Harris Semiconductor

APPNOTE

Harris Digital

80C286/80386 Hardware Comparison

Author: Ted Schaufelberger

Introduction

The Harris 80C286 static CMOS microprocessor, available with maximum operating frequencies of 16MHz and 20MHz, offers both performance and design advantages over the 80386 when operating at the same frequency. When both the 80C286 and 80386 are operated on a 16-bit data bus, which fully supports industry standard 8086/80286 code, the 80C286 has better performance, and is significantly simpler to design with than the 80386. The 80C286 also uses significantly lower power than the 80386. The 80C286 also uses significantly lower power than the 80386, leading to less expensive, more reliable overall system design (See Figure 1).
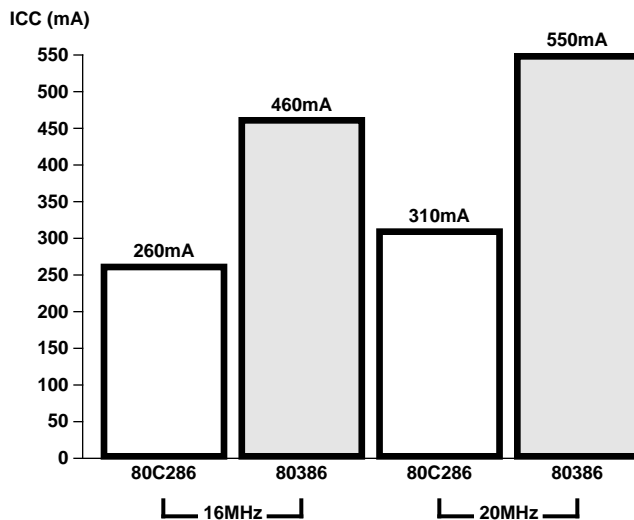

FIGURE 1. 80C286/80386 POWER CONSUMPTION COMPARISON

The following comparison highlights some of the performance advantages that exist on the 80C286:

Summary of 16-Bit Data Bus Performance

80C286

- The 80C286 already has all necessary control lines needed to implement a 16-bit data bus (Ref. section on Control Signals Required to Implement a 16-Bit Data Bus).

- The 80C286 supports a fully pipelined mode of operation for maximum system performance (Ref. section on Pipelined Operation on a 16-Bit Data Bus).

- The 80C286 remains in a pipelined mode of operation even when idle bus cycles occur (Ref. section on Idle Cycles).

- The 80C286 instruction prefetch takes one bus cycle to execute, thereby minimizing the time that the processor Execution Unit must wait should it need the bus (Ref. section on Instruction Prefetching a 16-Bit Data Bus).

80386

- The 80386 requires five additional control lines to be generated by external logic in order to implement a 16-bit data bus (Ref. section on Control Signals Required to Implement 16-Bit Data Bus).

- The 80386 does not support a fully pipelined mode of operation. Some pipelining can be achieved, but to accomplish this, external bus 'monitor' logic must be added to the system (Ref. section on Pipelined Operation on a 16-Bit Data Bus).

- The 80386's pipelining is disrupted by idle bus cycles. A non-pipelined bus cycle, usually with an additional wait state, must be executed before the 80386 can return to pipelined mode. Idle bus cycles occur an average of 9% of the time (Ref. section on Idle Cycles).

- The 80386 instruction prefetch takes two bus cycles to execute, which can cause performance degradation by forcing the Execution Unit of the processor to wait a full bus cycle for use of the bus in order to complete an instruction (Ref. section of Instruction Prefetching on a 16-Bit Data Bus).

16-Bit Data Bus Operation

This section will discuss the control signals required to implement a 16-bit data bus, as well as pipelined operation, idle cycles, and instruction prefetching on a 16-Bit Data Bus.

Control Signals Required to Implement a 16-Bit Data Bus

The 80C286 microprocessor has all the control lines needed to implement a 16-bit data bus resident on chip, no further control lines are required.

80386 SIGNALS				16-BIT BUS SIGNALS			
BE3#	**BE2#**	**BE1#**	**BE0#**	**A1**	**BHE#**	**BLE# (A0)**	**COMMENTS**
H*	H*	H*	H*	X	X	X	X -No Active Bytes
H	H	H	L	L	H	L	
H	H	L	H	L	L	H	
H	H	L	L	L	L	L	
H	L	H	H	H	H	L	
H*	L*	H*	L*	X	X	X	X - Not Contiguous Bytes
H	L	L	H	L	L	H	
H	L	L	L	L	L	L	
L*	H*	H*	L*	X	X	X	X - Not Contiguous Bytes
L*	H*	L*	H*	X	X	X	X - Not Contiguous Bytes
L*	H*	L*	L*	X	X	X	X - Not Contiguous Bytes
L	L	H	H	H	L	L	
L*	L*	H*	L*	X	X	X	X - Not Contiguous Bytes
L	L	L	H	L	L	H	
L	L	L	L	L	L	L	

NOTE: BLE# Asserted when D0-D7 of 16-Bit Bus is Active.
 BHE# Asserted when D8-D15 of 16-Bit Bus is Active.
 A1 Low for All Even Words; A1 High for All Odd Words.

