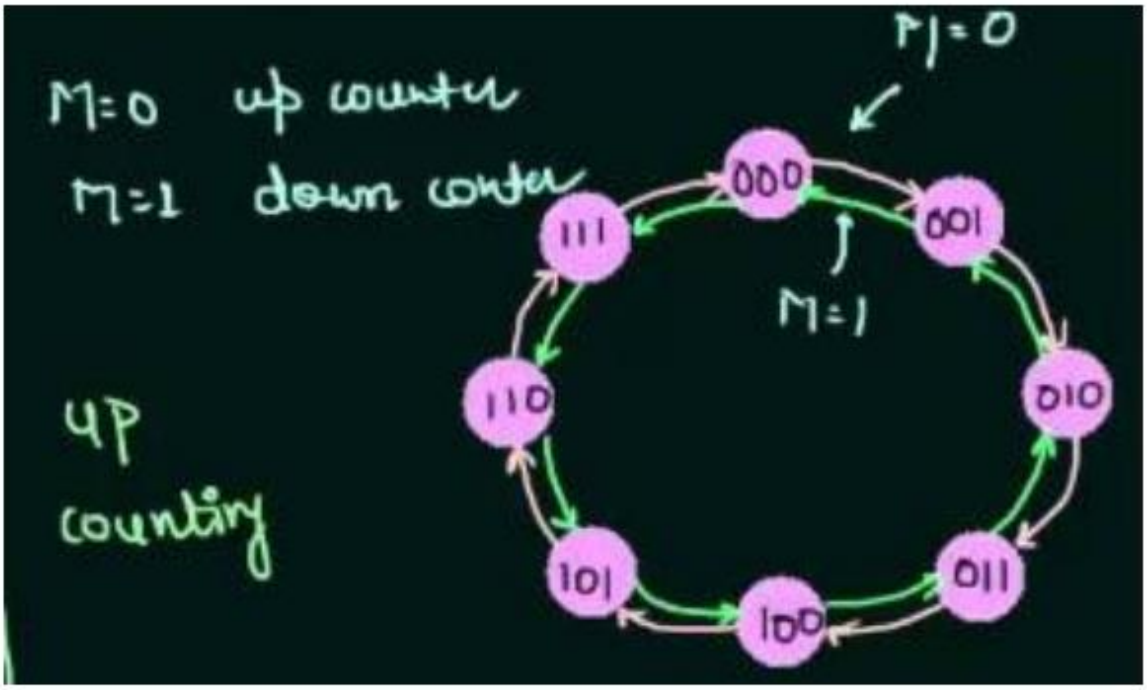
J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q'(t)

