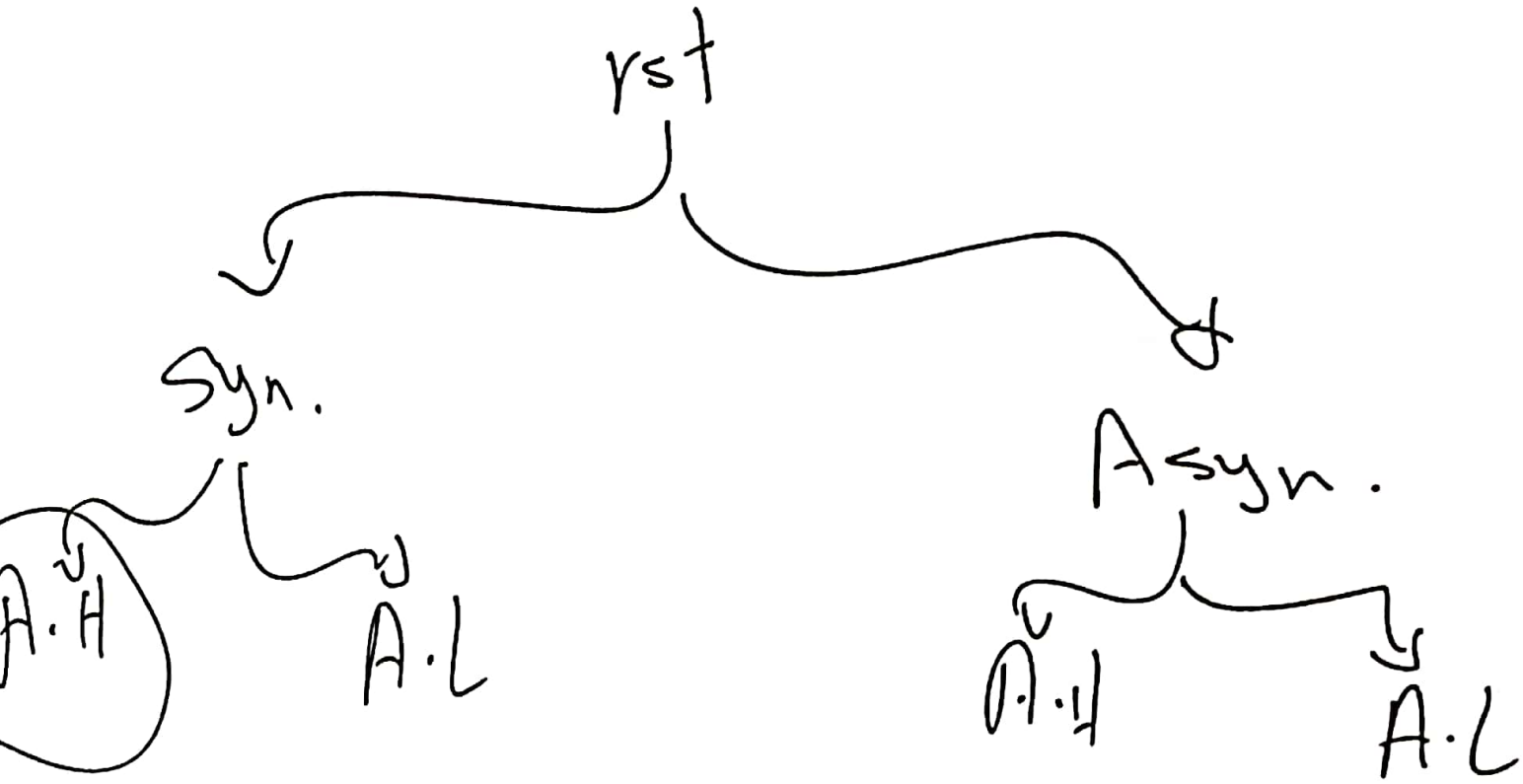


(clk' event and clk = '1')

