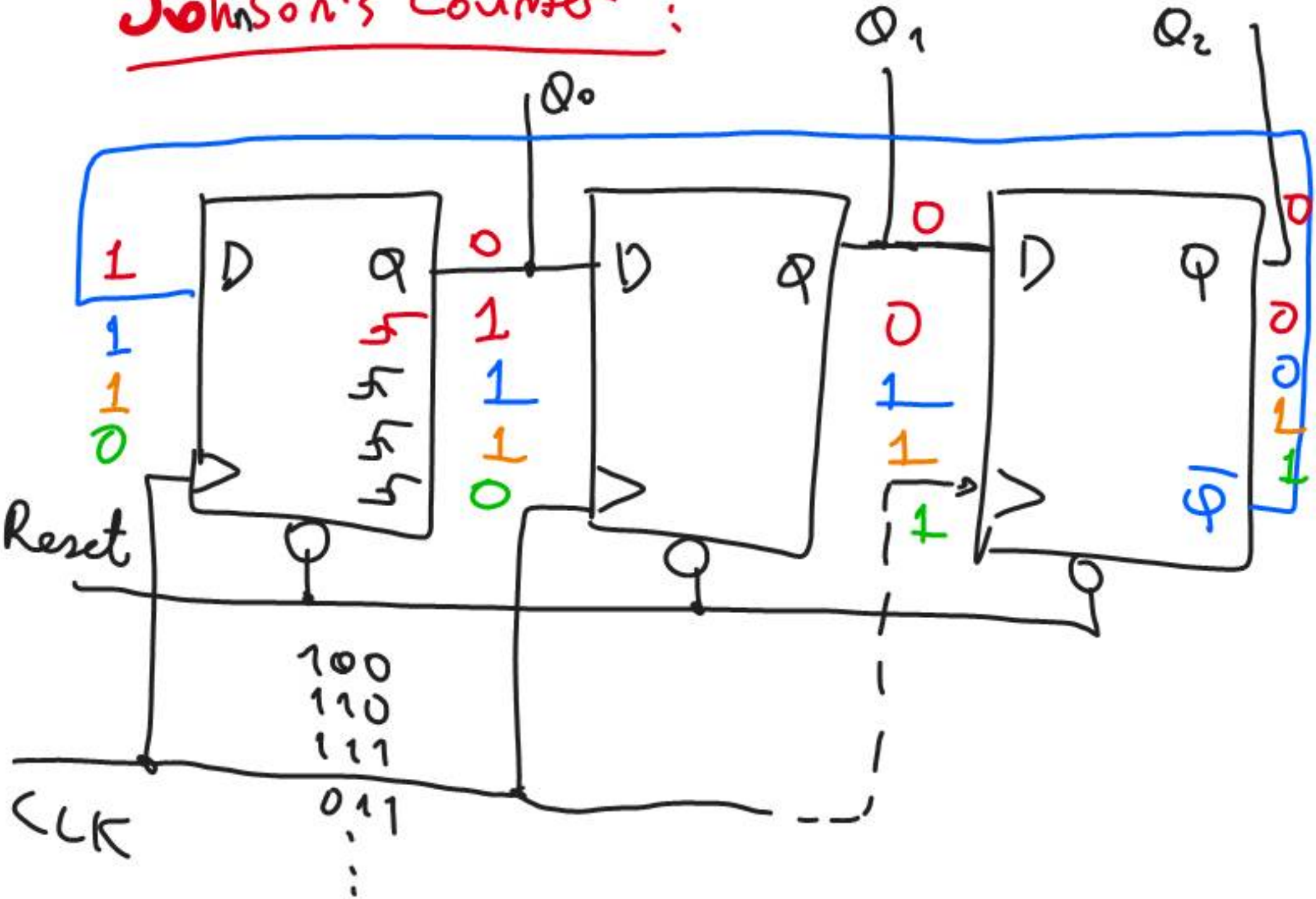
Johnson's Counter :



