Overview

- Bus Arbitration
- Switches

Bus Masters and Slaves

- A bus master is either a CPU or a hardware component (DMA Controller for instance) that controls the buses.
- A bus slave is a device that takes its orders from the bus master.
- What if two or more bus masters want to control the buses simultaneously?
  - Bus arbitration is required.

Bus Arbitration

- Daisy chain bus arbitration.
- Hardware priority bus arbitration.
Daisy Chain Bus Arbitration

• Whenever a device wants to control the bus, it asserts HOLD and opens the switch in the HOLDA (HOLD_ACKNOWLEDGE) line.
• When HOLDA is asserted by the CPU, it passes through each of the inactive one.
• If a bus master farther right on the chain asserts HOLD before another master is finished, HOLDA is not passed along until the higher priority device (closer to CPU) is finished with its tasks and closes its switch.

Hardware Priority Bus Arbitration

• Each device is pre-assigned a priority.
• Simultaneous HOLD signals are encoded so that only the highest priority device receives HOLDA.
Input Switches

- Most basic of all binary input devices.
- The switch output is high or low depends on the switch position.
- Pull-up resistors are necessary in each switch to provide a high logic level when the switch is open.
- Problem with switches:
  - Switch bounce.
    - When a switch makes contact, its mechanical springiness will cause the contact to bounce, or make and break, for a few millisecond (typically 5 to 10 ms).

Software Debouncing

Two software debouncing approaches:

- Wait and see:
  - If the software detects a low logic level, indicating that switch has closed, it simply waits for longer than 10 ms, say 20 to 100ms, and then test for the switch still being low.
- Counter-based approach:
  - Initialize a counter to 10.
  - Poll the switch every millisecond until the counter is either 0 or 20. If the switch output is low, decrement the counter; otherwise, increment the counter.
  - If the counter is 0, we know that switch output has been low for at least 10 ms. If, on the other hand, the counter reaches 20, we know that the switch has been closed for at least 10 ms.
NOR Latch Debouncer

Integrating Debouncer with Schmitt Trigger

One-Dimensional Array of Switches

- Switch bounce problem must be solved.
- The array of switches must be scanned to find out which switches are closed or open.
  - Software is required to scan the array. As the software outputs a 3-bits sequence from 000 to 111, the multiplexer selects each of the switch inputs. The software scanner then read one bit at an input port.
- The output of switch array could be interfaced directly to an eight-bit port at point A.
- To save I/O lines, a 74LS151 8_Inpu Multiplexer can be used.
Keyboard Matrix of Switches (Cont.)

- A keyboard is an array of switches arranged in a two-dimensional matrix.
- A switch is connected at each intersection of vertical and horizontal lines to the vertical.
- Closing the switch connects the horizontal line to the vertical line.
- 8*8 keyboard can be interfaced directly into 8-bit output and input ports at point A and B.
- Some input and output lines can be saved by using a 74LS138 3-of-8 decoder and a 74LS151 8-Input Multiplexer.

Ghosting

- Software can scan the keyboard by outputting a three-bit code to 74LS138 and then scanning the 74LS151 multiplexer to find the closed switch.
  - The combination of the two 3-bit scan codes identifies which switch is closed. For example, the code 00000 scan switch 00 in the upper left-hand corner.
- The diode prevents a problem called ghosting.
Ghosting (Cont.)

- Ghosting occurs when several keys are pushed at once.
- Consider the case shown in the figure where three switches 01, 10 and 11 are all closed. Column 0 is selected with a logic low and assume that the circuit does not contain the diodes. As the rows are scanned, a low is sensed on Row 1, which is acceptable because switch 10 is closed. In addition, Row 0 is seen to be low, indicating switch 00 is closed, which is NOT true. The diodes in the switches eliminate this problem by preventing current flow from R1 through switches 01 and 11. Thus Row 0 will not be low when it is scanned.

Reading