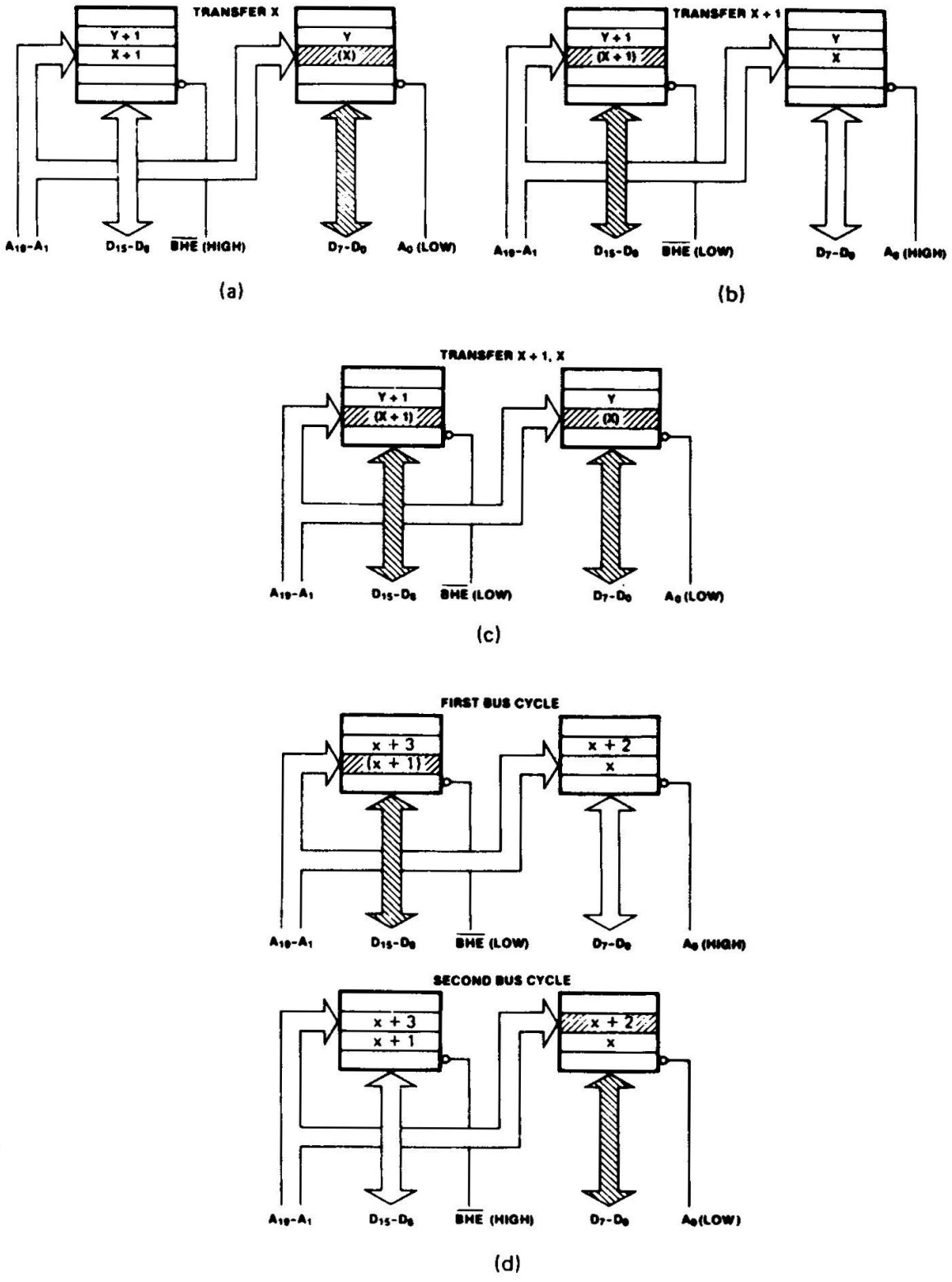

Figure 8–17 (a) Even-address byte transfer by the 8086. (Reprinted with permission of Intel Corporation, © 1979) (b) Odd-address byte transfer by the 8086. (Reprinted with permission of Intel Corporation, © 1979) (c) Even-address word transfer by the 8086. (Reprinted with permission of Intel Corporation, © 1979) (d) Odd-address word transfer by the 8086. (Reprinted with permission of Intel Corporation, © 1979)

