CS 397 -- Computer Systems Architecture Laboratory (Software and Hardware)

IBM PC-AT Industry Standard Architecture (ISA) Bus Information

The ISA bus is a popular bus architecture used in IBM and compatible personal computer systems. It has evolved during the years since its introduction in the original IBM PC. The particular implementation with which you need to be familiar is the IBM PC-AT version of the bus. Most contemporary IBM and compatible personal computers come equipped with this bus. This version of the ISA bus runs at 8MHz and has a sixteen bit data path. This document should provide you with enough details about the ISA bus to implement the lab exercises.

ISA Bus Cycles

There are three types of bus cycles on the ISA bus. Each type of bus cycle can be used to access either a memory mapped device or an I/O device. In this course, you need only be concerned with memory mapped devices. The three types of bus cycles are *no-wait-state*, *standard*, and *ready*. In addition, the ISA bus protocol allows for peripheral devices to own the bus and initiate bus accesses. However, *no-wait-state* cycles can only be executed when the CPU owns the bus. Most bus cycles are *standard* cycles which require three clock periods for sixteen bit transfers and six clock periods for eight bit transfers. The *no-wait-state* cycle is a shortened bus cycle which requires two clock periods for sixteen bit transfers and three clock periods for eight bit transfers. The *ready* cycle is a standard cycle that can be lengthened by the memory resource if it cannot respond to the access in a timely fashion. In this course, we will concern ourselves only with the *standard* bus cycle.

ISA Bus Implementation

The ISA bus uses an asynchronous protocol to transfer data between system components. However, two clock signals are also available on the bus (*BCLK* and *OSC*). Due to the relatively high frequency of the *OSC* signal (compared to the ISA bus frequency), it is possible to implement the ISA bus protocol synchronously. You will use the *OSC* signal as the clock input for your synchronous ISA bus interface. Timing diagrams for *standard* ISA bus cycles are included in this document. Your interface must respond correctly to the access cycles detailed in the diagrams.

ISA Bus Signals

The ISA bus has many more signals than those described below. However, you need only be concerned with the following signals.

OSC - Oscillator

The *OSC* signal line is driven by the system. It is a 14.32 MHz clock with a 50% duty cycle.

BALE - Bus Address Latch Enable

The *BALE* signal line is driven by the CPU. During a CPU mastered bus cycle, an active *BALE* signal indicates that valid data resides on the *LA/SA* signal lines and the *SBHE** signal line. To ensure proper address decoding, the most significant four bits of the address (*LA*) should be latched with the *BALE* signal because these signal lines do not contain valid data for the entire bus cycle. The *BALE* signal is driven active when a peripheral owns the bus. Therefore, during a peripheral mastered bus cycle, the *BALE* signal is not important.

LA<17-23> - Logical Address

The *LA* signal lines are driven by the ISA bus master. These address signal lines extend the address space to sixteen megabytes. The *LA<17-19>* signal lines are identical to the *SA<17-19>* signal lines and need not be of concern. During a CPU mastered bus cycle, these signals are not driven for the entire bus cycle. Therefore, these signals must be latched with the *BALE* signal so that they can be decoded later in the cycle. During a peripheral mastered bus cycle, these signals are driven for the entire bus cycle.

MEMCS16* - Memory Chip Select 16

The *MEMCS16** signal line is driven by the memory resource. This signal indicates that the memory resource supports a sixteen bit data access cycle. However, the bus master may still request an eight bit access. The *SBHE** and *SA<0>* signal lines must be decoded to determine the size of the access.

SA<0-19 > - System Address

The *SA* signal lines are driven by the ISA bus master. These address signal lines are capable of addressing one megabyte of memory space and are driven for the entire bus cycle.

SBHE* - System Byte Enable High

The *SBHE** signal line is driven by the ISA bus master. The *SBHE** signal indicates that valid data resides on the *SD<8-15>* signal lines. This signal and the *SA<0>* signal are responsible for determining the data size of a bus access on a sixteen bit peripheral. For a sixteen bit access to occur, *SBHE** must be driven active and *SA<0>* must be driven inactive (indicating an even half-word boundary). Any other combination of the two signals results in an eight bit access.

MEMR* - Memory Read

The *MEMR** signal line is driven by the ISA bus master. An active *MEMR** signal is a request for a memory resource to drive the data bus during the bus cycle.

MEMW* - Memory Write

The *MEMW** signal line is driven by the ISA bus master. An active *MEMW** signal is a request for a memory resource to accept the data from the data bus during the bus cycle.

SD<0-15> - System Data

The *SD* signal lines can be driven by the ISA bus master and the slave peripheral being accessed. During an eight bit access cycle, only lines *SD<0-7>* are driven. During a sixteen bit access cycle, all of the signal lines (*SD<0-15>*) are driven.

* - An asterisk immediately following the signal name indicates that the signal is an active low signal. The active low signals are *MEMCS16**, *MEMR**, *MEMW**, and *SBHE**. All other signals are active high.

