The 80386SX Processor Bus

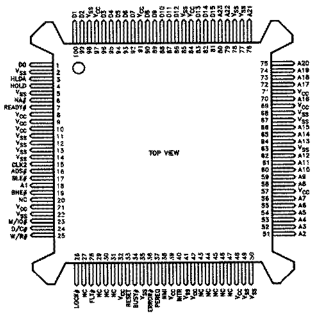
This lecture describes the signals and operation of the Intel 80386SX processor bus.
After this lecture you should be able to state the values of the processor bus signals described below during memory or I/O read and write cycles and during interrupt acknowledge cycles.

History

Intel's first 16-bit CPU was the 8086. A version of the 8086, the 8088, was used in the popular IBM PC and many later compatible machines. Intel's first 32-bit CPU was the 80386. It was designed to be backwards-compatible with the large amount of software which was available for the 8086 but extended the data and address registers to 32 bits and added support for memory protection and virtual memory. We will describe the processor bus of the 80386SX which is a version of the 80386 with a slightly simplified processor bus.

'386SX CPU Signals

The 386SX is packaged in a 100 pin package. It has a 24-bit address bus and a 16-bit data bus. The names of the signals are shown below:



Utility Bus

The utility bus includes the power, ground, and clock pins that are required for the processor to operate properly.

The most important pins on the CPU chip are for power supply and ground. The processor will operate erratically (or not at all) unless the power supply is held at the proper voltage regardless of the current drawn by the CPU.

The next most important signal is the clock, CLK2. Output signal transitions happen on the rising edge of CLK2 and inputs are also sampled on the rising edge of CLK2.

A data transfer over the bus is called a *bus cycle*. Each bus cycle requires two or more *processor cycles* (one T1 cycle plus one or more T2 cycles). Each of these processor cycles requires two CLK2 periods. Figure **??**[1] shows how two CLK2 cycles make up a processor cycle and how two processor cycles (T1 and T2) make up a bus cycles.

Exercise: A 386SX CPU is operating with a 25 MHz CLK2 signal. What is the CLK2 period? How long does a processor cycle take? How long does a bus cycle take?

Address and Data Busses

The 80386SX has a 16-bit data bus (D_{15} to D_0) and a 24-bit address bus (A_{23} to A_1). BHE* and BLE* (high and low byte enables) signals indicate to external circuits which byte(s) are to be transferred. The high byte is the one on D_{15} to D_8.

Endianness

Intel processors, unlike Motorola processors, use "little-endian" byte order. This means that 16- or 32-bit words are stored with the least-significant byte at the lowest-numbered address.

Exercise: The 16-bit word 1234H is to be stored at memory location 1FFH. What value will be stored at memory location 1FFH? At which address will the second byte be stored? Which byte enable(s) will be asserted? How many bus cycles will be required? What if the value 12345678H was to be stored at the same memory location? At location 100H? Write out your answer in the form of a table showing the values of the address bus, the data bus, BHE* and BLE*.

[1] From Intel i386SX data sheet.

