$\overline{\text{LOCK}}$: This signal is used in multiprocessor systems. In multiprocessor systems resources are shared, such as global memory. The $\overline{\text{LOCK}}$ is signal is used to ensure that the 80386DX has uninterrupted control of the system bus and the shared resource. By making $\overline{\text{LOCK}}$ output, 0 the 80386DX can lock up the shared resource for the other masters in the system.

Bus control signals :

The three bus control signals allow external logic to control the bus cycle. These signals are 1. $\overline{\text{READY}}$ 2. $\overline{\text{Next Address}}$ ($\overline{\text{NA}}$) 3. Bus Size 16 ($\overline{\text{BS16}}$).

$\overline{\text{READY}}$: It is used primarily to synchronize slower peripherals with the microprocessor. This signal is produced by the microcomputer's memory or I/O subsystem. When $\overline{\text{READY}}$ signal is logic 0, slow memory or I/O devices tell 80386DX that they are ready for next data transfer. If ready is logic 1 then processor enters wait state since logic 1 on ready pin indicates that, the data transfer of current cycle is not yet completed.

Next Address Request ($\overline{\text{NA}}$) : The external bus control logic control this signal. It activates pipelining by making next address request input low. Pipelining increases the address to data access time. By increasing the address to data access time, same level of performance can be obtained with slower, memory devices.

BUS Size 16 ($\overline{\text{BS16}}$) : This signal activates 16-bit data bus operation; data is transferred on the low-order 16-bits of the data bus, and an extra cycle is provided for transfers of more than 16-bits.

13.2.2 Interrupt Interface Signals

There are three interrupt interface signals :

1. Interrupt request (INTR) 2. Nonmaskable interrupt request (NMI)
3. System reset (RESET).

INTR : The INTR input of the 80386 allows external devices to interrupt 80386 program execution. This input is sampled at the beginning of each instruction cycle. To ensure recognition of interrupt by the 80386, the INTR input must be held high until the 80386 acknowledges the interrupt by performing the interrupt acknowledge cycles. Thus it must be high at least eight CLK2 periods prior to the instruction to guarantee recognition as a valid interrupt. This specific requirement reduces the false triggering.

Nonmaskable Interrupt (NMI) : As name indicates this interrupt input is non maskable. This input is edge-triggered. A valid interrupt on this pin causes 80386 to execute interrupt service routine. The 80386 will not service subsequent NMI requests until the current request has been serviced.

Reset : Reset input forces 80386 to go into the reset state. This is an active high signal. When this signal in high it resets system resources, such as I/O ports, and the interrupt flag. After reset 80386 starts execution of program from memory address FFFFFFF0H in real mode.

13.2.3 DMA Interface Signals

HOLD and HLDA : These pins are used to interface DMA controller. The DMA controller can request for bus access by asserting HOLD pin and 80386DX tells the DMA controller that the buses are available by asserting HLDA signal. The 80386DX activates its HLDA signal after completion of current bus cycle and it enters in HOLD state. In HOLD state, its local bus signals are in high impedance state.

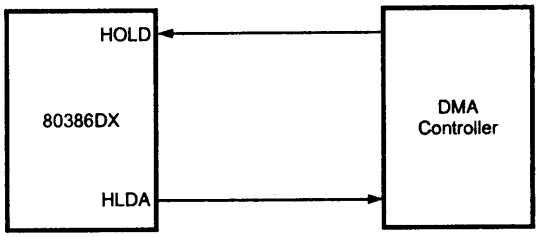

Fig. 13.19 DMA interface

13.2.4 Coprocessor Interface Signals

These signals are used to interface either 80287 or 80387 to the 80386DX. This group includes three signals (1) $\overline{\text{BUSY}}$ (2) $\overline{\text{ERROR}}$ (3) PEREQ (Coprocessor request)

$\overline{\text{BUSY}}$: $\overline{\text{BUSY}}$ and $\overline{\text{ERROR}}$ are status signals from the coprocessor. $\overline{\text{BUSY}}$ signal is used to tell 80386 that the coprocessor is executing a numeric instruction. Thus when $\overline{\text{BUSY}}$ is logic 0, 80386 does not request the numeric coprocessor to perform another calculation until $\overline{\text{BUSY}}$ returns to logic 1.

$\overline{\text{ERROR}}$: $\overline{\text{ERROR}}$ signal is used to indicate occurrence of error in the calculation. If an error occurs in a calculation performed by the numeric coprocessor, it informs 80386 that error has occurred by activating $\overline{\text{ERROR}}$ signal.

PEREQ : The coprocessor cannot transfer data over the data bus by itself. Whenever the coprocessor needs to read or write data from memory, it signals 80386 to initiates data transfer. The coprocessor does this by making PEREQ signal high.

13.3 Bus Interface

The internal and external bus operations of 80386 are synchronized by the clock signal. The 80386 perform variety of bus cycles in response to internal requirements and external requirements. There are seven types of bus cycles / operations.

1) Memory read
2) Memory write

3) I/O read
4) I/O write
5) Instruction fetch
6) Interrupt acknowledge
7) Halt / Shut down

$M/\overline{IO}$	$D/\overline{C}$	$W/\overline{R}$	Bus Cycle Type	Locked ?
Low	Low	Low	INTERRUPT ACKNOWLEDGE	Yes
Low	Low	High	Does not occur	–
Low	High	Low	I/O DATA READ	No
Low	High	High	I/O DATA WRITE	No
High	Low	Low	MEMORY CODE READ	No
High	Low	High	HALT : Address=2 — ($\overline{BE0}$-$\overline{BE1}$ $\overline{BE2}$ $\overline{BE3}$ A2-A31) = (High High Low High Low); SHUT DOWN : Address=0 — ($\overline{BE0}$ $\overline{BE1}$ $\overline{BE2}$ $\overline{BE3}$ A2-A31) = (Low High High High Low)	No
High	High	Low	MEMORY DATA READ	Some cycles
High	High	High	MEMORY DATA WRITE	Some cycles

Table 13.6

In each bus cycles corresponding status signals are activated. The Table 13.6 shows the status signals along with the bus cycles. Table 13.6 also shows that memory read and memory write bus cycles can be locked to prevent another bus master from using the bus. Before going to see bus cycles it is necessary to know about bus states in 80386DX and system clock.

13.3.1 System Clock

System clock synchronizes the internal and external bus operations in the 80386DX. The 80386DX can operate on four different clock speeds : 80386DX - 16 (16 MHz), 80386DX - 20 (20 MHz), 80386DX - 25 (25 MHz) and 80386DX - 33 (33 MHz). Operating frequency of the 80386DX is half of the CLK2 frequency. Therefore, CLK2 of an 80386DX - 20 is driven by 40 MHz signal. Fig. 13.20 shows the CLK2 and internal clock signals.