 Key: X = Don't Care
 H = High Voltage Level
 L = Low Voltage Level
 * = A Non-Occurring Pattern of Byte Enables; Either None are Asserted, or the Pattern has Byte Enables Asserted
 for Non-Contiguous Bytes.

FIGURE 2. A1, BLE#, AND BHE# SIGNAL GENERATION TABLE

In order to implement a 16-bit data bus with the 80386 microprocessor, it is necessary to create at least the following five additional control signals: Address Line 1 (A1), Bus Low Enable (BLE#), Bus High Enable (BHE#), Bus Size 16-Bits (BS16#), and Next Address (NA#).

The first of these signals, A1, is an additional address line required to convert the granularity of the 80386's address space from double-word size entities (32-bit) to word size entities (16-bit). The second two signals, BLE# and BHE#, primarily serve as chip selects which enable the appropriate byte or bytes onto the 16-bit data bus. These three signals are generated from the four 80386 byte enables (BE0# - BE3#) as shown in Figure 2. The logic to implement these signals is shown in Figure 3.

In addition, to these three control signals generated from 80386 signals as outputs, two input control signals to the 80386 must be generated to external logic (BS16# and NA#)

BS16# is used to inform the 80386, on a cycle-by-cycle basis that a 16-bit bus size is to be used for data transfer. NA# is used to request that the 80386 put the next cycle address on the bus early, thereby pipelining that cycle.

The 80386, therefore, requires five additional control lines, three outputs and two inputs, in order to implement a 16-bit data bus. The generation of these control lines, in turn, requires additional 'glue' logic (which also introduces additional signal propagation delay, thereby reducing address access time available to the system), and finally, there is additional bus cycle 'monitor' logic necessary if the 16-bit data bus is to be pipelined for higher performance.
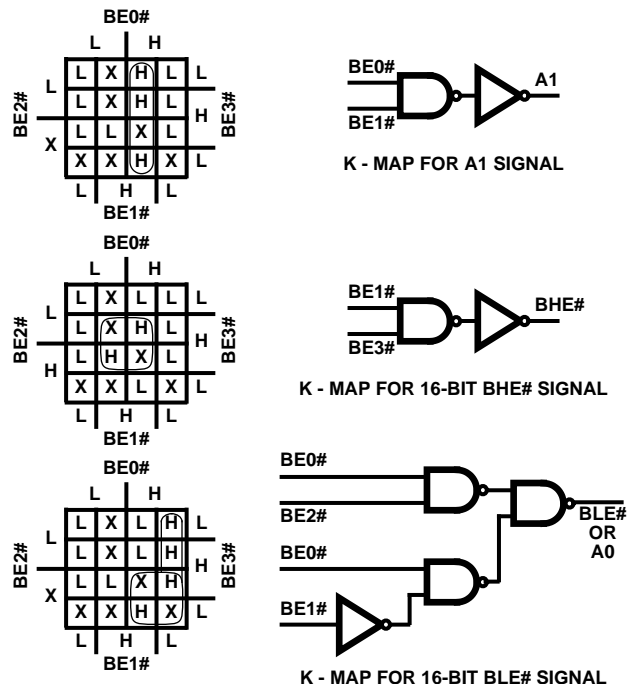

FIGURE 3. A1, BLE#, AND BHE# LOGIC

Pipelined Operation on a 16-Bit Data Bus

At a given clock frequency, pipelined address operation increases a system's performance, while simultaneously allowing relatively slower memories and I/O devices to be used. Pipelined address operation provides the system increased address access time, and increased address decoding time.

The 80C286 is optimized for, and directly supports, fully pipelined bus operations on a 16-bit data bus. In other words, the 80C286 performs all bus operations in a fully pipelined mode.

The 80386 does not support fully pipelined operation on a 16-bit data bus. In order to pipeline a bus cycle on the 80386, the Next Address (NA#) signal must be asserted to the processor. If the Next Address (NA#) signal and the Bus Size 16-Bit (BS16#) signal are both asserted in the same bus cycle, the NA# signal will not be recognized. Since, the BS16# signal must be asserted to the 80386 for many patterns of 16-bit and 8-bit transfers to take place correctly, the pipelining of transfers over a 16-bit bus is limited.

To allow pipelining of 16-bit data, external logic must be implemented to monitor the type of bus cycle taking place, decide if the cycle can be pipelined, and, if so, negate the BS16# signal to the 80386 and assert the NA# signal. Pipelining is possible only if the bus cycle is one of the following three types:

(1) A read operand cycle using only the lower half of the data bus.

(2) A write operand cycle using only the lower half of the data bus.

