Flip-flop di tipo D *edge triggered*

```
ENTITY flip_flop_D IS
   PORT (d, clock: IN BIT ;  q: OUT BIT );
END flip_flop_D;

ARCHITECTURE arch1 OF flip_flop_D IS
BEGIN
  PROCESS (clock)
  BEGIN
    IF (clock'event and clock='1') THEN
      q <= d;
    END IF;
  END PROCESS;
END arch1;
```

oppure

```
ARCHITECTURE arch2 OF flip_flop_D IS
BEGIN
  PROCESS -- no sensitivity list
  BEGIN
    WAIT UNTIL clock'event and clock='1';
    q <= d;
  END PROCESS;
END arch2;
```

Variante con reset sincrono

```
ENTITY flip_flop_D_reset IS
   PORT (d, reset, clock: IN BIT ;  q: OUT BIT );
END flip_flop_D_reset;

ARCHITECTURE behav OF flip_flop_D_reset IS
BEGIN
  PROCESS (clock)
  BEGIN
    IF (clock'event and clock='1') THEN
      IF (reset='1') THEN
        q <= '0';
      ELSE
        q <= d;
      END IF;
    END IF;
  END PROCESS;
END behav;
```

Variante con reset asincrono

```
-- in questo caso il reset e' prioritario!
ARCHITECTURE behav OF flip_flop_D_asyn_reset IS
BEGIN
  PROCESS (clock, reset)
  BEGIN
    IF (reset='1') THEN
      q <= '0';
    ELSEIF (clock'event and clock='1') THEN
      q <= d;
    END IF;
  END PROCESS;
END behav;
```

Latch D

```
ARCHITECTURE behaviour of d_latch IS
BEGIN
  PROCESS (enable)
  BEGIN
    IF (enable = '1') THEN
      outa <= ina;
    END IF;
  END PROCESS;
END behaviour;
```

Latch SR

```
ARCHITECTURE behaviour of set_reset_latch IS
BEGIN
  PROCESS (set, reset)
  BEGIN
    IF (set = '1' and reset = '0') THEN
      outa <= '1';
    ELSEIF (set = '0' and reset = '1') THEN
      outa <= '0';
    END IF;
  END PROCESS;
END behaviour;
```

Flip-flop SR

```
ARCHITECTURE behaviour of set_reset_flip_flop IS
BEGIN
  PROCESS (clock, set, reset)
  BEGIN
    IF (clock'event and clock='1') THEN
      IF (set = '1' and reset = '0') THEN
        q <= '1';
      ELSEIF (set = '0' and reset = '1') THEN
        q <= '0';
      END IF;
    END IF;
  END PROCESS;
END behaviour;
```

Registro 4 bit

```
ENTITY register4 IS
   PORT (    d: IN BIT_VECTOR(3 downto 0);
          clock: bit;
              q: OUT BIT_VECTOR(3 downto 0) );
END register4;

ARCHITECTURE struct of register4 IS
  COMPONENT flip_flop_D
    PORT(d,clock: IN BIT;
                q: OUT);
BEGIN
  ff0: flip_flop_D PORT MAP(d(0),clock,q(0));
  ff1: flip_flop_D PORT MAP(d(1),clock,q(1));
  ff2: flip_flop_D PORT MAP(d(2),clock,q(2));
  ff3: flip_flop_D PORT MAP(d(3),clock,q(3));
END struct;
```

Registro a 4 bit (2)

```
ARCHITECTURE behavior OF register4 IS
BEGIN
  PROCESS (clock)
  BEGIN
    IF (clock'event and clock='1') THEN
      q <= d;
    END IF;
  END PROCESS;
END behavior;
```

Contatore 8 bit

```
ENTITY cont8 IS
PORT(clock, reset: IN STD_LOGIC;
                   outa: OUT STD_LOGIC_VECTOR(0 to 8));
END cont8;

ARCHITECTURE rtl OF cont8 IS
  signal t: STD_LOGIC_VECTOR(0 to 3);
BEGIN
  PROCESS(clock, reset)
  BEGIN
    IF (reset = '1' ) THEN
      t <= "00000000";
    ELSEIF(Šclockevent and clock = '1' ) THEN
      t <= t + "00000001";
    END IF;
  END PROCESS;
```