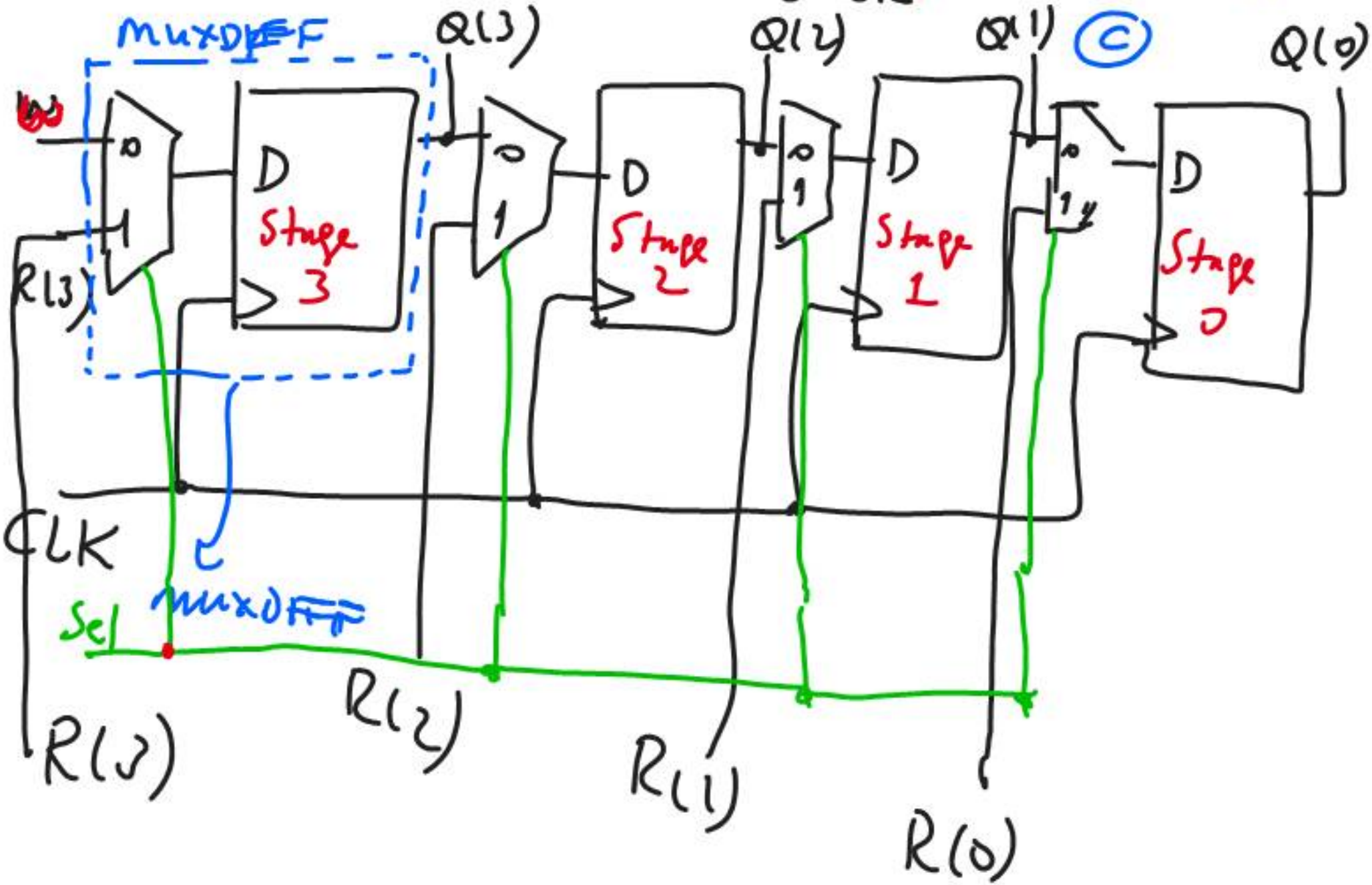
A 4-bit SHIFT REGISTER (with load capability)

02.05.2011 ©



VHDL code 1 :

```
ENTITY ShiftReg IS
PORT ( R : IN STD_LOGIC_VECTOR( 3 DOWNTO 0);
       L, W, CLK : IN STD_LOGIC;
       Q : OUT BUFFER STD_LOGIC_VECTOR ( 3 DOWNTO 0));
END ShiftReg;
```

Sel

input and at the same time could be output

```
ARCHITECTURE Structure OF ShiftReg IS
  COMPONENT muxDFF
  PORT ( D0, D1, Sel, CLK : IN STD_LOGIC;
                 Q        : OUT STD_LOGIC);
  END COMPONENT;
```

```
BEGIN
Stage3 : MUXDFF PORTMAP (W, R(3), L, CLK, Q(3));
Stage2 : MUXDFF PORTMAP (Q(3), R(2), L, CLK, Q(2));
Stage1 : MUXDFF PORTMAP (Q(2), R(1), L, CLK, Q(1));
Stage0 : MUXDFF PORTMAP (Q(1), R(0), L, CLK, Q(0));
END Structure;
```

VHDL Code 2:

```
  ⋮
ARCHITECTURE BEHAVIOR OF ShiftReg IS
BEGIN
 PROCESS                 * PROCESS has no sensitivity List
 BEGIN                     because we use WAIT UNTIL -- statement.
  WAIT UNTIL CLK'EVENT AND CLK = '1';
  IF L = '1' THEN
   Q <= R; --Parallel 4 bit data(R) is loaded.
  ELSE                          Q(3) <= W;
    Q(0) <= Q(1);          END IF;
    Q(1) <= Q(2);          END PROCESS;
    Q(2) <= Q(3);          END Behavior;
```

VHDL Code 3:

```
:
ENTITY Shifter IS
GENERIC ( N : INTEGER := 8);
PORT ( R : IN STD_LOGIC_VECTOR( N-1 DOWN TO 0);
       CLK, L, W : IN STD_LOGIC;
       Q : BUFFER STD_LOGIC_VECTOR ( N-1 DOWN TO 0));
END Shifter;
ARCHITECTURE Behavior OF Shifter IS
 BEGIN
 PROCESS -- no sensitivity list
  BEGIN
  WAIT UNTIL CLK'EVENT AND CLK = '1';
```

```
IF L='1' THEN
  Q <= R;
ELSE
Genbits: FOR i IN 0 TO N-1 LOOP
    Q(i) <= Q(i+1)
  END LOOP;
  Q(N-1) <= W;
END IF;
END PROCESS;
END Behavior;
```

COUNTERS

UPCOUNTER: (4-bit)

```
:
USE ieee.std_logic_unsigned.all; -- an Number!
ENTITY upcount IS
PORT( CLK, Reset, E : IN STD_LOGIC;
      Q             : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END upcount;
ARCHITECTURE Behavior OF upcount IS
SIGNAL Count : STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
```

A count signal

```
PROCESS ( CLK, Resetn)
BEGIN
  IF Resetn = '0' THEN
    Count <= "0000" ;   -- asynchronous reset
  ELSIF (CLK'EVENT AND CLK = '1') THEN
    IF E = '1' THEN   -- Enable count
      Count <= Count+1 -- counts up, (Arithmetic!)
    ELSE
      Count <= Count;
    END IF;
  END IF;
```

END PROCESS;

Q <= Count; — Final count calculated in the PROCESS is assigned to Q output

End Behavior;