8.14 TYPES OF INPUT/OUTPUT

- The 8086 employs two different types of input/output (I/O): **isolated I/O** and **memory-mapped I/O**. These I/O methods differ in how I/O ports are mapped into the 8086's address spaces. We will only consider isolated I/O.
- **Isolated Input/Output:** In this scheme, the I/O devices are treated **separate from** memory (see **Fig. 8-44**). I/O ports are organized as **bytes of data**; the memory address space contains 1M consecutive byte addresses in the range 00000_H, through $FFFFF_H$; and the I/O address space contains 64K consecutive byte addresses in the range 0000_H through $FFFF_H$.

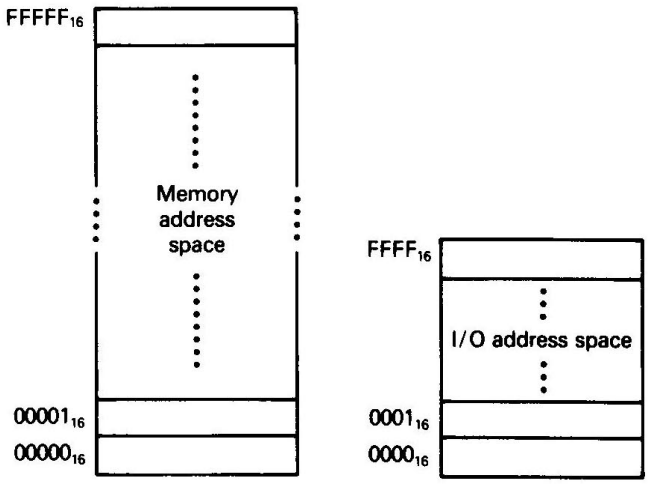

Figure 8–44 8088/8086 memory and I/O address spaces.

- **Fig. 8-45** shows that bytes of data in two consecutive I/O addresses could be accessed as **word-wide data**. For instance, I/O addresses 0000_H, 0001_H, 0002_H, and 0003_H can be treated as independent byte-wide I/O ports: **ports 0, 1, 2 and 3**, or as word-wide **ports 0 and 1**.
- For example, in **Emu8086** the Seven Segment Display was a **word-wide I/O** device that was assigned the addresses **199 and 200**, while the stepper motor was a **byte-wide I/O** device that is assigned the address **7**.

8.15 ISOLATED INPUT/OUTPUT INTERFACE

- The way in which the microprocessor deals with input/output circuitry is similar to the way in which it interfaces with memory circuitry. The only difference is that unlike memory, this time just the 16 least significant lines of the bus, AD_0 through AD_{15}, are in use (because I/O addresses are 16 bit long), and throughout the bus cycles, the $\mathbf{M/\overline{IO}}$ control signal is set to **0**.

