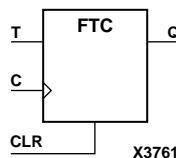


FTC

Toggle Flip-Flop with Toggle Enable and Asynchronous Clear

Architectures Supported

FTC	
Spartan-II, Spartan-IIIE	Macro
Spartan-3	Macro
Virtex, Virtex-E	Macro
Virtex-II, Virtex-II Pro, Virtex-II Pro X	Macro
XC9500, XC9500XV, XC9500XL	Macro
CoolRunner XPLA3	Macro
CoolRunner-II	Macro
CoolRunner-IIS	No



FTC is a synchronous, resettable toggle flip-flop. The asynchronous clear (CLR) input, when High, overrides all other inputs and resets the data output (Q) Low. The Q output toggles, or changes state, when the toggle enable (T) input is High and CLR is Low during the Low-to-High clock transition.

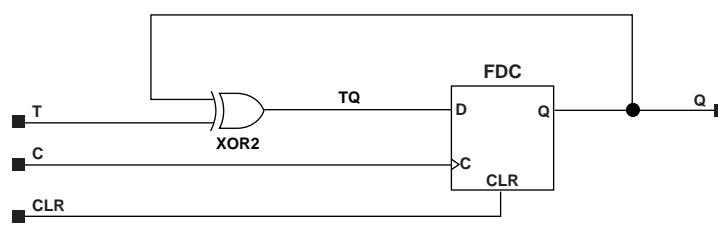
The flip-flop is asynchronously cleared, output Low, when power is applied.

For XC9500/XV/XL, CoolRunner XPLA3, and CoolRunner-II, the power-on condition can be simulated by applying a High-level pulse on the PRLD global net.

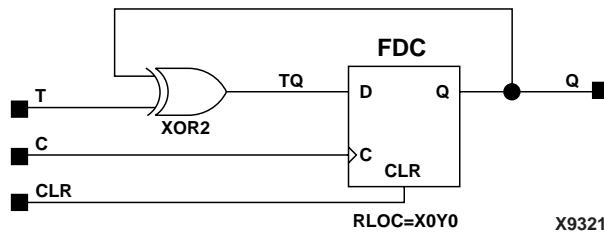
Spartan-II, Spartan-IIIE, Spartan-3, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, and Virtex-II Pro X simulate power-on when global set/reset (GSR) is active.

GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the STARTUP_SPARTAN2, STARTUP_SPARTAN3, STARTUP_VIRTEX, or STARTUP_VIRTEX2 symbol.

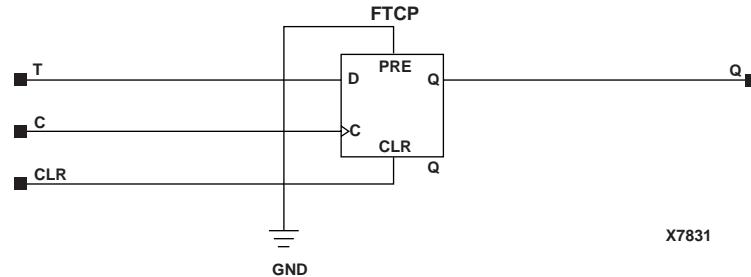
Inputs			Outputs
CLR	T	C	Q
1	X	X	0
0	0	X	No Chg
0	1	↑	Toggle



FTC Implementation Spartan-II, Spartan-IIIE, Virtex, Virtex-E



FTC Implementation Spartan-3, Virtex-II, Virtex-II Pro, Virtex-II Pro X



FTC Implementation XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II

Usage

For HDL, this design element can be instantiated or inferred.

VHDL Inference Code

```
architecture Behavioral of ftc is
begin
  process (C, CLR)
  begin
    if (CLR='1') then
      Q <= '0';
    elsif (C'event and C='1') then
      if (T='1') then
        Q <= not Q;
      end if;
    end if;
  end process;
end Behavioral;
```

Verilog Inference Code

```
always @(posedge C or posedge CLR)
begin
  if (CLR)
    Q <= 0;
  else if (T)
    Q <= !Q;
end
```

VHDL Instantiation Template

```
-- Component Declaration for FTC should be placed
-- after architecture statement but before begin keyword

component FTC
    port (Q : out STD_ULOGIC;
          C : in STD_ULOGIC;
          CLR : in STD_ULOGIC;
          T : in STD_ULOGIC);
end component;

-- Component Attribute specification for FTC
-- should be placed after architecture declaration but
-- before the begin keyword

-- Enter attributes here

-- Component Instantiation for FTC should be placed
-- in architecture after the begin keyword

FTC_INSTANCE_NAME : FTC
    port map (Q => user_Q,
              C => user_C,
              CLR => user_CLR,
              T => user_T);
```

Verilog Instantiation Template

```
FTC FTC_instance_name (.Q (user_Q),
                      .C (user_C),
                      .CLR (user_CLR),
                      .T (user_T));
```

Commonly Used Constraints

BLKNM, HBLKNM, HU_SET, INIT, IOB, LOC, REG, RLOC, TIMEGRP, TNM, U_SET, XBLKNM