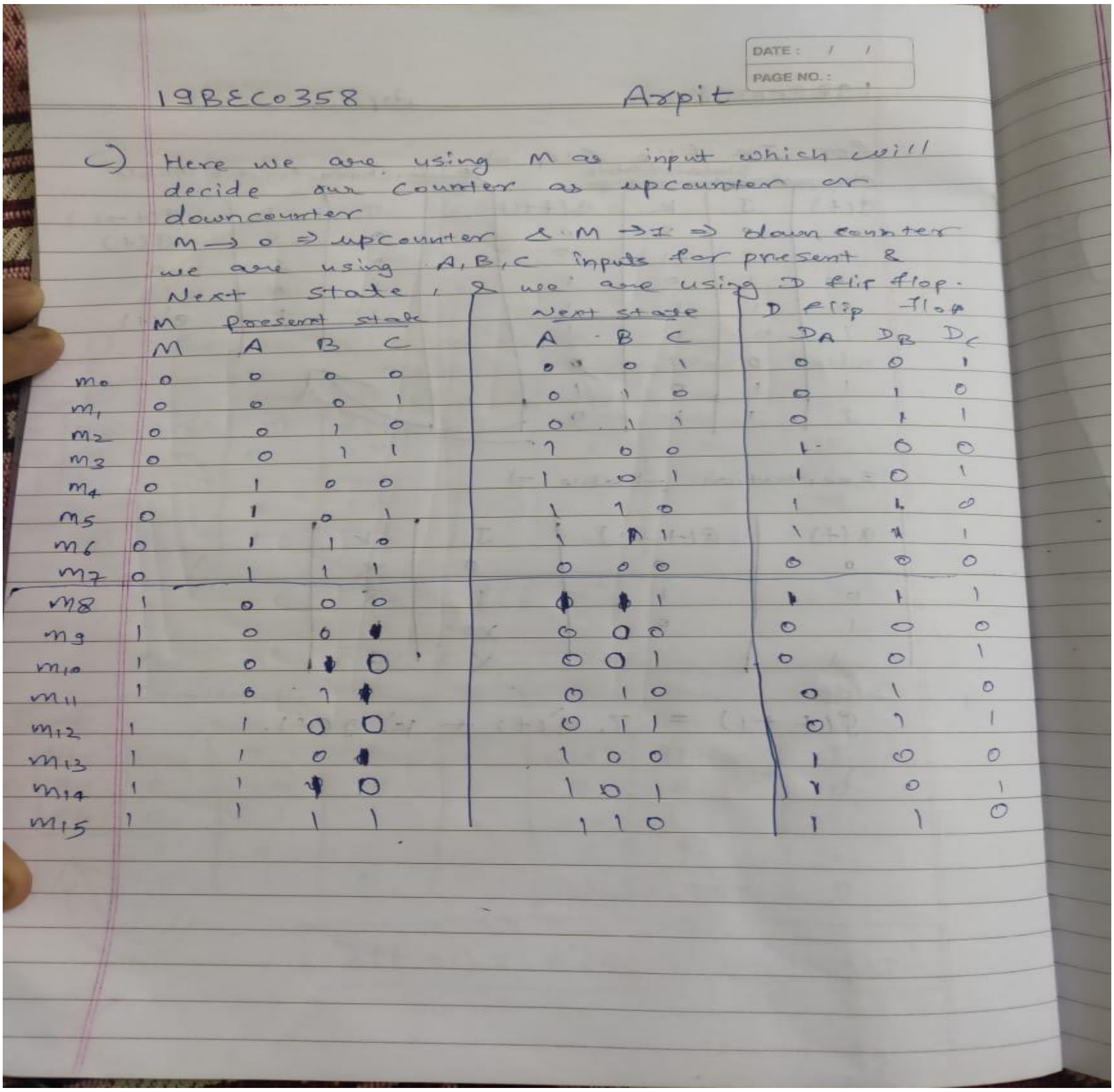
Excitation table =>

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$$Q(t+1) = J.\,Q'(t) + K'.\,Q(t).$$

c) Write a Verilog code for 3 bit up/down counter





DATE : / /

PAGE NO. :

19BEC0358 Arpit

c) Here we are using M as input which will decide our counter as upcounter or downcounter

M → 0 ⇒ upcounter & M → 1 ⇒ down counter

we are using A, B, C inputs for present & Next state, & we are using D flip flop.

	M	Present state			Next state			D flip flop		
	M	A	B	C	A	B	C	D_A	D_B	D_C
m_0	0	0	0	0	0	0	1	0	0	1
m_1	0	0	0	1	0	1	0	0	1	0
m_2	0	0	1	0	0	1	1	0	1	1
m_3	0	0	1	1	1	0	0	1	0	0
m_4	0	1	0	0	1	0	1	1	0	1
m_5	0	1	0	1	1	1	0	1	1	0
m_6	0	1	1	0	1	1	1	1	1	1
m_7	0	1	1	1	0	0	0	0	0	0
m_8	1	0	0	0	1	1	1	1	1	1
m_9	1	0	0	1	0	0	0	0	0	0
m_{10}	1	0	1	0	0	0	1	0	0	1
m_{11}	1	0	1	1	0	1	0	0	1	0
m_{12}	1	1	0	0	0	1	1	0	1	1
m_{13}	1	1	0	1	1	0	0	1	0	0
m_{14}	1	1	1	0	1	0	1	1	0	1
m_{15}	1	1	1	1	1	1	0	1	1	0

19BEC0358 Arpit

DATE: / /
PAGE NO.:

