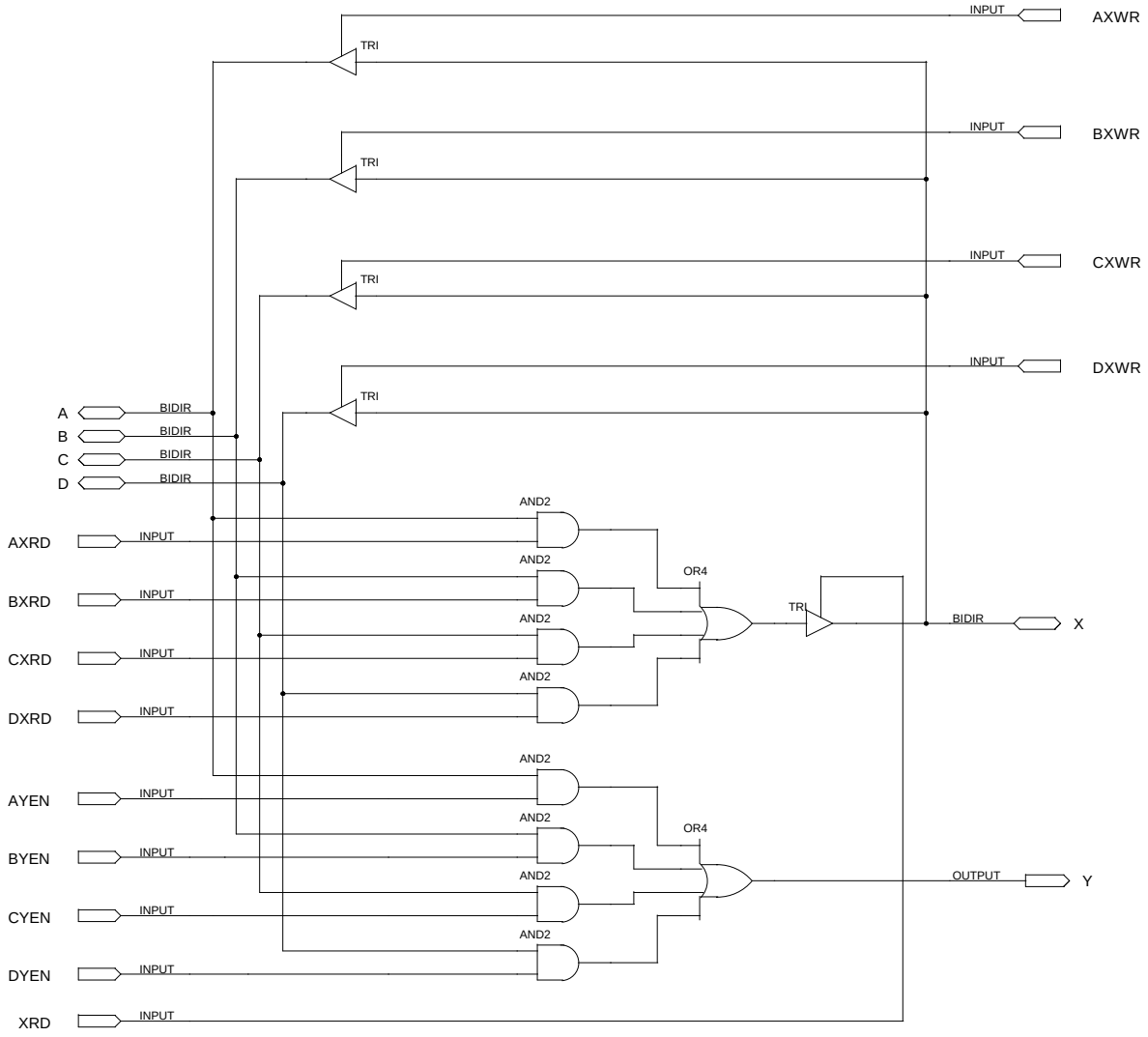
```
%*****************************************************************************
2 x 4 x 16 bit Crossbar Switch for Data
6 Feb 02 pak
*****************************************************************************%

TITLE "2x4x16 Data Crossbar";

INCLUDE "onebit";

SUBDESIGN datacrossbar
(/xrd       : INPUT;        % tristate enable %
 axwr       : INPUT;        % Select inputs %
 bxwr       : INPUT;
 cxwr       : INPUT;
 dxwr       : INPUT;
 axrd       : INPUT;
 bxrd       : INPUT;
 cxrd       : INPUT;
 dxrd       : INPUT;
 /cnl0      : INPUT;
 /cnl1      : INPUT;
 /cnl2      : INPUT;
 /cnl3      : INPUT;
 ad[15..0]  : BIDIR;        % Data bus ports %
 bd[15..0]  : BIDIR;
 cd[15..0]  : BIDIR;
 dd[15..0]  : BIDIR;
 xd[15..0]  : BIDIR;
 yd[15..0]  : OUTPUT;
)

VARIABLE
    switch[15..0]   : onebit;       % Sixteen switch cells %

BEGIN
    switch[15..0].xrd = !(/xrd);        % Invert /XRD and apply to all switches %
    switch[15..0].axrd = axrd;      % Apply AXRD to all switches %
    switch[15..0].bxrd = bxrd;      % Apply BXRD to all switches %
    switch[15..0].cxrd = cxrd;      % Apply CXRD to all switches %
% The DSP never reads the D-port. A FIFO drives
    these lines directly during DMA %
    switch[15..0].dxrd = B"0";
    switch[15..0].axwr = axwr;      % Apply AXWR to all switches %
    switch[15..0].bxwr = bxwr;      % Apply BXWR to all switches %
    switch[15..0].cxwr = cxwr;      % Apply CXWR to all switches %
% The DSP writes the D-port directly using its external memory interface. A FIFO drives
    these lines directly during DMA %
    switch[15..0].dxwr = B"0";
    switch[15..0].ayen = !/cnl0;    % Apply AYEN to all switches %
    switch[15..0].byen = !/cnl1;    % Apply BYEN to all switches %
    switch[15..0].cyen = !/cnl2;    % Apply CYEN to all switches %
    switch[15..0].dyen = !/cnl3;    % Apply DYEN to all switches %

    ad[15..0] = switch[15..0].a;    % Wire up slave port data buses %
    bd[15..0] = switch[15..0].b;
    cd[15..0] = switch[15..0].c;
    dd[15..0] = switch[15..0].d;
    xd[15..0] = switch[15..0].x;
    yd[15..0] = switch[15..0].y;

END;
```

```
--
--  Copyright (C) 1988-2000 Altera Corporation
