GAL®16VP8/20VP8: Bus Arbitration Circuit

Introduction

Lattice Semiconductor's GAL16VP8 and GAL20VP8 devices combine the programmable logic flexibility of the industry standard GAL16V8 and GAL20V8 devices with the high output drive capability of the 74240 series of TTL devices. The GAL16VP8 and GAL20VP8 devices have been designed to implement bus and memory interface logic as a one chip solution, as opposed to the historical two chip solution.

The addition of 64mA output drive capability, and the option to individually configure the outputs either as standard totem pole outputs or open-drain outputs, has taken these 16/20VP8 devices beyond the realm of common glue logic integration. These capabilities allow the devices to drive heavy capacitive loads in applications such as bus and memory address and control signal drivers.

The following bus arbitration design example takes full advantage of the bus driving capability and the open-drain option of the GAL16VP8 to implement the bus arbitration circuit.

Design Example

A bus arbitration circuit is used to determine which board connected to the system bus gets control of the bus for data transfers. One of the most common methods of arbitrating the bus is to assign a priority level to each board and to award bus ownership to the board with the highest priority request. This normally requires a combination of a priority encoder and decoder logic to determine the priority. The GAL16VP8 highlighted in this design example uses wired-OR logic on the bus, implemented with open-drain outputs, to determine the relative priority, eliminating the use of dedicated priority encoder/decoder logic. This scheme is similar to the one used in IBM's Micro Channel bus standard.

Using this scheme, the board with the lowest numeric value ID has the highest priority — 0000 being the highest priority and 1111 being the lowest priority. Priority is resolved between competing boards by making the arbitration outputs (ARB3-ARB0) and bus request signals (BREQ) open-drain with external pull-up resistors.

A typical bus arbitration cycle begins by detection of an active BREQ signal driven by all requesting boards. (If BREQ is inactive, it means that none of the boards on the bus has requested the bus.) The requesting boards drive the ARB3-ARB0 signals according to the predetermined priority that is assigned by the ID3-ID0 inputs. The bit-by-bit resolution of the arbitration begins with a comparison at ARB3. When comparing the ARB3 bit, a logic low on ARB3 will prevail over a logic high on ARB3 since the bus has a wired-OR structure.

The comparison continues with ARB2, which compares ARB3 from the previous stage as well as the prevailing ARB2. The requesting boards that do not match the priority driven on ARB3 will no longer drive ARB2-ARB0. For example, if ARB3=0 on the bus but a board has ID3=1, the board will no longer drive the low order bits (ARB2-ARB0). This process continues until ARB0 is reached. After resolving ARB0, the lowest numerical value (highest priority) will be driven on ARB3-ARB0. A local bus grant signal is generated from the result of the arbitration if the value on ARB3-ARB0 matches that on the ID3-ID0 inputs to a given board. The board winning the bus keeps BREQ active until its bus access is complete.

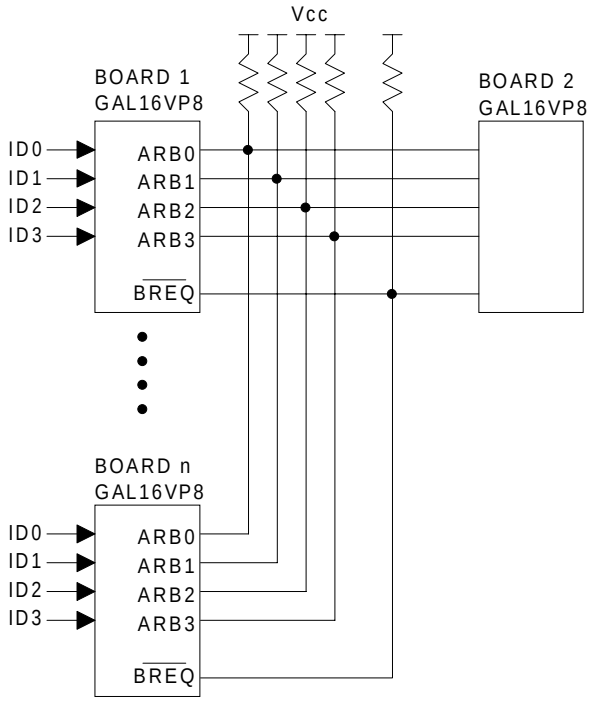
An example CUPL source file which implements the arbitration logic is shown on the following pages. Figure 1 illustrates a typical bus interface block diagram. As a footnote to the open-drain configuration, use the FUSES statement and the fuse numbers provided in the datasheet to implement open-drain outputs in ABEL.