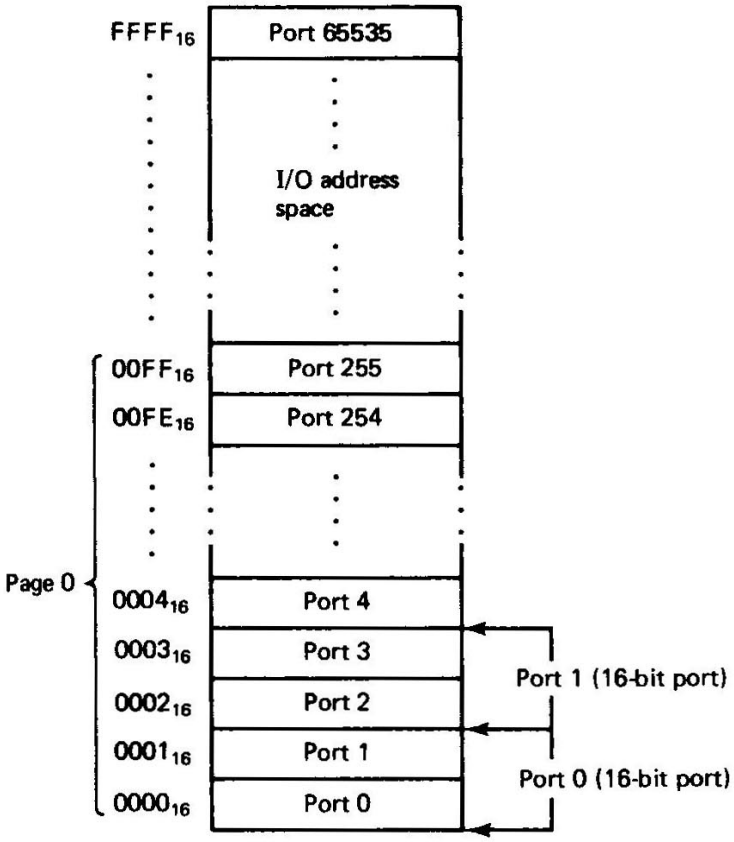

Figure 8-45 Isolated I/O ports.

8.18 NPUT/OUTPUT BUS CYCLES

- All the timing signals in the I/O read and write bus cycles other than the **M/$\overline{\text{IO}}$** are identical to those already described in the memory read/write bus cycle (See **Figs. 8-52** and **8-53**).

8.16 NPUT/OUTPUT DATA TRANSFERS

- Input/output data transfers in the 8086 microcomputers can be either **byte-wide** or **word-wide**.
- Data transfers to byte-wide I/O ports always require **one bus cycle**.
- Word data transfers between the 8086 and I/O devices are accompanied by the code A_0 **$\overline{\text{BHE}}$** = 00 and are performed over the complete data bus D_0 through D_{15}. To ensure that just **one bus cycle** is required for the word data transfer, word-wide I/O ports should be aligned at even-address boundaries.

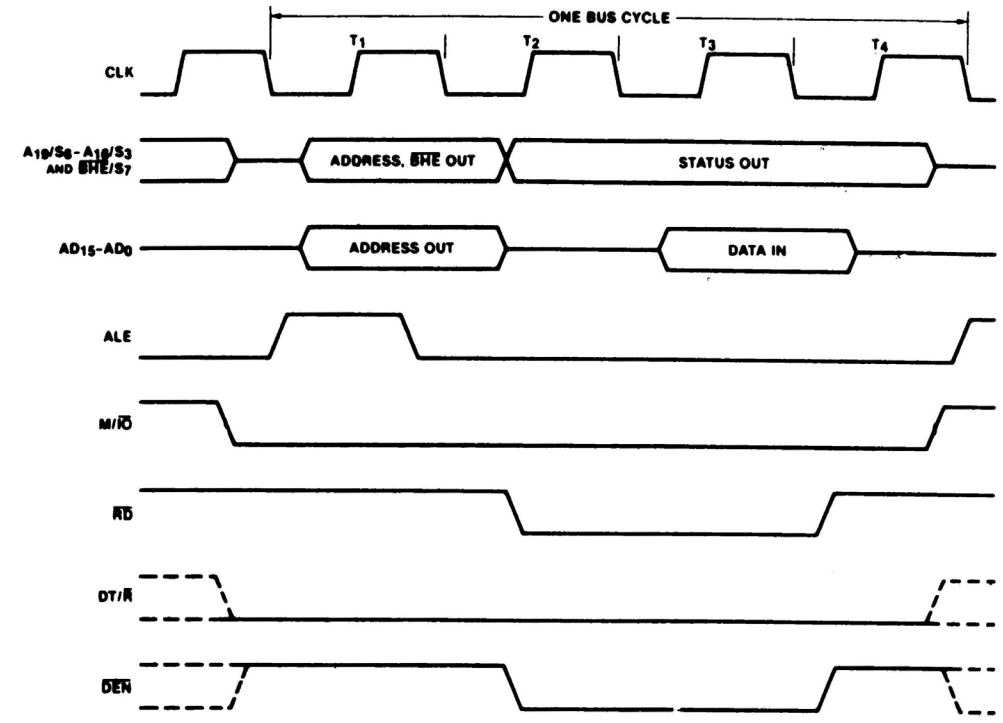

Figure 8–52 Input bus cycle of the 8086.