STORAGE ELEMENTS

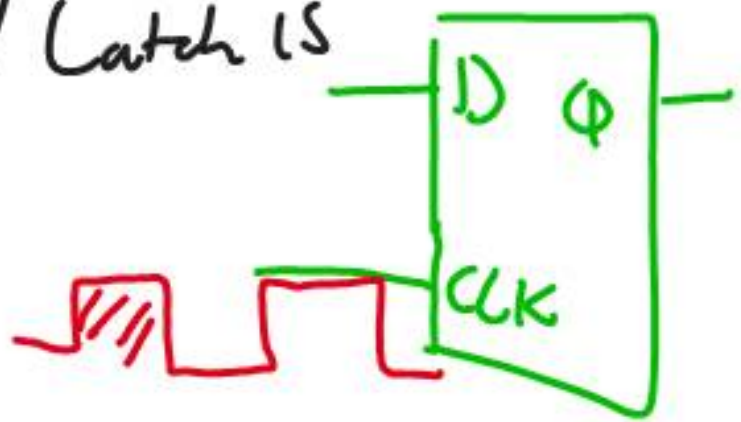
Remember:

```
PROCESS ( A, B)
BEGIN
IF A=B THEN
  AeqB <= '1';
END IF;
END PROCESS;
END Behavior;
```

This is a memory element: it keeps this value forever.

a Gated D-Latch

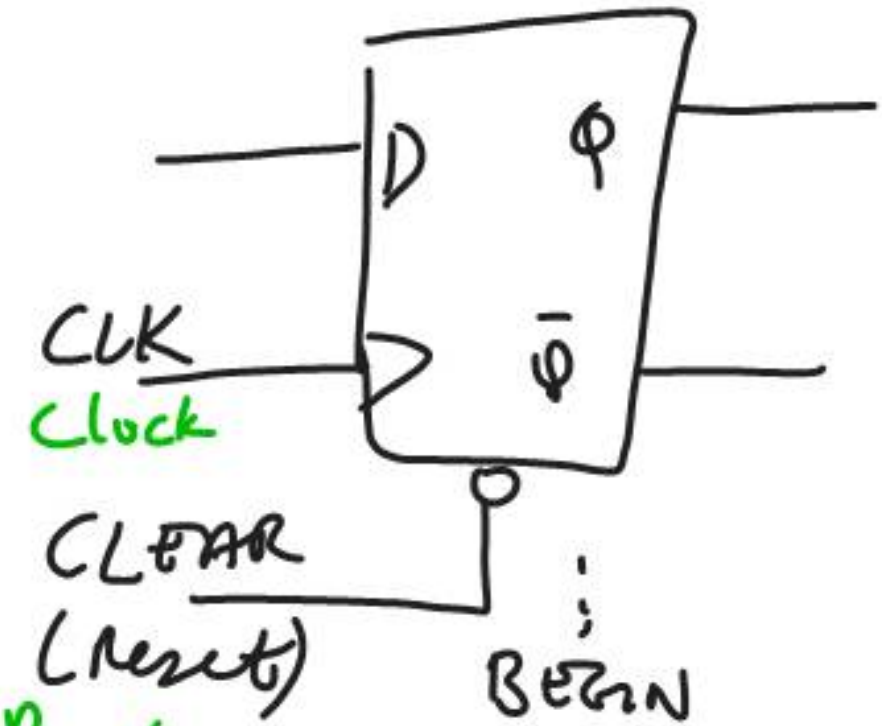
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY Latch IS
PORT ( D, CLK : IN  STD_LOGIC;
       Q      : OUT STD_LOGIC);
END Latch;
ARCHITECTURE Behavior OF Latch IS
BEGIN
PROCESS (D, CLK)
BEGIN
  IF CLK = '1' THEN
    Q <= D;
  END IF;
END PROCESS;
END Behavior;
```



Alternative code for a DFF:

```
:
ARCHITECTURE Behavior of DFF IS
BEGIN
PROCESS                          -- no sensitivity list
BEGIN
  WAIT UNTIL Clock'EVENT AND Clock = '1';
  Q <= D;
END PROCESS;
END Behavior;
```

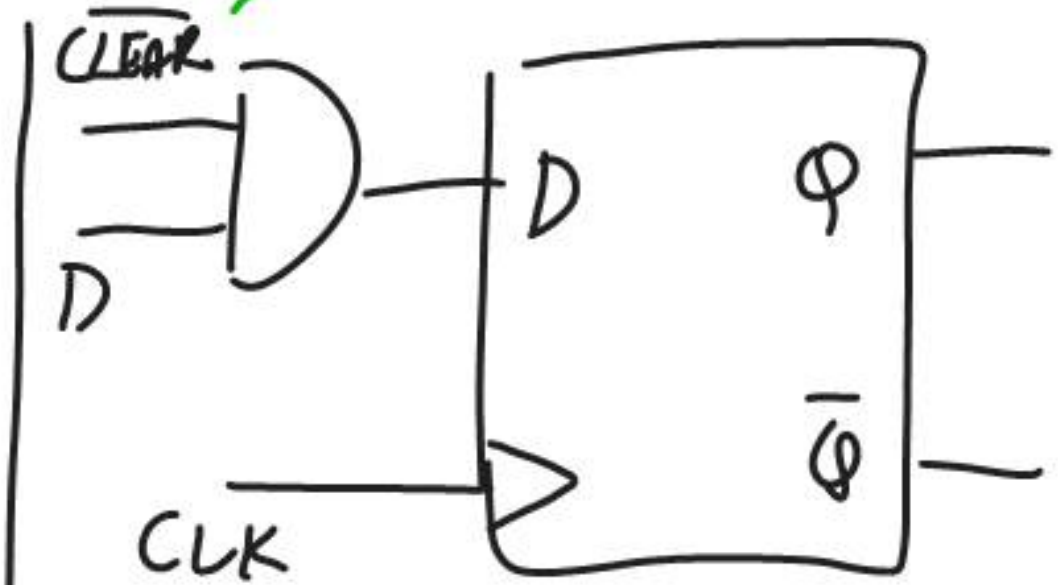
ASYNCHRONOUS RESET




```
;
BEGIN
PROCESS(Resetn, Clock)
BEGIN
IF Resetn='0' THEN
   Q<='0';
ELSIF Clock'EVENT AND Clock='1' THEN
   Q<=D;
ENDIF
```

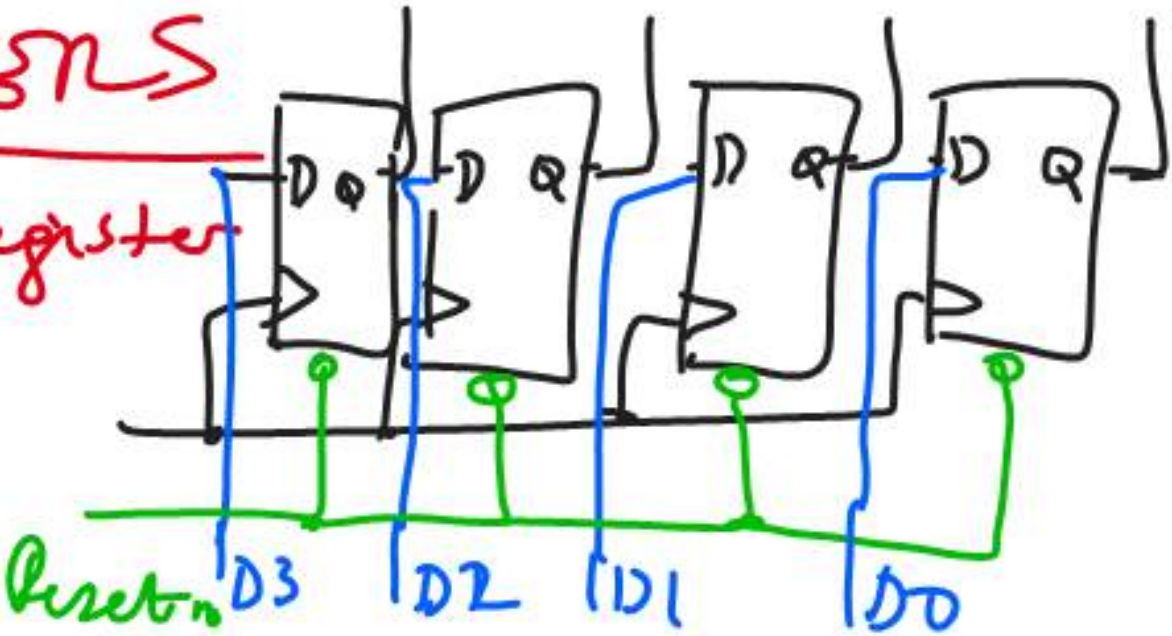
SYNCHRONOUS RESET

Resetn