GAL16VP8/20VP8: Bus Arbitration Circuit

Summary

The unique ability for a logic device to drive 64mA and the option to have open-drain outputs have made the GAL16VP8 and GAL20VP8 ideal for bus and memory interface logic. The GAL16/20VP8 devices allow the design engineer a one chip solution for complex memory or bus applications that require 64mA Iol. The one chip solution reduces power and cost, while increasing speed and reliability.

Figure 1. Typical System Interface



Listing 1. Example of a CUPL Source File

```
Name            VP8ARB;
Partno          U00;
Device          G16VP8MA;
Revision        1.0;
Date            09/09/99;
Designer        John Doe;
Company         Lattice Semiconductor Corp.;
Location        Hillsboro, OR;
Assembly        16VP8 bus arbitration circuit;

/*************************************************************/
/*  This bus arbitration circuit is based on the priority     */
/*  assigned to the board.  ID3..0 is configured to the board */
/*  priority code.  This example uses ID 3..1=0111 for the    */
/*  priority code.  ARB3..0 is driven to the same priority   */
/*  code.                                                     */
/*************************************************************/

/** Dedicated Inputs definition**/

Pin 1  = ID3;  /* priority ID must be driven to ....     */
Pin 2  = ID2;  /* ... the appropriate priority levels ...*/
Pin 3  = ID1;  /* ... by the local board.                */
Pin 4  = ID0;
/** Note Pin 5 is Vcc on this package **/
Pin 6  = LARB; /* Local arbitrate signal. Enables local board arbitration */



/** I/O pin definitions **/

Pin 17 = BREQ;  /* active low open-drain bus request */
Pin 12 = ARB3;  /* open-drain arbitration bits */
```

```
Pin 13 = ARB2;
Pin 14 = ARB1;
Pin 16 = ARB0;
/** Note Pin 15 is GND on this package **/

/** Output Equations **/

!BREQ = !ARB3 & ARB2 & ARB1 & ARB0;
 BREQ.TEC = 'b'0;   /* .TEC=0 specifies the open-drain output..TEC=1 or
                    the default is the totem pole output. */
 BREQ.OE = LARB;    /* Enables local board arbitration */

!ARB3 = !ARB3 & !BREQ
      # ID3 & BREQ;    /* arbitration bit 3 */
 ARB3.TEC = 'b'0;
 ARB3.OE  = LARB;   /* Enables local board arbitration */

 ARB2 = !ARB3 & ARB2 & !BREQ
      # ID2 & BREQ;     /* arbitration bit 2 */
 ARB2.TEC = 'b'0;
 ARB2.OE  = LARB;   /* Enables local board arbitration */

 ARB1 = !ARB3 & ARB2 & ARB1 & !BREQ
      # ID1 & BREQ;     /* arbitration bit 1 */
 ARB1.TEC = 'b'0;
 ARB1.OE  = LARB;   /* Enables local board arbitration */

 ARB0 = !ARB3 & ARB2 & ARB1 & ARB0 & !BREQ
      # ID0 & BREQ;     /* arbitration bit 0 */
 ARB0.TEC = 'b'0;
 ARB0.OE  = LARB;   /* Enables local board arbitration */
```