--  Any megafunction design, and related net list (encrypted or decrypted),
--  support information, device programming or simulation file, and any other
--  associated documentation or information provided by Altera or a partner
--  under Altera's Megafunction Partnership Program may be used only to
--  program PLD devices (but not masked PLD devices) from Altera.  Any other
--  use of such megafunction design, net list, support information, device
--  programming or simulation file, or any other related documentation or
--  information is prohibited for any other purpose, including, but not
--  limited to modification, reverse engineering, de-compiling, or use with
--  any other silicon devices, unless such use is explicitly licensed under
--  a separate agreement with Altera or a megafunction partner.  Title to
--  the intellectual property, including patents, copyrights, trademarks,
--  trade secrets, or maskworks, embodied in any such megafunction design,
--  net list, support information, device programming or simulation file, or
--  any other related documentation or information provided by Altera or a
--  megafunction partner, remains with Altera, the megafunction partner, or
--  their respective licensors.  No other licenses, including any licenses
--  needed under any third party's intellectual property, are provided herein.
--
CHIP datacrossbar
BEGIN
    |/cnl3 :    INPUT_PIN = 9;
    |xd0 :  BIDIR_PIN = 10;
    |yd0 :  OUTPUT_PIN = 11;
    |ad0 :  BIDIR_PIN = 12;
    |bd0 :  BIDIR_PIN = 14;
    |cd0 :  BIDIR_PIN = 15;
    |dd0 :  BIDIR_PIN = 16;
    |xd1 :  BIDIR_PIN = 18;
    |yd1 :  OUTPUT_PIN = 19;
    |ad1 :  BIDIR_PIN = 21;
    |bd1 :  BIDIR_PIN = 22;
    |cd1 :  BIDIR_PIN = 23;
    |dd1 :  BIDIR_PIN = 25;
    |xd2 :  BIDIR_PIN = 27;
    |yd2 :  OUTPUT_PIN = 28;
    |ad2 :  BIDIR_PIN = 29;
    |bd2 :  BIDIR_PIN = 30;
    |cd2 :  BIDIR_PIN = 31;
    |dd2 :  BIDIR_PIN = 32;
    |xd3 :  BIDIR_PIN = 34;
    |yd3 :  OUTPUT_PIN = 35;
    |ad3 :  BIDIR_PIN = 36;
    |bd3 :  BIDIR_PIN = 37;
    |cd3 :  BIDIR_PIN = 38;
    |dd3 :  BIDIR_PIN = 39;
    |xd4 :  BIDIR_PIN = 40;
    |yd4 :  OUTPUT_PIN = 41;
    |ad4 :  BIDIR_PIN = 42;
    |bd4 :  BIDIR_PIN = 43;
    |cd4 :  BIDIR_PIN = 44;