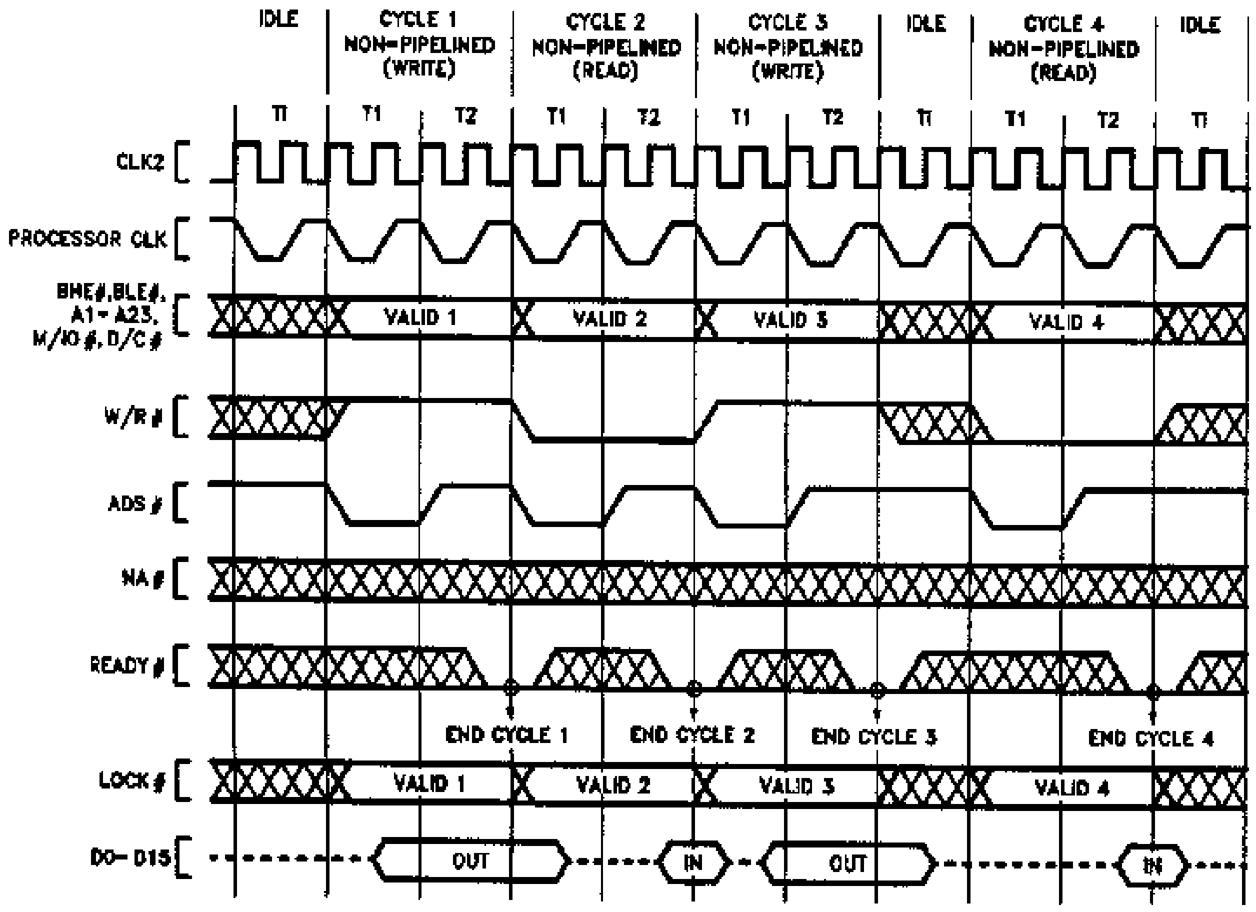

IDLE
CYCLE 1 NON-PIPELINED (WRITE)
CYCLE 2 NON-PIPELINED (READ)
CYCLE 3 NON-PIPELINED (WRITE)
IDLE
CYCLE 4 NON-PIPELINED (READ)
IDLE
Ti T1 T2 T1 T2 T1 T2 Ti T1 T2 Ti
CLK2
PROCESSOR CLK
BHE#,BLE#, A1-A23, M/IO#,D/C#
VALID 1
VALID 2
VALID 3
VALID 4
W/R#
ADS#
NA#
READY#
END CYCLE 1
END CYCLE 2
END CYCLE 3
END CYCLE 4
LOCK#
D0-D15
OUT
IN
OUT
IN

Figure 1: 80386SX Bus Cycles

Memory and I/O Address Spaces

The 80386 has separate I/O and memory address spaces. Different instructions are used to read/write these two address spaces (e.g. IN and OUT versus MOV). The I/O address space is limited to 64kB.

Bus Control

If the READY* input is not asserted at the end of a T2 processor cycle the 80386SX will generate an additional T2 cycle(s) (*"wait states"*).

The W/R* (write/read), D/C* (data/control), and M/IO* (memory/I/O) signals indicate the type of bus cycle being executed (read, interrupt acknowledge or write). The table below shows the possible bus cycles:

D/C*	M/IO*	W/R*	Bus Cycle
H	H	L	memory read
H	H	H	memory write
H	L	L	I/O read
H	L	H	I/O write
L	H	L	instruction fetch
L	L	L	interrupt acknowledge
L	H	H	halt

The ADS* output indicates that the address and the three above signals are valid.

Reset and Interrupts

The RESET input resets the processor state to a starting state in which the CPU will fetch the next instruction from memory location 0FFFFF0H. The memory at this location must therefore contain instructions that will restart the operating system.

The NMI and INTR inputs are used to generate non-maskable and maskable interrupts respectively.

Asserting the NMI input causes the processor to execute the interrupt handler pointed to by an interrupt vector stored in memory.

If interrupts are enabled then asserting INTR causes the CPU to carry out an interrupt acknowledge bus cycle which reads a 1-byte interrupt number from the bus (typically from an interrupt control chip). The corresponding interrupt vector is then fetched and the corresponding interrupt handler executed as with NMI.

In either case the current instruction is completed before the interrupt is recognized.

Other Signals

The ’386SX has a number of other signals which we will not cover at this time. For completeness, these are: HOLD and HOLDA (used by other devices to request the CPU to give up control of the processor bus by tri-stating all of its outputs), LOCK* (used to prevent other devices from requesting use of the processor bus), NA* (“next address” used to “pipeline” processor cycles), and PEREQ, BUSY*, and ERROR* (used to interface to a floating point co-processor).

80386SX Bus Cycles

Execution of each 80386SX instruction requires one or more bus cycles. Typically, this involves reading an instruction from memory possibly followed by transfers of data between the CPU and memory or I/O devices.

In addition to the read and write bus cycles from memory and I/O address space the CPU can also execute an interrupt acknowledge bus cycle and can be in an idle or halted mode.

Read and Write Cycles

The address and bus control signals go active at the start of the T1 processor cycle. During a write cycle the data bus is driven with the value to be written during the second half of T1.

At the end of each T2 cycle the processor checks the READY* input. If it is active, the bus cycle is terminated, otherwise and additional T2 cycle is run. These additional “wait states” are used to accommodate slow memory by increasing the time available between when the address is output and the time when the data is required. If the memory being designed into a system will require wait states a wait state generator circuit must be designed so that READY* is only asserted after two or more T2 states have elapsed.

The only difference between memory and I/O read/write cycles is that the M/IO* signal is low during I/O cycles.

Interrupt Acknowledge Cycle

An interrupt acknowledge cycle is the same as a read cycle except that an address of 0 or 2 is placed on the address bus and the bus control signals are set to indicate an interrupt acknowledge cycle. The value read during the interrupt acknowledge cycle is then multiplied by 4 and used to load an interrupt vector from this address in memory.