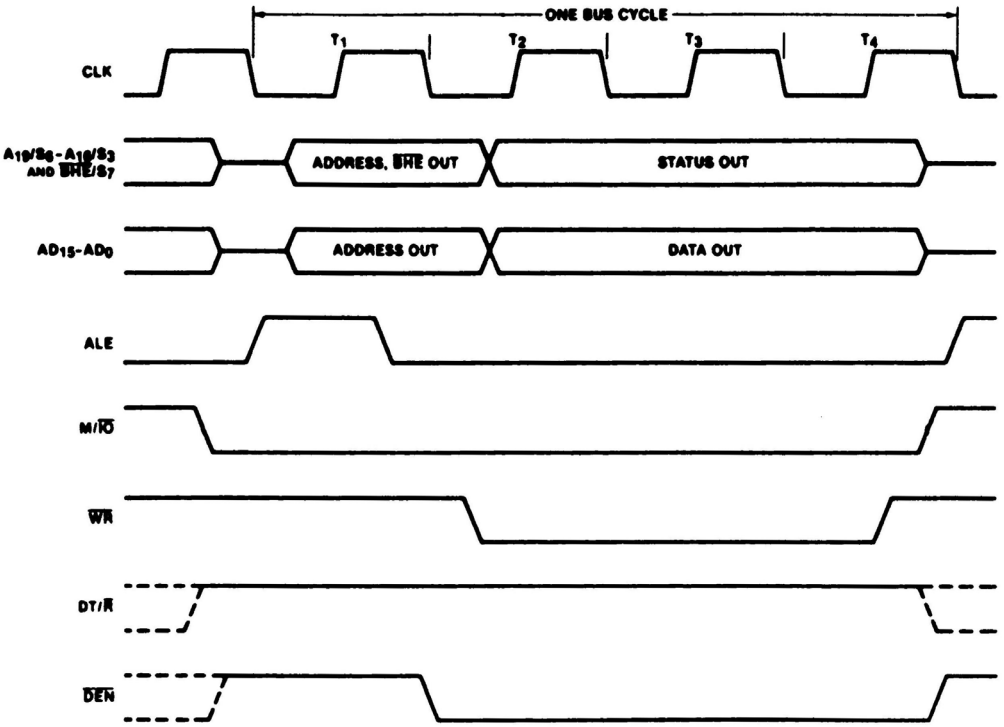

Figure 8–53 Output bus cycle of the 8086.

8.17 INPUT/OUTPUT INSTRUCTIONS

- For **isolated I/O**, special input and output instructions are used. These instructions are (IN) and (OUT), and they are described below:

Mnemonic	Meaning	Format	Operation
IN	Input direct	**IN AL, Address-8-bit**	Port → AL (Byte)
		IN AX, Address-8-bit	Port → AX (Word)
	Input indirect	**IN AL, DX**	Port → AL (Byte)
		IN AX, DX	Port → AX (Word)
OUT	Output direct	**OUT Address-8-bit, AL**	AL → Port (Byte)
		OUT Address-8-bit, AX	AX → Port (Word)
	Output indirect	**OUT DX, AL**	AL → Port (Byte)
		OUT DX, AX	AX → Port (Word)

- Byte transfers involve the AL register, and word transfers the AX Register.
- In a **direct I/O** instruction, the address of the I/O port is specified as part of the instruction. **Eight bits** are provided for this direct address. For this reason, its value is limited to the address range from **00_H to FF_H**. This range is referred to as **page 0** in the I/O address space.
- **Example:** `IN AL, 0FEh` causes the byte-wide I/O port at address FE_h to send its input to the AL register.
- **Example:** To output the data FF_h to a byte-wide output port at address AB_h of the I/O address space, we use:

```
MOV AL, 0FFh
OUT 0BAh, AL
```

- The **indirect I/O** instructions use a 16-bit address that resides in the DX register. The value in DX is not an offset. It is the actual address that is to be output on AD_0 through AD_{15}. Variable I/O instructions can access ports located anywhere in the 64K-byte I/O address space.

- **Example:** To input the contents of the byte-wide input port at $A000_h$ of the I/O address space into BL, we use

```
MOV DX, 0A000h
IN AL, DX
MOV BL, AL
```

- **Example:** To read data from two byte-wide input ports at addresses AA_h and $A9_h$ and the output the data as a word to the word-wide output port at address $B000_h$.

```
IN AL, 0AAh
MOV AH, AL
IN AL, 0A9h
MOV DX, 0B000h
OUT DX, AX
```