K-map for $D_A \rightarrow D_A = \Sigma m(3, 4, 5, 6, 8, 13, 14, 15)$

MA \ BC	00	01	11	10
00	$_{0}$	$_{1}$	1_{3}	$_{2}$
01	1_{4}	1_{5}	$_{7}$	1_{6}
11	$_{12}$	1_{13}	1_{15}	1_{14}
10	1_{8}	$_{9}$	$_{11}$	$_{10}$

$D_A = M'AB' + MAC + ABC' +$
$+ M'A'BC + MA'B'C'$

Kmap for $D_B \rightarrow$

$D_B = \Sigma m(1, 2, 5, 6, 8, 11, 12, 15)$

MA \ BC	00	01	11	10
00	$_{0}$	1_{1}	$_{3}$	1_{2}
01	$_{4}$	1_{5}	$_{7}$	1_{6}
11	1_{12}	$_{13}$	1_{15}	$_{14}$
10	1_{8}	$_{9}$	1_{11}	$_{10}$

$$D_B = MB'C' + M'B'C + MBC + M'BC'$$
$$= M(BC + B'C') + M'(BC' + B'C)$$
$$= M(\overline{B \oplus C}) + M'(B \oplus C)$$
$$= M \oplus B \oplus C$$

Kmap for $D_C \rightarrow$

$D_C = \Sigma m(0, 2, 4, 6, 8, 10, 12, 14)$

MA \ BC	00	01	11	10
00	1_{0}	$_{1}$	$_{3}$	1_{2}
01	1_{4}	$_{5}$	$_{7}$	1_{6}
11	1_{12}	$_{13}$	$_{15}$	1_{14}
10	1_{8}	$_{9}$	$_{11}$	1_{10}

$D_C = C'$

DATE : / /

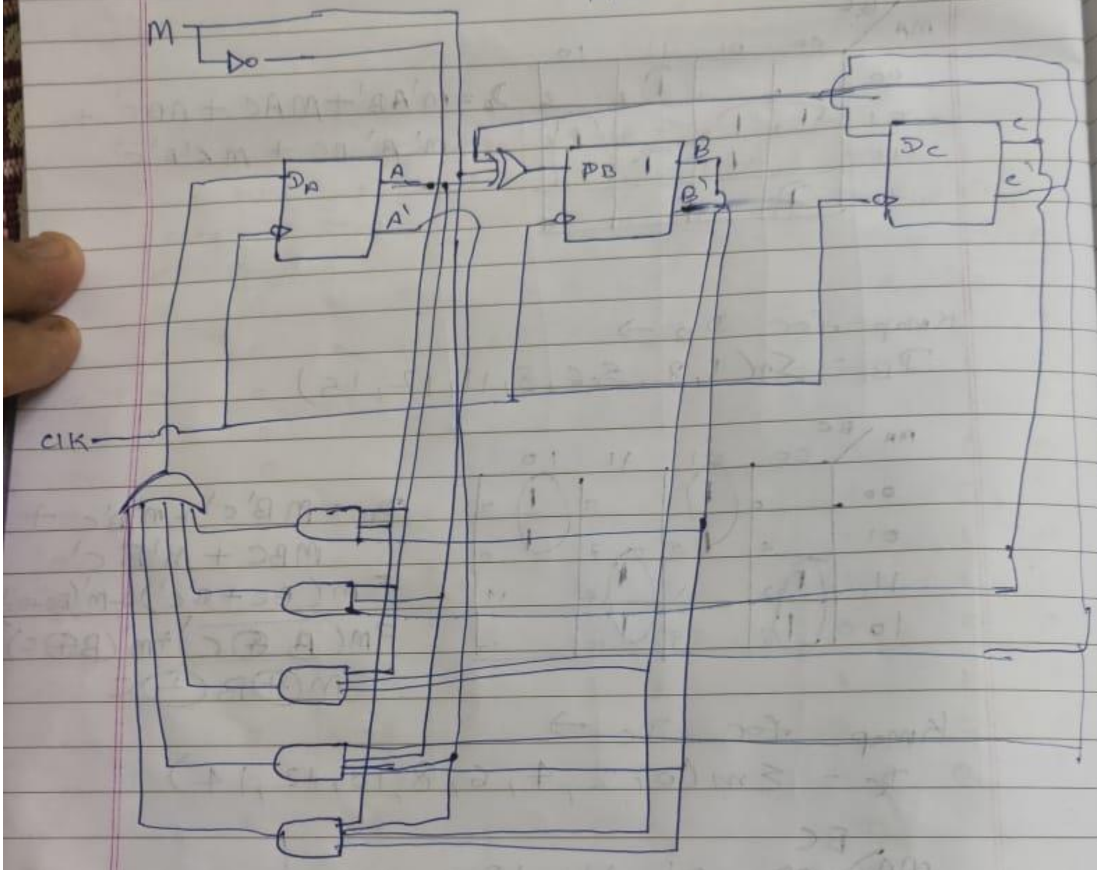
PAGE NO. :

