

3 A

```
module async_dlatch(input d,clk,pre,clr,output q,qc);
  wire notd,s,r;
  not(notd,d);
  nand(s,d,clk);
  nand(r,notd,clk);
  nand(q,s,pre,qc);
  nand(qc,r,clr,q);
endmodule
```

```
module async_dlatch_tb;
  reg d,clk,pre,clr;
  wire q,qc;
  async_dlatch gate3(d,clk,pre,clr,q,qc);
  initial begin
    pre=0;clr=0;clk=0;d=0;
    #10 pre=0;clr=1;clk=0;d=0;
    #10 pre=1;clr=0;clk=0;d=0;
    #10 pre=1;clr=1;clk=0;d=0;
    #10 pre=1;clr=1;clk=0;d=1;
    #10 pre=1;clr=1;clk=1;d=0;
    #10 pre=1;clr=1;clk=0;d=0;
    #10 pre=1;clr=1;clk=1;d=1;
    #10 pre=1;clr=1;clk=0;d=0;
    #10 pre=1;clr=1;clk=1;d=0;
    #10 pre=1;clr=1;clk=0;d=1;
  end
endmodule
```

3 B

```
//D latch synchronous preset clear
module dlatch_sync(input d, clk, pre, clr, output reg q);
  always @(d, clk) begin
    if(clk) begin
      if(~pre) begin
        q <= 0;
      end
      else if(~clr) begin
        q <= 1;
      end
      else begin
        q <= d;
      end
    end
  end
endmodule
```

```
module dlatch_sync_tb;
  reg d,clk,pre,clr;
  wire q;
  dlatch_async inst0(d, clk, pre, clr, q);
  initial begin
```

```

pre=0;clr=0;clk=0;d=0;
#10 pre=0;clr=1;clk=0;d=0;
#10 pre=1;clr=0;clk=0;d=0;
#10 pre=1;clr=1;clk=0;d=0;
#10 pre=1;clr=1;clk=0;d=1;
#10 pre=1;clr=1;clk=1;d=0;
#10 pre=1;clr=1;clk=0;d=0;
#10 pre=1;clr=1;clk=1;d=1;
#10 pre=1;clr=1;clk=0;d=0;
#10 pre=1;clr=1;clk=1;d=0;
#10 pre=1;clr=1;clk=0;d=1;

end
endmodule

```

3 C

```

module dflipflop_async(input d,clk,pre,clr,output reg q);
always @(posedge clk, negedge pre,negedge clr)begin
if(clr==1'b0)begin
q=0;
end
else if(pre==0)begin
q=1;
end
else if(pre==1 & clr==1)begin
if(clk==0)begin
q=q;
end
else if(clk==1)begin
q=d;
end
end
end
endmodule

```

```

module dflipflop_async_tb;
reg d,clk,pre,clr;
wire q;
dflipflop_async inst1(d,clk,pre,clr,q);
initial begin
repeat (10)begin
d=$random();
clk=$random();
pre=$random();
clr=$random();
#10;
end
end
endmodule

```

3 D

```

module dflipflop_sync(input clr, pre, d, clk, output reg Q);

```

```
always @ (posedge clk) begin
  if(~clr) begin
    Q <= 0;
  end
  else if(~pre) begin
    Q <= 1;
  end
  else begin
    Q <= d;
  end
end
endmodule
```

```
module dflop_sync_tb;
  reg d,clk,pre,clr;
  wire q;
  dflop_sync inst0(clr, pre, d, clk, q);
  initial begin
    pre=0;clr=0;clk=0;d=0;
    #10 pre=0;clr=1;clk=0;d=0;
    #10 pre=1;clr=0;clk=0;d=0;
    #10 pre=1;clr=1;clk=0;d=0;
    #10 pre=1;clr=1;clk=0;d=1;
    #10 pre=1;clr=1;clk=1;d=0;
    #10 pre=1;clr=1;clk=0;d=0;
    #10 pre=1;clr=1;clk=1;d=1;
    #10 pre=1;clr=1;clk=0;d=0;
    #10 pre=1;clr=1;clk=1;d=0;
    #10 pre=1;clr=1;clk=0;d=1;

  end
endmodule
```