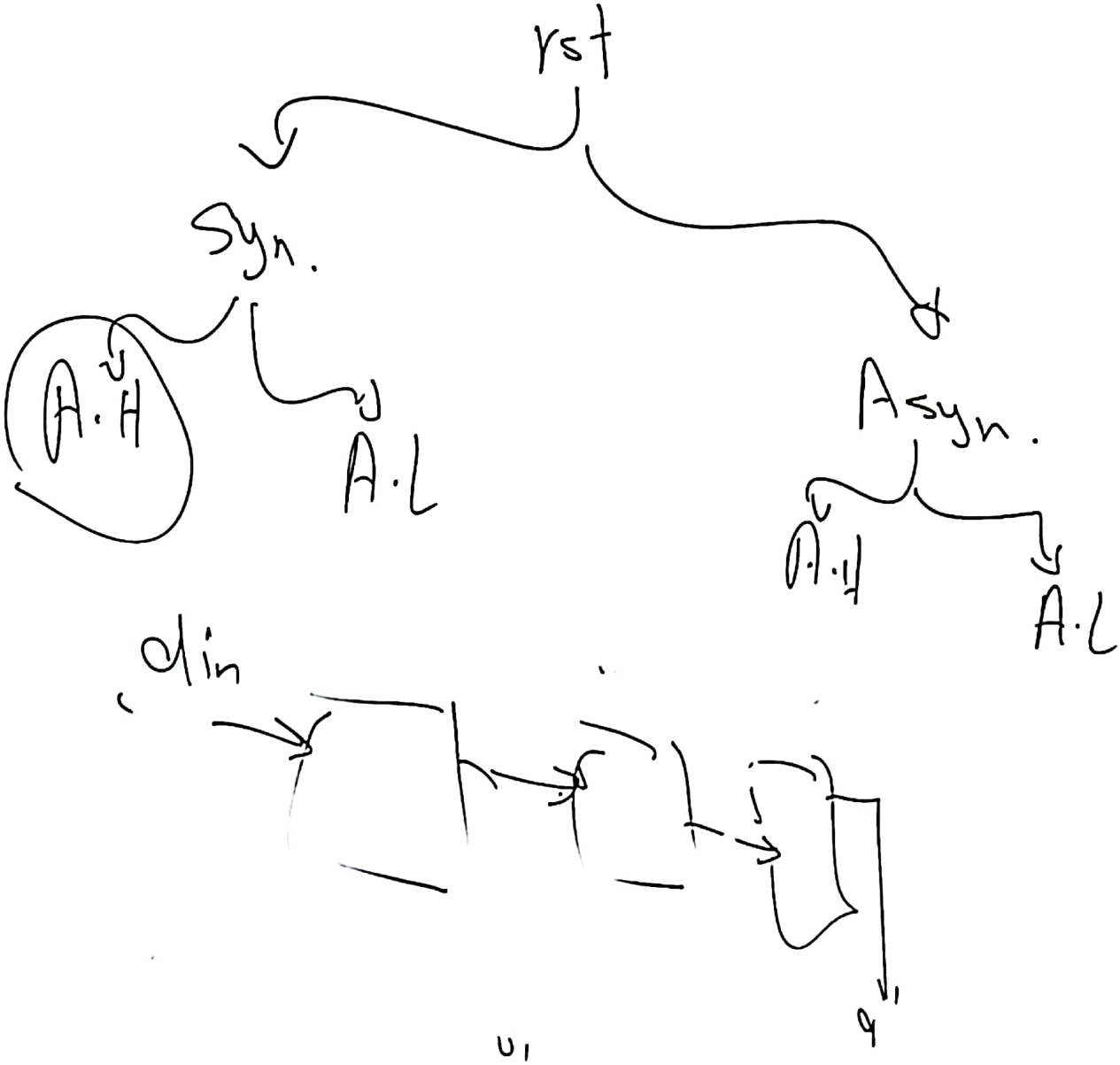
```
entity D_ff is
 Port (din, clk, rst : in std_logic;
       dout          : out std_logic);
end D_FF;
architecture rtl of D_FF is
begin
process (clk)
 begin
② → If (rising_edge(clk)) Then
① → If (rst = '1') Then
    dout <= '0';
   else
    dout <= din;
   endif;
end if;
end process;
end RTL;
```

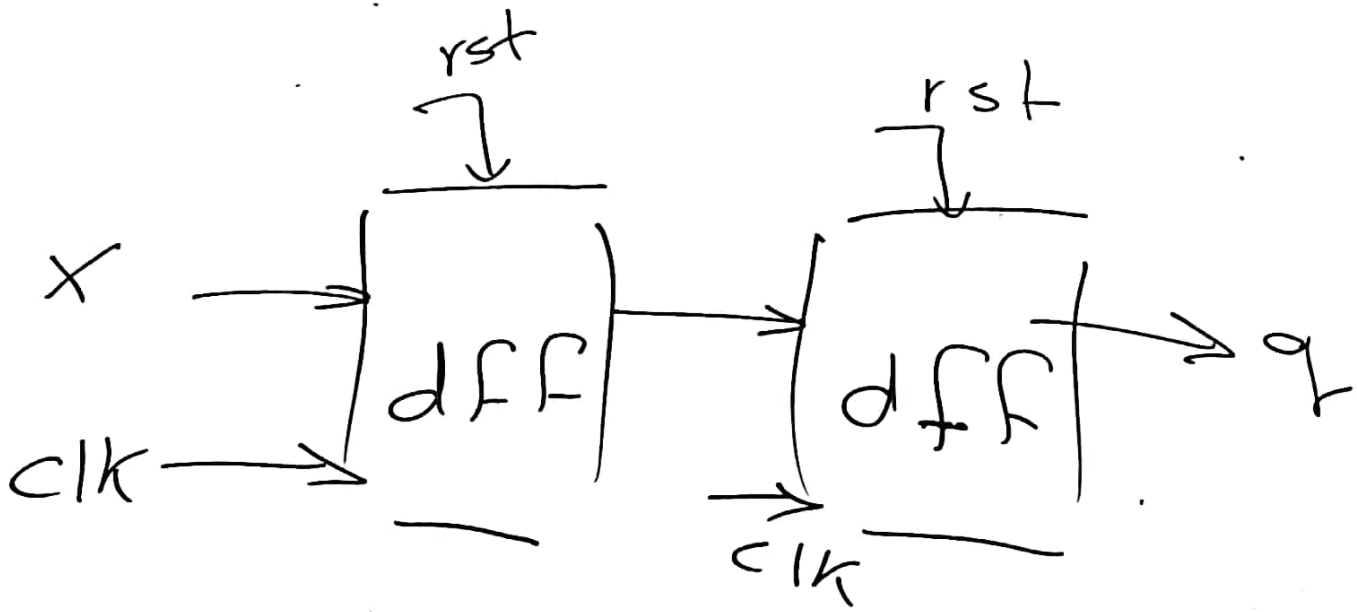
(clk' event and clk = '1')

```
entity D_FF is
 Port (din, clk, rst : in std_logic;
       dout          : out std_logic);
end D_FF;
architecture rtl of D_FF is
begin
process (clk)
 begin
① → If (rising_edge(clk)) Then
② → If (rst = '1') Then
     dout <= '0';
    else
     dout <= din;
    end if;
end if;
end process;
end RTL;
```