Arpit

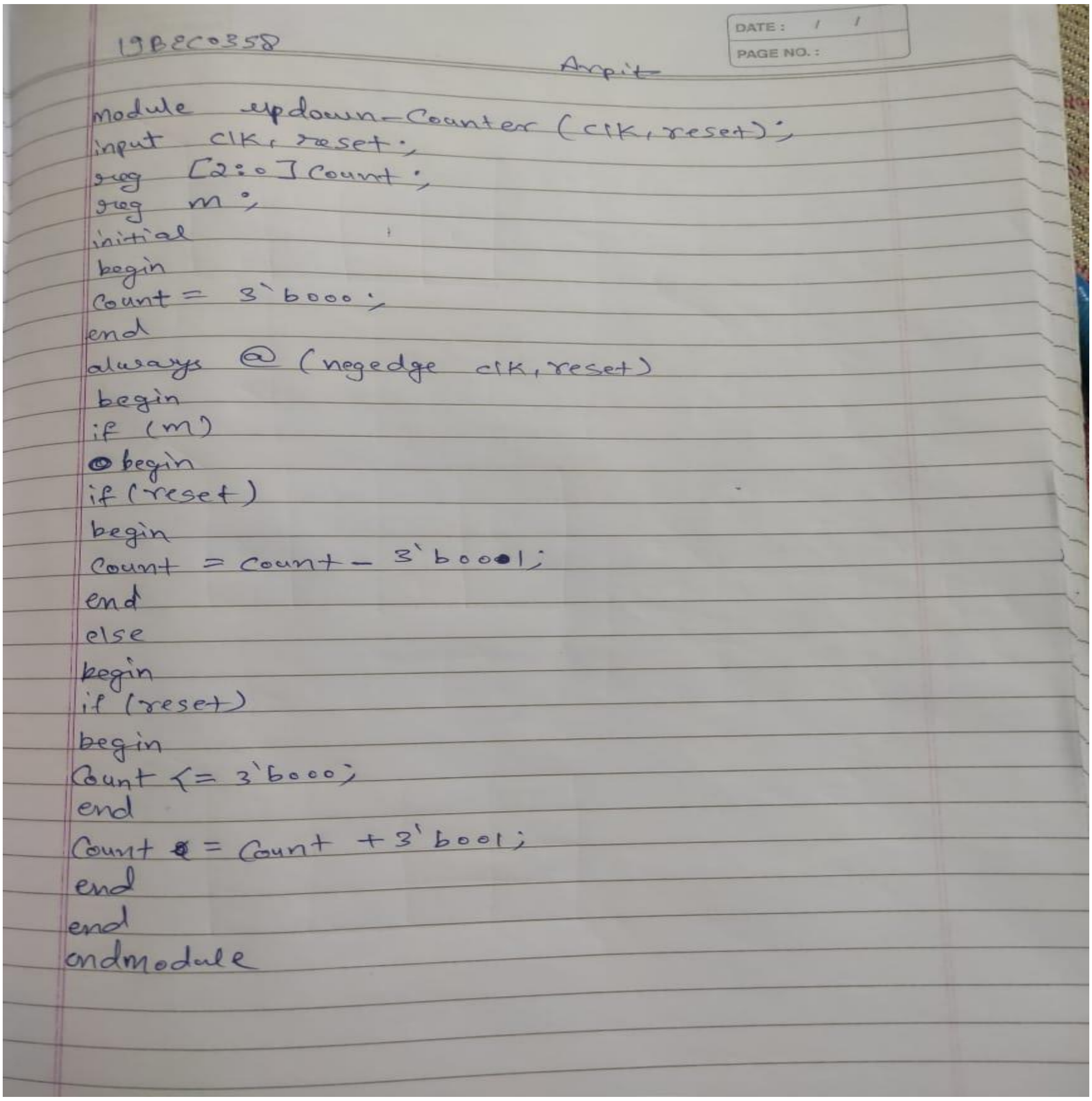
19BEC0358

M=0 → up Counter

M=1 → down Counter



3 bit up/down Counter using D FF



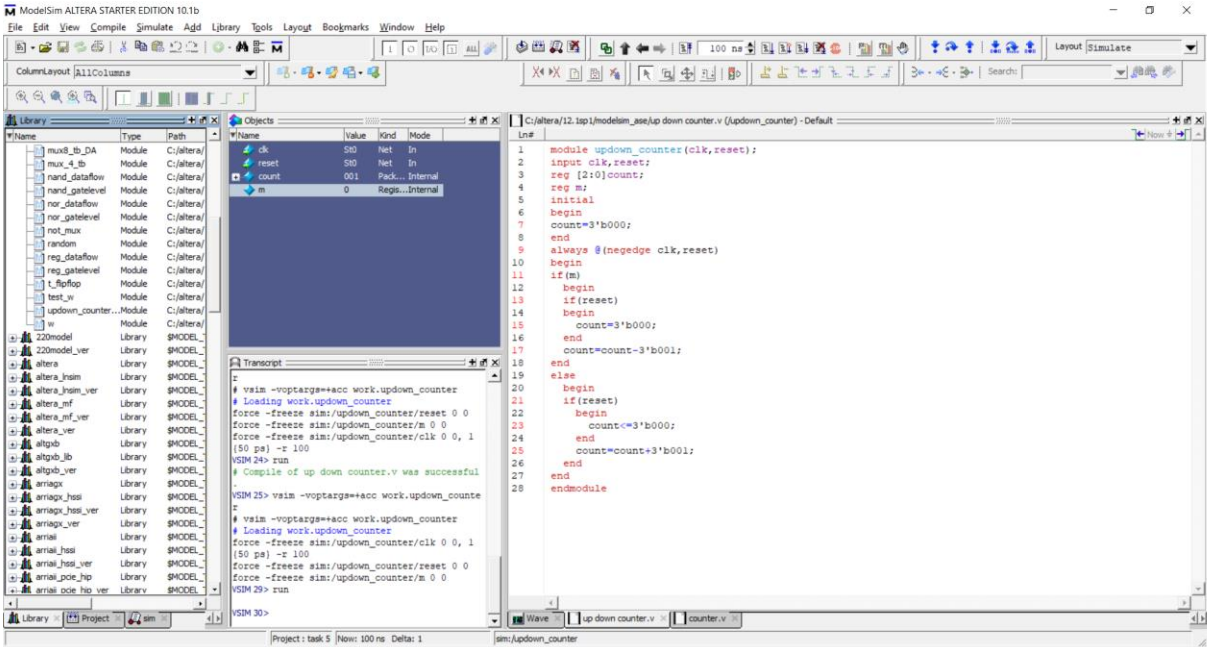
19BEC0358

DATE : / /

PAGE NO. :

Arpit

```
module updown_Counter (clk, reset);
input clk, reset;
reg [2:0] count;
reg m;
initial
begin
count = 3`b000;
end
always @ (negedge clk, reset)
begin
if (m)
begin
if (reset)
begin
count = count - 3`b001;
end
else
begin
if (reset)
begin
Count <= 3`b000;
end
Count = Count + 3`b001;
end
end
endmodule
```



