PLC42VA12 DMA APPLICATIONS

The PLC42VA12 contains 10 flip-flops that may flexibly be configured to build counters, shifters or any customized state machine required. With today's 32-bit micro-processors, there is a need for user-designed, system-specific DMA controllers that can generate addresses or count nibbles, bytes, half-words or words. Applications for these controllers include I/O concentration and cache subsystem updating. Typically, these devices can be preset or cleared and count (up) by 1, 2, or 4 depending on the chosen circumstances. A solution for the problem is presented in this section to illustrate solving the problem with Philips Semiconductors SNAP design software. The SNAP files are presented in Figure 1.

```
@PINLIST
CLK  I ;MODE0   I ;MODE1  I;RST I;LOAD I;CO  O;TOE  I;
OUTA O;OUTB  O;OUTC O;OUTD B;OUTE B;OUTF B;OUTG  G;OUTH B;OUTI B;OUTJ B;

@Logic Equations
               "  mode1  mode0    function
                    0      0     count by 1
                    0      1     count by 2
                    1      0     count by 4
                    1      1       illegal
               "
QUOTA.J = 1*/load*/mode0*/mode1           "load disables P-terms"
          + mode0*/load                   "force 0 for count by 2"
          + mode1*/load          + XOUTA; "or count by 4"
QUOTA.K = 1*/load*/mode0*/mode1  + YOUTA; "XOUTA,YOUTA are outputs of "
                                          "tristate inputs "
QOUTB.J = outa*/load*/mode0*/mode1
          + 1*/mode1*mode0*/load
          + mode1*/mode0*/load   + XOUTB; "force 0 count by four"
QOUTB.K = outa*/load*/mode0*/mode1
          + mode1*/mode0*/load   + YOUTB;

DOUTC = outa*outb/load*/mode0*/mode1
          + outb*/mode1*mode0*/load;
QOUTC.J = DOUTC + XOUTC;
QOUTC.K = DOUTC + YOUTC;

DOUTD = outa*outb*outc*/load*/mode0*mode1        "count by 1"
          + outb*outc*/mode1*mode0*/load         "count by 2"
          + outc*mode1*/mode0*/load;             "count by 4"
QOUTD.J = DOUTD + XOUTD;
QOUTD.K = DOUTD + YOUTD;

DOUTE = outa*outb*outc*outd*/load*/mode0*/mode1
          + outb*outc*outd*/mode1*mode0*/load
          + outc*outd*mode1*/mode0*/load;
QOUTE.J = DOUTE + XOUTE;
QOUTE.K = DOUTE + YOUTE;

DOUTF = outa*outb*outc*outd*oute*/load*/mode0*mode1
          + outb*outc*outd*oute*/mode1*mode0*/load
          + outc*outd*oute*mode1*/mode0*/load;
QOUTF.J = DOUTF + XOUTF;
QOUTF.K = DOUTF + YOUTF;

DOUTG = outa*outb*outc*outd*oute*outf*/load*/mode0*/mode1
          + outb*outc*outd*oute*outf*/mode1*mode0*/load
          + outc*outd*oute*outf*mode1*/mode0*/load;
QOUTG.J = DOUTG + XOUTG;
QOUTG.K = DOUTG + YOUTG;

DOUTH = outa*outb*outc*outd*oute*outf*outg*/load*/mode0*/mode1
          + outb*outc*outd*oute*outf*outg*/mode1*mode0*/load;
          + outc*outd*oute*outf*outg*mode1*/mode0*/load;
QOUTH.J = DOUTH + XOUTH;
QOUTH.K = DOUTH + YOUTH;
```

Figure 1. SNAP Files (1 of 3)

```
DOUTI = outa*outb*outc*outd*oute*outf*outg*outh*/load*/mode0*/mode1
          + outb*outc*outd*oute*outf*outg*outh*/mode1*mode0*/load
          + outc*outd*oute*outf*outg*outh*mode1*/mode0*/load;
QOUTI.J = DOUTI + XOUTI;
QOUTI.K = DOUTI + YOUTI;

DOUTJ = outa*outb*outc*outd*oute*outf*outg*outh*outi*/load*/mode0*/mode1
          + outb*outc*outd*oute*outf*outg*outh*outi*/mode1*mode0*/load
          + outc*coutd*oute*outf*outg*outh*outi*mode1*/mode0*/load;
QOUTJ.J = DOUTJ + XOUTJ;
QOUTJ.K = DOUTJ + YOUTJ;

CO = outa*outb*outc*outd*oute*outf*outg*outh*outi*outj*/load*/mode0*/mode1
         + outb*outc*outd*oute*outf*outg*outh*outi*outj*/mode1*mode0*/load
         + outc*outd*oute*outf*outg*outh*outi*outj*mode1*/mode0*/load;

" Reset for all flip-flops "
QOUTA.RST = RST;
QOUTB.RST = RST;
QOUTC.RST = RST;
QOUTD.RST = RST;
QOUTE.RST = RST;
QOUTF.RST = RST;
QOUTG.RST = RST;
QOUTH.RST = RST;
QOUTI.RST = RST;
QOUTJ.RST = RST;

" Flip-flops are followed by tristate outputs which drive the pin "
OUTA = /QOUTA;
OUTB = /QOUTB;
OUTC = /QOUTC;
OUTD = /QOUTD;
OUTE = /QOUTE;
OUTF = /QOUTF;
OUTG = /QOUTG;
OUTH = /QOUTH;
OUTI = /QOUTI;
OUTJ = /QOUTJ;
OUTA.OE = TOE;
OUTB.OE = TOE;
OUTC.OE = TOE;
OUTD.OE = TOE;
OUTE.OE = TOE;
OUTF.OE = TOE;
OUTG.OE = TOE;
OUTH.OE = TOE;
OUTI.OE = TOE;
OUTJ.OE = TOE;
```

Figure 1. SNAP Files (2 of 3)

```
"Pins are fed back to flip-flops using tristate inputs (FF load)"
XOUTA = /OUTA;      "feed.back to J is inverted"
YOUTA =  OUTA;      "feed-back to K IS NOT inverted"
XOUTB = /OUTB;
YOUTB =  OUTB;
XOUTC = /OUTC;
YOUTC =  OUTC;
XOUTD = /OUTD;
YOUTD =  OUTD;
XOUTE = /OUTE;
YOUTE =  OUTE;
XOUTF = /OUTF;
YOUTF =  OUTF;
XOUTG = /OUTG;
YOUTG =  OUTG;
XOUTH = /OUTH;
YOUTH =  OUTH;
XOUTI = /OUTI;
YOUTI =  OUTI;
XOUTJ = /OUTJ;
YOUTJ =  OUTJ;
XOUTA.LD = LOAD;
XOUTB.LD = LOAD;
XOUTC.LD = LOAD;
XOUTD.LD = LOAD;
XOUTE.LD = LOAD;
XOUTF.LD = LOAD;
XOUTG.LD = LOAD;
XOUTH.LD = LOAD;
XOUTI.LD = LOAD;
XOUTJ.LD = LOAD;
YOUTA.LD = LOAD;
YOUTB.LD = LOAD;
YOUTC.LD = LOAD;
YOUTD.LD = LOAD;
YOUTE.LD = LOAD;
YOUTF.LD = LOAD;
YOUTG.LD = LOAD;
YOUTH.LD = LOAD;
YOUTI.LD = LOAD;
YOUTJ.LD = LOAD;
```

Figure 1. SNAP Files (3 of 3)