    |dd4 :  BIDIR_PIN = 45;
    |xd5 :  BIDIR_PIN = 46;
    |yd5 :  OUTPUT_PIN = 47;
    |ad5 :  BIDIR_PIN = 48;
    |bd5 :  BIDIR_PIN = 49;
    |cd5 :  BIDIR_PIN = 53;
    |dd5 :  BIDIR_PIN = 54;
    |xd6 :  BIDIR_PIN = 55;
    |yd6 :  OUTPUT_PIN = 56;
    |ad6 :  BIDIR_PIN = 60;
    |bd6 :  BIDIR_PIN = 61;
    |cd6 :  BIDIR_PIN = 62;
    |dd6 :  BIDIR_PIN = 63;
```

```
|xd7 :  BIDIR_PIN = 65;
|yd7 :  OUTPUT_PIN = 66;
|ad7 :  BIDIR_PIN = 67;
|bd7 :  BIDIR_PIN = 68;
|cd7 :  BIDIR_PIN = 69;
|dd7 :  BIDIR_PIN = 70;
|xd8 :  BIDIR_PIN = 71;
|yd8 :  OUTPUT_PIN = 72;
|ad8 :  BIDIR_PIN = 74;
|bd8 :  BIDIR_PIN = 75;
|cd8 :  BIDIR_PIN = 78;
|dd8 :  BIDIR_PIN = 79;
|xd9 :  BIDIR_PIN = 80;
|yd9 :  OUTPUT_PIN = 81;
|ad9 :  BIDIR_PIN = 82;
|bd9 :  BIDIR_PIN = 83;
|cd9 :  BIDIR_PIN = 84;
|dd9 :  BIDIR_PIN = 86;
|xd10 : BIDIR_PIN = 87;
|yd10 : OUTPUT_PIN = 88;
|ad10 : BIDIR_PIN = 90;
|bd10 : BIDIR_PIN = 91;
|cd10 : BIDIR_PIN = 92;
|dd10 : BIDIR_PIN = 93;
|xd11 : BIDIR_PIN = 96;
|yd11 : OUTPUT_PIN = 97;
|ad11 : BIDIR_PIN = 98;
|bd11 : BIDIR_PIN = 99;
|cd11 : BIDIR_PIN = 100;
|dd11 : BIDIR_PIN = 101;
|xd12 : BIDIR_PIN = 102;
|yd12 : OUTPUT_PIN = 103;
|ad12 : BIDIR_PIN = 106;
|bd12 : BIDIR_PIN = 107;
|cd12 : BIDIR_PIN = 108;
|dd12 : BIDIR_PIN = 109;
|xd13 : BIDIR_PIN = 110;
|yd13 : OUTPUT_PIN = 111;
|ad13 : BIDIR_PIN = 112;
|bd13 : BIDIR_PIN = 113;
|cd13 : BIDIR_PIN = 116;
|dd13 : BIDIR_PIN = 117;
|xd14 : BIDIR_PIN = 118;
|yd14 : OUTPUT_PIN = 119;
|ad14 : BIDIR_PIN = 120;
|bd14 : BIDIR_PIN = 121;
|cd14 : BIDIR_PIN = 122;
|dd14 : BIDIR_PIN = 131;
|xd15 : BIDIR_PIN = 132;
|yd15 : OUTPUT_PIN = 133;
|ad15 : BIDIR_PIN = 134;
|bd15 : BIDIR_PIN = 136;
|cd15 : BIDIR_PIN = 137;
|dd15 : BIDIR_PIN = 138;
|/cnl0 :    INPUT_PIN = 141;
|/cnl1 :    INPUT_PIN = 1;
|/cnl2 :    INPUT_PIN = 139;
|axrd : INPUT_PIN = 5;
|bxrd : INPUT_PIN = 2;
|cxrd : INPUT_PIN = 140;
|dxrd : INPUT_PIN = 143;
|/xrd : INPUT_PIN = 128;
|dxwr : INPUT_PIN = 6;
|cxwr : INPUT_PIN = 142;
|bxwr : INPUT_PIN = 8;
|axwr : INPUT_PIN = 126;
```



Note: The DMAC (Y-port) does not write to slaves

TITLE	2X4 DATA SWITCH, ONE BIT		
COMPANY	Fermilab		
DESIGNER	Accelerator Controls		
SIZE C	NUMBER 1.00	REV A	
DATE 10:37a 2-28-2002		SHEET 1	OF 1