Name	Type	Path
mux8_tb_DA	Module	C:/altera/
mux_4_tb	Module	C:/altera/
nand_dataflow	Module	C:/altera/
nand_gatelevel	Module	C:/altera/
nor_dataflow	Module	C:/altera/
nor_gatelevel	Module	C:/altera/
not_mux	Module	C:/altera/
random	Module	C:/altera/
reg_dataflow	Module	C:/altera/
reg_gatelevel	Module	C:/altera/
t_flipflop	Module	C:/altera/
test_w	Module	C:/altera/
updown_counter...	Module	C:/altera/
w	Module	C:/altera/
220model	Library	$MODEL_
220model_ver	Library	$MODEL_
altera	Library	$MODEL_
altera_lnsim	Library	$MODEL_
altera_lnsim_ver	Library	$MODEL_
altera_mf	Library	$MODEL_
altera_mf_ver	Library	$MODEL_
altera_ver	Library	$MODEL_
altgxb	Library	$MODEL_
altgxb_lib	Library	$MODEL_
altgxb_ver	Library	$MODEL_
arriagx	Library	$MODEL_
arriagx_hssi	Library	$MODEL_
arriagx_hssi_ver	Library	$MODEL_
arriagx_ver	Library	$MODEL_
arriaii	Library	$MODEL_
arriaii_hssi	Library	$MODEL_
arriaii_hssi_ver	Library	$MODEL_
arriaii_pcie_hip	Library	$MODEL_
arriaii_pcie_hip_ver	Library	$MODEL_

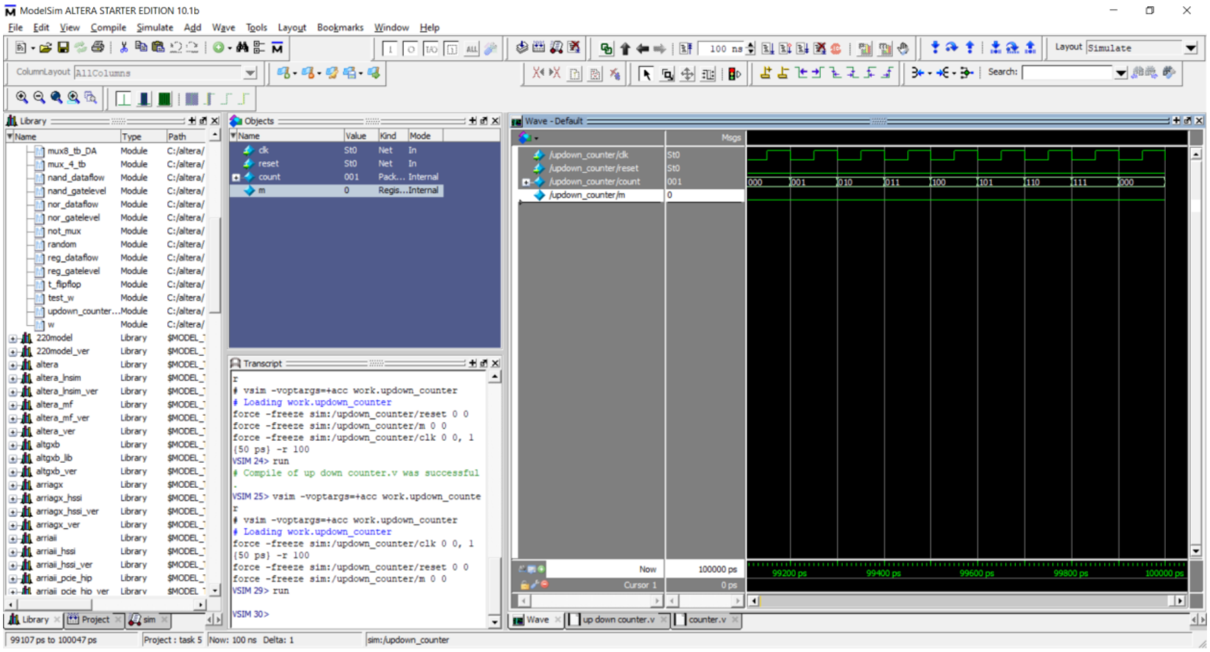
Name	Value	Kind	Mode
clk	St0	Net	In
reset	St0	Net	In
count	001	Pack...	Internal
m	0	Regis...	Internal

