

0.1 Implementazione

L'interfaccia del componente è la seguente:

```
[language=VHDL,caption=Implementazione data-flow dell'anodes manager.]
entity scan_chain is generic (width : integer := 8; shift_direction : std_logic := '1');
Port (clock : in STD_LOGIC; en : in STD_LOGIC; reset_n : in STD_LOGIC; scan_en : in STD_LOGIC;
d_reg : in STD_LOGIC_VECTOR (width - 1 downto 0); scan_in : in STD_LOGIC;
q_reg : out STD_LOGIC_VECTOR (width - 1 downto 0); scan_out : out STD_LOGIC);
endscan_chain;
```

In ingresso, oltre ai segnali di *clock*, *enable* e *reset*, il componente avrà i seguenti segnali:

- *scan_en*: bit di selezione della modalità di funzionamento (1 per modalità normale, 1 per modalità controllo);
- *d_reg*: valori in ingresso dei flip-flop nel registro;
- *scan_in*: valore in ingresso da inserire nel registro in caso di shift;

In uscita, invece, *q_reg* sarà l'uscita del registro, mentre *scan_out* sarà il bit tirato fuori dal registro in caso di shift.

Per quanto riguarda l'implementazione, è stata utilizzata una descrizione di tipo *structural*. In particolare, è stato generato un numero di flip-flop multiplexati pari al valore generico *width*, che rappresenta il parallelismo del registro. Tale componente è un tipo particolare di flip-flop D il cui ingresso viene prima selezionato tramite multiplexer. Per generare un registro che effettui shift a destra, si fissa come *scan_in* di ogni flip-flop il valore in uscita del flip-flop precedente (alla sua sinistra), dopodiché si utilizza il multiplexer per decidere quale ingresso portare nel flip-flop. In particolare, *scan_en* sarà il segnale di selezione: se 0 (modalità normale), in ingresso avremo il corrispondente valore in ingresso di *d_reg*, se 1 (modalità controllo) l'ingresso del flip-flop sarà *scan_in*, e dunque l'uscita di quello precedente. Il discorso è analogo nel caso di shift a sinistra: in base al valore generico *shift_direction* si stabilisce quale tipologia di shift register si vuole generare e dunque come collegare tra loro i flip-flop. L'implementazione completa è consultabile qui: `run:./esercizio05/design/scan_chain.vhdscan_chain.vhd`.

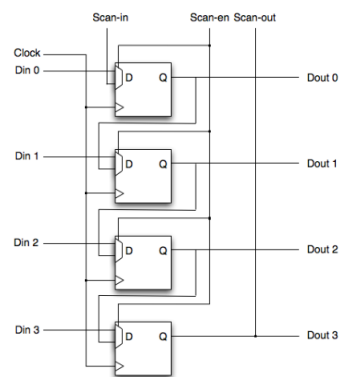


Figure 1: Architettura del componente scan_chain.