Description of Waveforms

The functional portion of a CPU mastered standard bus cycle begins when the *BALE* signal is pulsed high by the CPU. At this point, the memory resource must latch the high order address bits (*LA*). The memory resource should also activate the *MEMCS16** signal indicating to the CPU that it is capable of sixteen bit accesses. The CPU then drives the appropriate command line (*MEMR** or *MEMW**)active indicating the type of access. The *SBHE** and *SA<0>* signals must be decoded (as described in the signal description section) to determine if a sixteen bit access has been requested. Depending on the type of access, the memory resource must either drive the data lines with the requested data (read) or store the data driven onto the data lines by the CPU (write). Remember that only the low order eight bits of data are valid if an eight bit access is in progress. This marks the completion of a standard bus cycle.

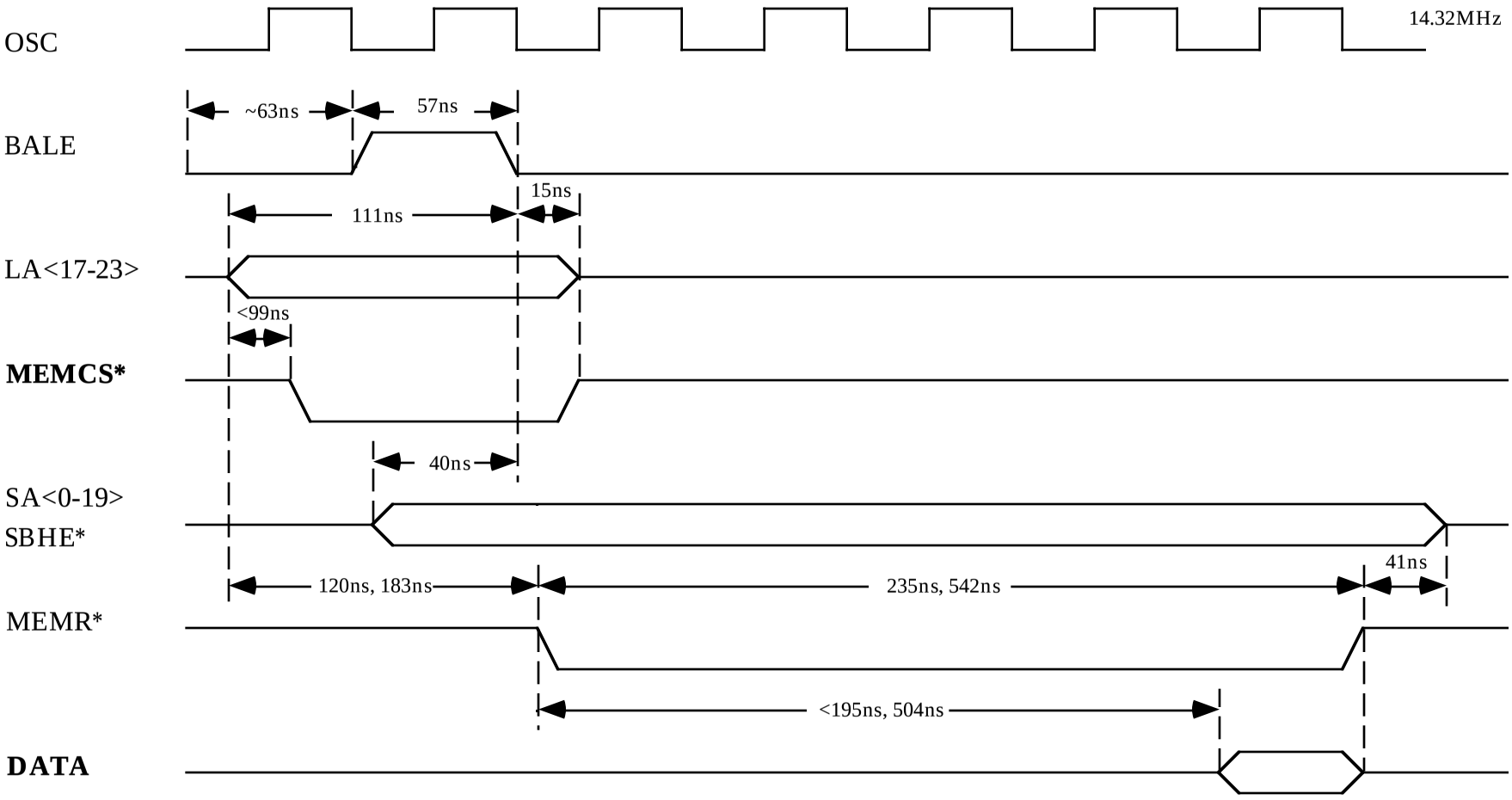
Notes on Waveforms

1. The times shown in the ISA bus waveform figure are typically the minimum times required to meet the specification. There are, however, three exceptions. First, the time from the beginning of the bus cycle to the assertion of *BALE* is only an estimate. Second, the time between the assertion of *LA<17-23>* and the assertion of *MEMCS16** is a maximum time. *MEMCS16** must be asserted within 99ns of the assertion of *LA<17-23>* for a sixteen bit access cycle to occur. Third, the time between the assertion of MEMR* and the DATA in a read access is a maximum time. The data must be valid within the specified time for the CPU to read the data correctly.

2. When there are two values shown for a given measurement, the first value represents the time for a sixteen bit cycle and the second value represents the time for an eight bit cycle. If only one value is present, then it is valid for both eight and sixteen bit cycles.

3. If the signal name is shown in bold lettering, then it means that the memory resource (your design) is responsible for driving the signals.

CPU Mastered Standard Read Cycle



CPU Mastered Standard Write Cycle