C:/altera/12.1sp1/modelsim_ase/up down counter.v (/updown_counter) - Default

```
module updown_counter(clk,reset);
input clk,reset;
reg [2:0]count;
reg m;
initial
begin
count=3'b000;
end
always @(negedge clk,reset)
begin
if(m)
  begin
  if(reset)
  begin
    count=3'b000;
  end
  count=count-3'b001;
end
else
  begin
  if(reset)
    begin
      count<=3'b000;
    end
    count=count+3'b001;
  end
end
endmodule
```

```
r
# vsim -voptargs=+acc work.updown_counter
# Loading work.updown_counter
force -freeze sim:/updown_counter/reset 0 0
force -freeze sim:/updown_counter/m 0 0
force -freeze sim:/updown_counter/clk 0 0, 1
{50 ps} -r 100
VSIM 24> run
# Compile of up down counter.v was successful
.
VSIM 25> vsim -voptargs=+acc work.updown_counte
r
# vsim -voptargs=+acc work.updown_counter
# Loading work.updown_counter
force -freeze sim:/updown_counter/clk 0 0, 1
{50 ps} -r 100
force -freeze sim:/updown_counter/reset 0 0
force -freeze sim:/updown_counter/m 0 0
VSIM 29> run

VSIM 30>
```

Project : task 5 Now: 100 ns Delta: 1 sim:/updown_counter



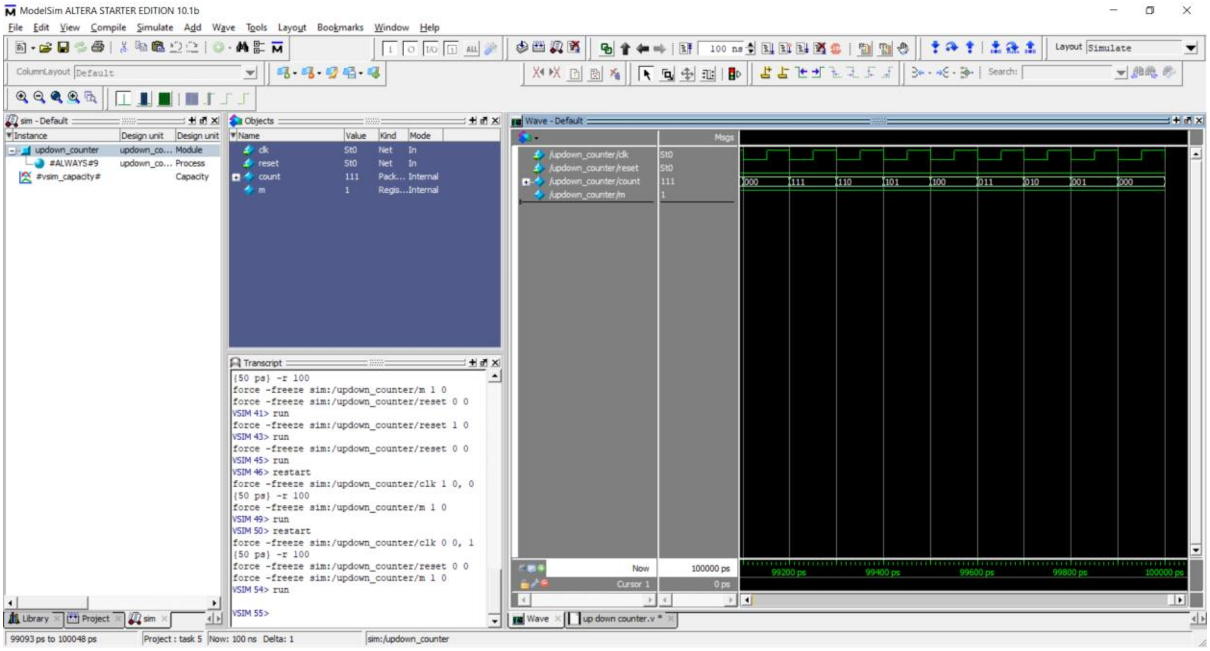
Name	Type	Path
mux8_tb_DA	Module	C:/altera/
mux_4_tb	Module	C:/altera/
nand_dataflow	Module	C:/altera/
nand_gatelevel	Module	C:/altera/
nor_dataflow	Module	C:/altera/
nor_gatelevel	Module	C:/altera/
not_mux	Module	C:/altera/
random	Module	C:/altera/
reg_dataflow	Module	C:/altera/
reg_gatelevel	Module	C:/altera/
t_flipflop	Module	C:/altera/
test_w	Module	C:/altera/
updown_counter...	Module	C:/altera/
w	Module	C:/altera/
220model	Library	$MODEL_
220model_ver	Library	$MODEL_
altera	Library	$MODEL_
altera_lnsim	Library	$MODEL_
altera_lnsim_ver	Library	$MODEL_
altera_mf	Library	$MODEL_
altera_mf_ver	Library	$MODEL_
altera_ver	Library	$MODEL_
altgxb	Library	$MODEL_
altgxb_lib	Library	$MODEL_
altgxb_ver	Library	$MODEL_
arriagx	Library	$MODEL_
arriagx_hssi	Library	$MODEL_
arriagx_hssi_ver	Library	$MODEL_
arriagx_ver	Library	$MODEL_
arriaii	Library	$MODEL_
arriaii_hssi	Library	$MODEL_
arriaii_hssi_ver	Library	$MODEL_
arriaii_pcie_hip	Library	$MODEL_
arriaii_pcie_hip_ver	Library	$MODEL_

