

Progettare un modulo VHDL che computi il quadrato di un numero per somme successive, quindi senza usare un moltiplicatore.

Esempio:

$$2^2 = 2 + 2$$

$$3^2 = 3 + 3 + 3$$

Il modulo VHDL accede a una memoria indirizzata al byte. L'operando da elevare al quadrato si trova nella locazione di memoria 0. Il risultato, di dimensione 2 byte, viene scritto nel seguente modo: il byte meno significativo viene scritto nel byte 1, mentre il byte più significativo all'indirizzo 2.

Il modulo VHDL è dotato di un ingresso START e una uscita DONE. Quando l'ingresso START è alto per almeno un ciclo di clock il modulo inizia la computazione. Dopo aver scritto il risultato, il modulo porta l'uscita DONE alta per un ciclo. Non è possibile inviare altri comandi START prima dell'arrivo del DONE.

Il componente da descrivere deve avere la seguente interfaccia.

```
entity square is
  port (
    i_clk          : in  std_logic;
    i_res          : in  std_logic;
    i_start        : in  std_logic;
    o_done         : out std_logic;
    i_data         : in  std_logic_vector(7 downto 0);
    o_data         : out std_logic_vector(7 downto 0);
    o_addr         : out std_logic_vector(15 downto 0);
    o_en           : out std_logic;
    o_we           : out std_logic;
  );
end square;
```

In particolare:

- i_clk è il segnale di CLOCK in ingresso generato dal TestBench;
- i_res è il segnale di RESET che inizializza la macchina pronta per ricevere il primo segnale di START;
- i_start è il segnale di START generato dal Test Bench;
- o_done è il segnale di uscita che comunica la fine dell'elaborazione e il dato di uscita scritto in memoria;
- i_data è il segnale (vettore) che arriva dalla memoria in seguito ad una richiesta di lettura;
- o_data è il segnale (vettore) di uscita dal componente verso la memoria.
- o_addr è il segnale (vettore) di uscita che manda l'indirizzo alla memoria;
- o_en è il segnale di ENABLE da dover mandare alla memoria per poter comunicare (sia in lettura che in scrittura);
- o_we è il segnale di WRITE ENABLE da dover mandare alla memoria (=1) per poter scriverci. Per leggere da memoria esso deve essere 0;