```
BEGIN
WAIT UNTIL Clock'EVENT
    AND Clock='1';
IF Resetn='0' THEN
   Q<='0';
ELSE
   Q<=D;
ENDIF;
```

REGISTERS



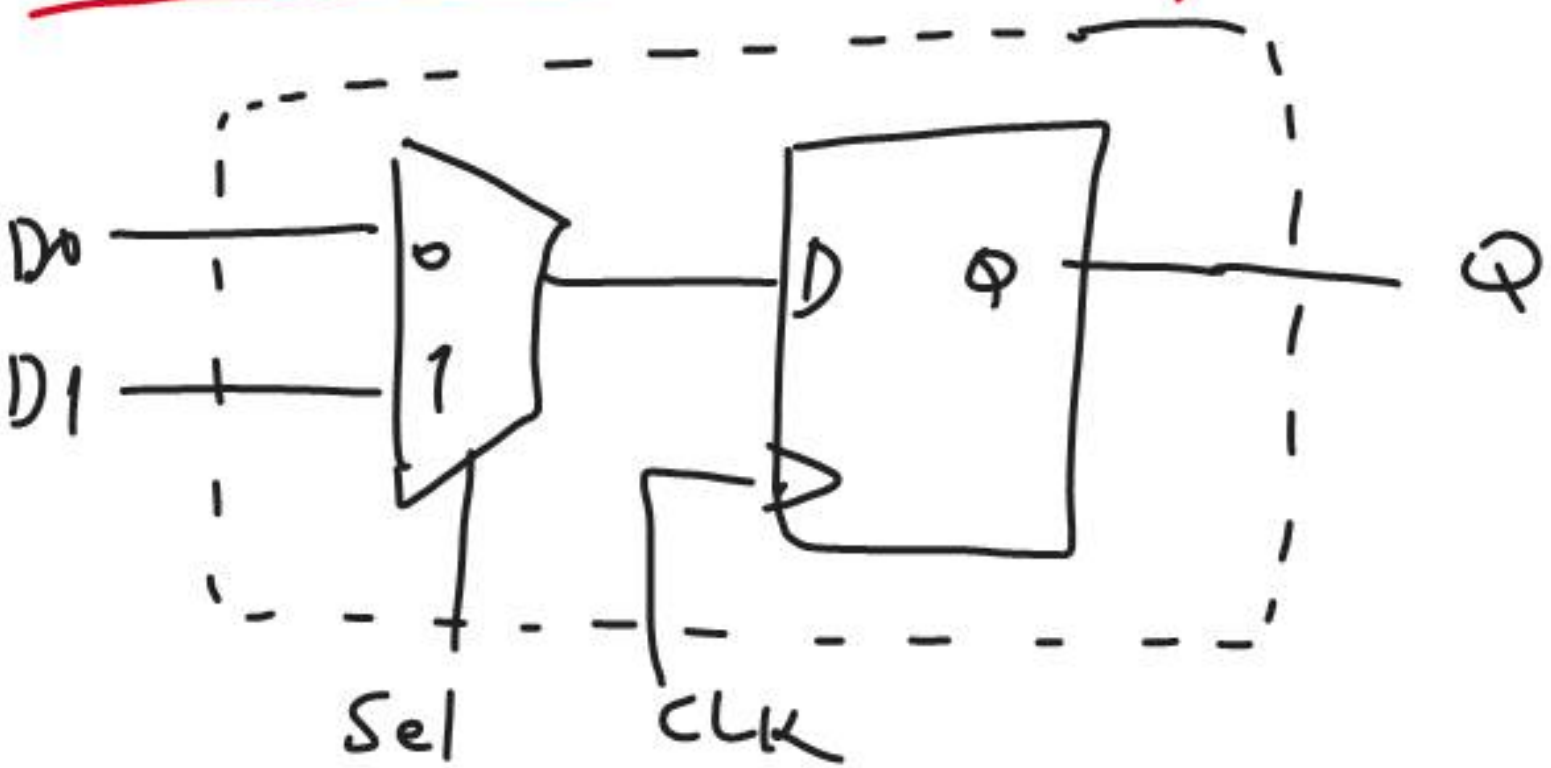
1) A 4-bit parallel Register