Name	Value	Kind	Mode
clk	St0	Net	In
reset	St0	Net	In
count	001	Pack...	Internal
m	0	Regis...	Internal

```
r
# vsim -voptargs=+acc work.updown_counter
# Loading work.updown_counter
force -freeze sim:/updown_counter/reset 0 0
force -freeze sim:/updown_counter/m 0 0
force -freeze sim:/updown_counter/clk 0 0, 1
{50 ps} -r 100
VSIM 24> run
# Compile of up down counter.v was successful
.
VSIM 25> vsim -voptargs=+acc work.updown_counte
r
# vsim -voptargs=+acc work.updown_counter
# Loading work.updown_counter
force -freeze sim:/updown_counter/clk 0 0, 1
{50 ps} -r 100
force -freeze sim:/updown_counter/reset 0 0
force -freeze sim:/updown_counter/m 0 0
VSIM 29> run

VSIM 30>
```

Signal	Msgs
/updown_counter/clk	St0
/updown_counter/reset	St0
/updown_counter/count	001
/updown_counter/m	0

99107 ps to 100047 ps Project : task 5 Now: 100 ns Delta: 1 sim:/updown_counter



Instance	Design unit	Design unit
updown_counter	updown_co...	Module
#ALWAYS#9	updown_co...	Process
#vsim_capacity#		Capacity

Name	Value	Kind	Mode
clk	St0	Net	In
reset	St0	Net	In
count	111	Pack...	Internal
m	1	Regis...	Internal

	Msgs
/updown_counter/clk	St0
/updown_counter/reset	St0
/updown_counter/count	111
/updown_counter/m	1

Now 100000 ps

Cursor 1 0 ps

```
{50 ps} -r 100
force -freeze sim:/updown_counter/m 1 0
force -freeze sim:/updown_counter/reset 0 0
VSIM 41> run
force -freeze sim:/updown_counter/reset 1 0
VSIM 43> run
force -freeze sim:/updown_counter/reset 0 0
VSIM 45> run
VSIM 46> restart
force -freeze sim:/updown_counter/clk 1 0, 0
{50 ps} -r 100
force -freeze sim:/updown_counter/m 1 0
VSIM 49> run
VSIM 50> restart
force -freeze sim:/updown_counter/clk 0 0, 1
{50 ps} -r 100
force -freeze sim:/updown_counter/reset 0 0
force -freeze sim:/updown_counter/m 1 0
VSIM 54> run

VSIM 55>
```

99093 ps to 100048 ps | Project : task 5 | Now: 100 ns Delta: 1 | sim:/updown_counter