(clk'event and clk = '1')

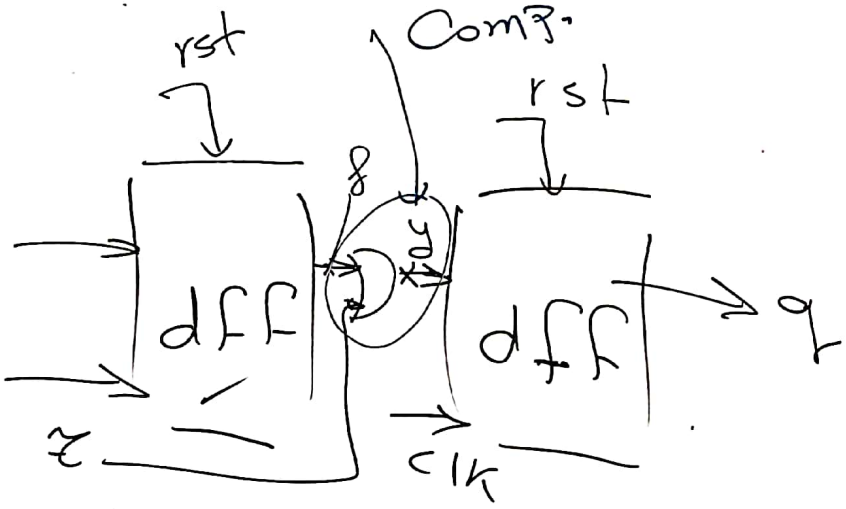
```
entity D_FF is
 Port (din, clk, rst : in std_logic;
       dout          : out std_logic);
end D_FF;
architecture rtl of D_FF is
begin
Process (clk, rst)
 begin
If (rst = '1') Then
 dout <= '0';
elsif ( rising_edge (clk) ) Then
   dout <= din;
endif;
end process;
end rtl;
```





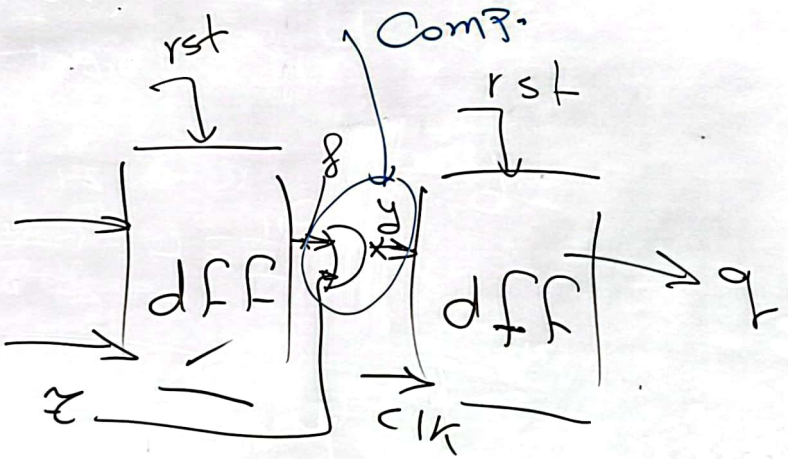
Circuit (1)

Cir-2




```
entity cir_1 is
Port(x, rst, clk : in std_logic;
     q : out std_logic);
end cir_1 ;
architecture rtl of cir_1 is
Signal y,f : std_logic;
begin
process(clk, rst)
begin
If (rst ='1') Then
  q <= '0';
  y <= '0';
elsif(rising_edge(clk)) Then
  f <= x;
  y <= f or
  q <= y;
end if;
end process;
end rtl;
```

Cir-2




```
entity cir_1 is
Port(z,x, rst, clk : in std_logic;
     q : out std_logic);
end cir_1 ;
architecture rtl of cir_1 is
Signal f : std_logic;
begin
process(clk, rst)
variable y : std_logic;
begin
if (rst ='1') Then
q <= '0';
y <= '0';
elsif(rising_edge(clk)) Then
f <= x;
y := f or z;
c <= y;
end if;
end process;
end rtl;
```