

0.1 Display su Nexys 4

La board Nexys 4 DDR ha installato a bordo due batterie di 4 digit ciascuna, per un totale di 8 digit. La soluzione che abbiamo visto precedentemente però permette di controllare una sola batteria di 4 digit, e non potendo instanziare due componenti che controllano ciascuno 4 digit, è necessario apportare alcune modifiche agli elementi che costituiscono il componente che pilota i display a 7 segmenti.

In particolare, di seguito, vediamo alcune modifiche che abbiamo apportato ai componenti per poter pilotare tutte le digit che mette a disposizione la Nexys 4 DDR.

```
[language=VHDL,caption=display7segment]entitydisplay7segmentisPORT(enable :
inSTDLOGIC;clock : inSTDLOGIC;reset : inSTDLOGIC;values : inSTDLOGIC_VECTOR(31downto0);
inSTDLOGIC_VECTOR(7downto0);enabledigit : inSTDLOGIC_VECTOR(7downto0);anodes :
outSTDLOGIC_VECTOR(7downto0);cathodes : outSTDLOGIC_VECTOR(7downto0));enddisplay7segment
...
component counterUpMod2nResrisisGENERIC(n : NATURAL := 3);PORT(enable :
inSTDLOGIC;resetn : inSTDLOGIC;clock : inSTDLOGIC;countn : outSTDLOGIC;COUNTS :
outSTDLOGIC_VECTOR((n - 1)downto0));endcomponent;
```

Nella top level entity del componente che permette di mostrare i valori sui display, il numero di segnali che indicano il la cifra da mostrare passano da 16 a 32, in quanto per ogni digit servono 4 bit per poter codificare i valori, in esadecimale, che possiamo mostrare su di essi. Poichè il numero di display da pilotare passa da 4 a 8, anche il numero di segnali che pilotano i punti , le digit e gli anodi aumentano. Inoltre il contatore non è più un contatore modulo 4 ma è un contatore modulo 8, in quanto devono essere abilitate 8 digit.

Tali modifiche riguardano anche l'anodes_manager che deve pilotare 8 digit e non più 4, pertanto il decoder da un decoder 2:4 diventa un decoder 3:8 come si nota di seguito.

```
[language=VHDL,caption=Abitazione degli anodi] with selectdigitselectanodesswitching <=
x"01"when"000",x"02"when"001",x"04"when"010",x"08"when"011",x"10"when"100",x"20"when"101",x
0')whenothers;endcase;
P
```

er il cathodes_manager cambia il numero di nibble che dobbiamo gestire, infatti anche essi passano da 4 a 8, ognuna di esse è costituita da 4 bit di values, stringa del valore da mostrare sul display codificato in codifica binaria classica. In particolare partendo dal LSB ogni 4 bit di values codifica il valore da mostrare su una delle digit. Anche il process che si occupa di assegnare ai catodi la corretta codifica del valore viene estesa per poter gestire 8 digit e non più 4.

```
[language=VHDL,caption=cathodesmanager.vhd..]aliasdigit0isvalues(3downto0);aliasdigit1isvalues(7do
...
digitsswitching : process(selectdigit,values)begincaseselectdigitiswhen"000" =>
nibble <= digit0;when"001" => nibble <= digit1;when"010" => nibble <=
digit2;when"011" => nibble <= digit3;when"100" => nibble <= digit4;when"101" =>
```

```

nibble <= digit5; when "110" => nibble <= digit6; when "111" => nibble <=
digit7; when others => nibble <= (others => '0'); endcase; endprocess;

```