```
//Correct mux connections for "Grey" coding
//Rev. A - 12/07/00 - First "Official" Release
//Rev. B - 05/31/01 - Add Voltage Parameter
//Rev. C - 06/25/01 - Split Bus at D10 for software bug

DESIGN databuf;

PART ispGDX80VA-5T100I;

PARAM VOLTAGE VCC;

////////////////////////////////////Reset Config
Word////////////////////////////////////
//Bit Number      0    1    2    3    4    5    6    7    8    9    10   11   12   13   14
15 ;
SET ResetWord_a [GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND,
GND,
                 GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND];
//Bit Number      16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
31 ;
////////////////////////////////////////////////////////////////////////////////////////////
////

////////////////////////////////////Reset Config
Word////////////////////////////////////
SET ResetWord_b                                                                  [GND, GND, GND, GND,
GND];
//Bit Number      16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
31 ;
////////////////////////////////////////////////////////////////////////////////////////////
////

//Define Bus Names
SET ProcData_a  [ProcData0..ProcData26];
SET ProcData_b  [ProcData27..ProcData31];
SET BuffData_a  [BuffData0..BuffData26];
SET BuffData_b  [BuffData27..BuffData31];

//Define Bus Pin-outs
BIDI    ProcData_a  {A0..A15, B0..B10};
BIDI    ProcData_b  {B11..B15};
BIDI    BuffData_a  {C0..C15, D0..D10};
BIDI    BuffData_b  {D11..D15};

//Define other input pin-outs
INPUT   nH_RESET    {A18}; //s0
INPUT   nXOE        {A19}; //s1
INPUT   RnW         {B18}; //s0
INPUT   nReadReset  {B19}; //s1

//Define dummy I/Os used to generate OEs based on direction input
OUTPUT  DIR         {C16}; //**
BIDI    OE_P        {C17}; //oe
BIDI    OE_B        {D17}; //oe


BEGIN

//Create mux logic to generate DIRECTION signal
    DIR.s0 = RnW;
    DIR.s1 = nXOE;

    DIR.m0 = GND;
    DIR.m1 = VCC;
    DIR.m2 = GND;
    DIR.m3 = GND;

//Create mux logic to generate OE for Buffered bus
    OE_B.oe = VCC;

    OE_B.s0 = RnW;
    OE_B.s1 = nXOE;
```

```
    OE_B.m0 = VCC;
    OE_B.m1 = GND;
    OE_B.m2 = GND;
    OE_B.m3 = GND;

//Create mux logic to generate OE for Processor bus
    OE_P.oe = VCC;

    OE_P.s0 = nH_RESET;
    OE_P.s1 = nReadReset;

    OE_P.m0 = VCC;
    OE_P.m1 = VCC;
    OE_P.m2 = DIR.a;
    OE_P.m3 = VCC;

//Mux for Processor Data Bus
    ProcData_a.oe = OE_P;

    ProcData_a.s0 = nH_RESET;
    ProcData_a.s1 = nReadReset;

    ProcData_a.m0 = ResetWord_a;
    ProcData_a.m1 = ResetWord_a;
    ProcData_a.m2 = BuffData_a;
    ProcData_a.m3 = ResetWord_a;

//Mux for Processor Data Bus
    ProcData_b.oe = OE_P;

    ProcData_b.s0 = nH_RESET;
    ProcData_b.s1 = nReadReset;

    ProcData_b.m0 = ResetWord_b;
    ProcData_b.m1 = ResetWord_b;
    ProcData_b.m2 = BuffData_b;
    ProcData_b.m3 = ResetWord_b;



//Mux for Buffered Data Bus
    BuffData_a.oe = OE_B;

    BuffData_a.m0 = ProcData_a;

//Mux for Buffered Data Bus
    BuffData_b.oe = OE_B;

    BuffData_b.m0 = ProcData_b;

END
```