Elektronikpraktikum

16. Juli 2002

Erweiterung 4-Bit-Zähler auf 8-Bit-Zähler

1 Funktionsweise des 4-Bit-Zählers

Das Programm besteht lediglich aus vier Gleichungen. Der Zählerstand ist in dem Bit-Array q[0:3] gespeichert, und wird durch das externe Programm leddigit (Datei leddigit.pds) für die Ausgabe auf einer 7-Segment-Anzeige umformatiert (dort sind in einem Array die zu allen 16 Hexadezimalziffern gehörenden Bitmuster für die 7-Segment-Darstellung abgespeichert).

Die einzelnen Bits des Zählers werden auf ein Clock-Signal invertiert (jeweils bei steigender Flanke), aber der Unterschied ist, welches Signal sie als Clock benutzen. Das niederwertigste Bit q0 verwendet den externen Clock-Eingang, und wird somit bei jedem Takt der externen Clock invertiert. Die nächsten Bits benutzen das jeweilige vorherige Bit als Clock-Signal, d.h. das Bit q1 wird immer dann invertiert, wenn q0 von 0 auf 1 wechselt (d.h. zu jedem 2. Takt der externen Clock), q2 wird invertiert, wenn q1 von 0 auf 1 wechselt (d.h. zu jedem 4. Takt der externen Clock), und bei q3 ist es analog.

Mit den vier Bit des Counters können 16 Werte dargestellt werden ($2^4 = 16$), was genau für eine Stelle einer 7-Segment-Anzeige in hexadezimaler Darstellung reicht. Wenn man den Counter mit dem maximalen Startwert lädt (1111_b) und den externen Clock-Eingang mit dem Funktionsgenerator betreibt (Rechtecksignal, sehr niedrige Frequenz von ca. 1 Hz), kann man das Herunterzählen schön beobachten.

2 Erweiterung

Wir wollten beide 7-Segment-Anzeigen des Experimentierboards nutzen, d.h. der Counter muss dazu erweitert werden - er darf jetzt 8 Bit breit sein. Dazu müssen in der Pin-Deklaration die Anschlüsse der zweiten 7-Segment-Anzeige angegeben werden (d.h. jetzt ist s[0:13] statt s[0:6]), und der Counterstand q muss von 4 auf 8 Bit erweitert werden (q[0:7] statt q[0:3]). Außerdem müssen zusätzliche Gleichungen hinzugefügt werden, die - analog zu den anderen - jedes der neuen Bits genau dann invertieren, wenn das nächst-niederwertigere von 0 auf 1 wechselt. Wenn man will, kann man außerdem den Simulations-Abschnitt anpassen („VECTOR out ...” und „PRLDF ...” auf 8 Bit erweitern).

Hindernisse und Probleme: Gab es nicht - erstaunlicherweise funktionierte diese Modifikation gleich beim ersten Mal...

3 Quelltext

```
;---------------------------------------
; A 8-bit down counter built from DFF flip-flops
;---------------------------------------
TITLE   8-bit down counter

CHIP    dncnt8  NFX780_84

PIN  47      clock      ;* clock signal for driving counter
PIN  [48:51] unused[0:3]
PIN  [77:78] unused[4:5]
PIN          q[0:7]     ;* 8-bit counter output
PIN  [34:37] s[0:3]     ;* LED1 segment drivers
PIN  [39:41] s[4:6]     ;* LED1 segment drivers
PIN  [??:??] s[7:10]     ;* LED2 segment drivers
PIN  [??:??] s[11:13]     ;* LED2 segment drivers
MODULE leddigit( d[0:3]=q[0:3], d3=GND, s[0:6]=s[0:6] )
MODULE leddigit( d[4:7]=q[4:7], d3=GND, s[7:13]=s[7:13] )

EQUATIONS
        ; The following statements make the DFF act like a
        ; toggle FF because we load it with the inverse of
        ; what it currently stores.  That makes the output
        ; of this TFF toggle at 1/2 the clock frequency.
        q0 := /q0
        q0.ACLK = clock

        ; This is another toggle FF, but this one is
        ; clocked with the output of the first TFF.  That
        ; makes the output of this TFF toggle at 1/4 of
        ; the clock frequency.
        q1 := /q1
        q1.ACLK = q0

        ; A final TFF whose output toggles at 1/8 the
        ; clock frequency.
        q2 := /q2
        q2.ACLK = q1

        q3 := /q3
```

```
    q3.ACLK = q2

    q4 := /q4
    q4.ACLK = q3

    q5 := /q5
    q5.ACLK = q4

    q6 := /q6
    q6.ACLK = q5

    q7 := /q7
    q7.ACLK = q6

SIMULATION
    VECTOR out := [q7, q6, q5, q4, q3, q2,q1,q0]
    TRACE_ON clock out
    SETF /clock         ; set the clock low
    PRLDF /q0 /q1 /q2 /q3 /q4 /q5 /q6 /q7; clear the counter bits to 0

    ; clock it for awhile to see what it does
    FOR i:=0 to 20 DO
        BEGIN
        CLOCKF clock
        END
```