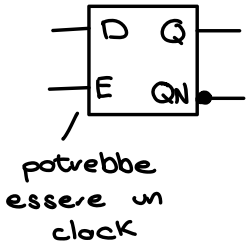
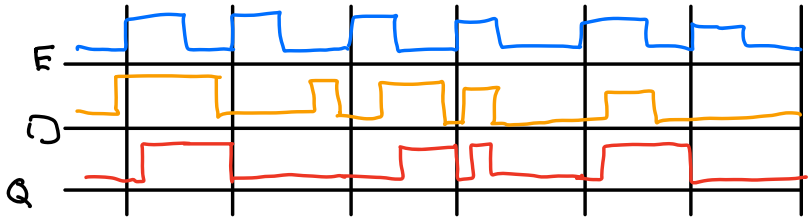
→ **D Latch** → componente di memoria





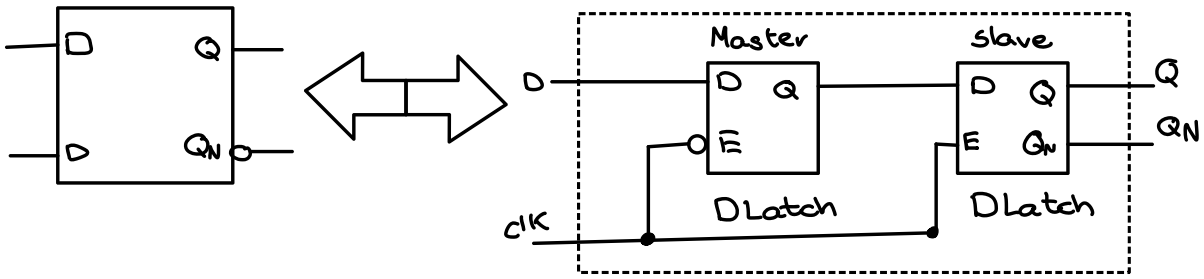
Enable comanda

Su ModelSim

```
process(D, clk)
begin
      if clk = '1' then
       Q <= D;
       QN <= not(D);
       end if;
end process;
```

```
Q <= D when clk = '1' else Q
QN <= not Q;
```

→ **Master - Slave Edge D-Flip-Flop**



	INPUTs		OUTPUTs	
NO TOGGLE	0	4	0	1
	1	4	1	0

Il clock da l'enable e fa cambiare stato.

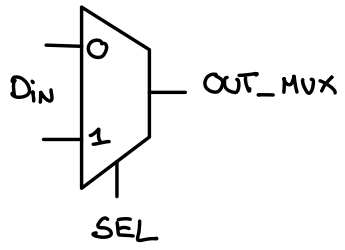
→ Registers → Utilizzati per salvare dati input bus

- sincro → clk
- enable + reset
- PIPO Registers

```
DATA_OUT <= support_signal1;
support_signal2 <= DATA_IN
```
} input e output

```
g1: for k in N0 - 1 downto 0 generate
    ni : flip_flop_D port map (...)
  end generate;
```
} genero i flip flop

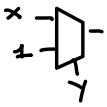
→ Multiplexer → selezionano il bit da prendere




```
if SEL = 0
   OUT_MUX = DIN(0)
else
   OUT_MUX = DIN(1)
endif;
```

$Din = 2^{SEL}$

z = x AND y



z = x OR y