```
Library ieee;
use ieee.---;

ENTITY ParReg IS
PORT ( D : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
       Reset, Clock : IN STD_LOGIC;
       Q : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END ParReg;

ARCHITECTURE Behavior OF ParReg IS
BEGIN
```

```
PROCESS (Resetn, Clock);
BEGIN
  IF Resetn = '0' THEN
    Q <= "0000";
  ELSIF Clock'EVENT AND CLOCK = '1' THEN
    Q <= D;
  END IF;
END PROCESS;
END Behavior;
```

Asynchronous reset

A Mux and a D-FF ⇒ muxDFF

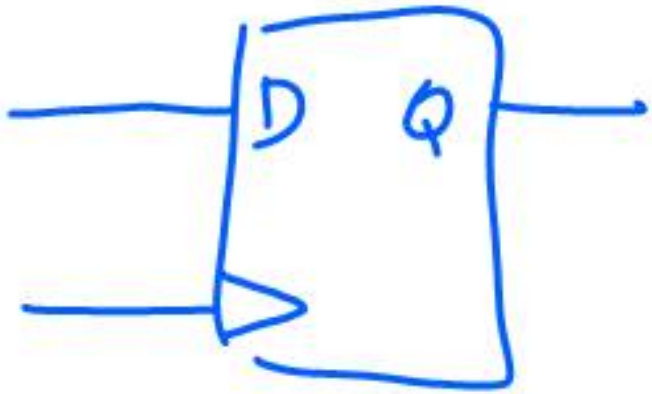


We will use this later. Lets define it as a COMPONENT :

```
LIBRARY ieee;

USE ieee.----;
COMPONENT MUXDFF
ENTITY muxDFF IS

PORT( D0, D1, Sel, Clock : IN  STD_LOGIC;

      Q  : OUT  STD_LOGIC);

END muxDFF;

ARCHITECTURE Behaviour of muxDFF IS
IS BEGIN
PROCESS
BEGIN
```

```
WAIT UNTIL Clock'EVENT AND Clock = '1';
IF Sel = '0' THEN
    Q <= D0;
ELSE
    Q <= D1;
END IF;
END PROCESS;
END Component;
END Behavior;
```

A D Flip Flop



⋮

```
ARCHITECTURE Behavior of DFF IS
 BEGIN
 PROCESS( Clock)
  BEGIN
  IF Clock'EVENT AND Clock = '1' THEN
   Q <= D;
  END IF;
 END PROCESS;
END Behavior;
```

a positive edge triggered clock