(3) A write operand cycle using only the upper half of the data bus.

The 80386 will not allow 16-bit pipelining of read or write cycles that have byte alignments that do not conform to one of the previously mentioned three types.

The 80C286, then, fully supports address pipelining, yielding the highest possible system performance, while using relatively lower performance (and therefore cheaper) memories and peripherals. The 80386, however, does not support fully pipelined 16-bit bus operation, and, to support even partial pipelining, requires external bus 'monitor' logic.

Idle Cycles

Another factor to consider when evaluating 80C286 and 80386 performance is the effect of idle cycles on pipelined operation. Calculations have shown that, on average, bus idle cycles occur in the system approximately 9% of the time. The effect of idle cycles on pipelining is quite different on the 80C286 that on the 80386.

The 80C286 pipelined operation is not affected by idle cycles. When an idle cycle or cycles occur in a stream of pipelined bus cycles, the 80C286 returns to pipelining bus cycles immediately after the last idle cycle. In this way, each device on the bus (e.g. memory, peripheral) maintains a fixed timing associated with that device, and therefore

always uses the minimum number of wait states required for that device.

On the other hand, the 80386 pipelined operation is disrupted by idle cycles. With the 80386, an idle cycle or cycles occurring in a pipelined stream of bus cycles breaks the pipelining operation. Once an idle cycle has occurred, a non-pipelined bus cycle must always be executed prior to resuming pipelining. Since a non-pipelined bus cycle will have different timing than pipelined bus cycle (even to the same device), an additional wait state must be added to this bus cycle. This not only degrades performance, but requires additional external logic to differentiate between a pipelined bus cycle access, and a non-pipelined bus cycle access, event to the same device with the same address.

From the preceding, it can be seen that **when executing 16-bit code, the 80C286 has a 9% performance increase over the 80386, due to the manner in which each processor handles idle cycles alone.** Note, that with the 80386, a pipelined stream of bus cycles will always be disrupted when an idle cycle occurs, whether using a 16-bit data bus or a 32-bit data bus. In either case, a non-pipelined bus cycle must be executed prior to resuming pipelined operation.

Instruction Prefetching on a 16-Bit Data Bus

One final factor needs to be considered in the evaluation of 80C286 and 80386 performance on a 16-bit data bus; the effect that prefetching instructions has on instruction execution time. Prefetching of instructions is done by the processor Bus Unit on both the 80C286 and 80386. The prefetch is done when the bus would otherwise be idle for the upcoming cycle, and the prefetch queue is not full.

The 80C286 does word size (16-bit) prefetching of instructions, and therefore completes it's prefetch activities in one bus cycle. This minimizes the waiting period to gain access to the bus by other processor entities, such as the Execution Unit.

The 80386 does doubleword (32-bit) prefetching of instructions, even on a 16-bit bus. This means that once a prefetch has begun execution, two bus cycles are required to complete the prefetch. If, for instance, the processor, Execution Unit requires the bus for a data fetch or write in order to complete an executing instruction, it must wait for the two bus cycles of prefetch to complete before it can access the bus. This can substantially degrade instruction execution time.

32-Bit Data Bus Operation

This section discusses operating the 80386 on a 32-bit bus in order to overcome some of the handicaps it suffers on a 16-bit bus. In addition, several advantages and disadvantages associated with the 80386 on a 32-bit bus are considered.

Hardware Advantages of the 80386 on a 32-Bit Data Bus

There are several advantages to operating the 80386 on a 32-bit data bus as opposed to a 16-bit data bus. Some of the control lines that were required for a 16-bit data bus are

eliminated (A1, BLE#, BHE#, and BS16#). It is possible to come closer to a fully pipelined mode of operation, although idle cycles will still disrupt the pipelining 9% of the time. Finally, prefetching on a 32-bit bus executes in one bus cycle instead of two. Offsetting these advantages, however are several major disadvantages.

Hardware Disadvantages of the 80386 on a 32-bit Data Bus

When using a full 32-bit data bus, the chip complexity of a 80386 based system is increased over a 16-bit system. Twice as many transceivers (four instead of two) are required. In addition, in order to accommodate the additional 16 data lines of the 32-bit bus, twice as many memory devices are typically required with the 32-bit system as compared to a 16-bit system. This amounts to an increase in DRAM, alone, of from 18 devices in a typical 16-bit system to 36 devices in a typical 32-bit system.

The 16 additional data lines of the 32-bit bus increase the EMI problems inherent in the system. The additional coupling and crosstalk between data lines must be taken into consideration when laying out the system PC board.

There is significant increase in the amount of board space used as result of the additional chips required to implement a 32-bit bus, as well as the 16 additional data lines. This results in a larger, more complex (and more expensive) PC board that with a 16-bit system, often requiring an increase in the number of board layers.

Reference

[1] "80386 Hardware Reference Manual", Intel Corporation, 1987.