Assembly Language and Microcomputer Interface

Chapter 9: 8086/8088 Hardware Specifications

Ming Cai

cm@zju.edu.cn

College of Computer Science and Technology
Zhejiang University

9–1 PIN-OUTS AND THE PIN FUNCTIONS

- In this section, we explain the function and the multiple functions of each of the microprocessor's pins.
- In addition, we discuss the I/O characteristics to provide a basis for understanding the later sections of I/O interface.

The Pin-Out

- Figure 9–1 illustrates pin-outs of 8086 & 8088.
  - both are packaged in 40-pin **dual in-line** packages (DIPs)
- 8086 is a 16-bit microprocessor with a 16-bit data bus; 8088 has an 8-bit data bus.
  - 8086 has pin connections AD_0–AD_{15}
  - 8088 has pin connections AD_0–AD_7
- Data bus width is the only major difference.
  - thus 8086 transfers 16-bit data more efficiently

Minimum & Maximum Mode Operations

- The 8086 can operate in two modes:
  - Minimum Mode
  - Maximum Mode
- Minimum Mode:
  - The simplest and least expensive mode.
  - All the control signals for memory & I/O operations are generated by the processor.
- Maximum Mode:
  - Allows the system to use an external coprocessor such as 8087 (floating-point coprocessor) .
  - Some of the control signals must be externally generated (requires an external bus controller 8288)

The Pin-out of the 8086 in Maximum Mode and Minimum Mode

- VCC (+5 V Power Supply)
- GND (Ground)
- MN / $\overline{MX}$ (Minimum/Maximum):
  - indicates what mode the processor is to operate in.
  - Minimum mode: HIGH
  - Maximum mode: LOW

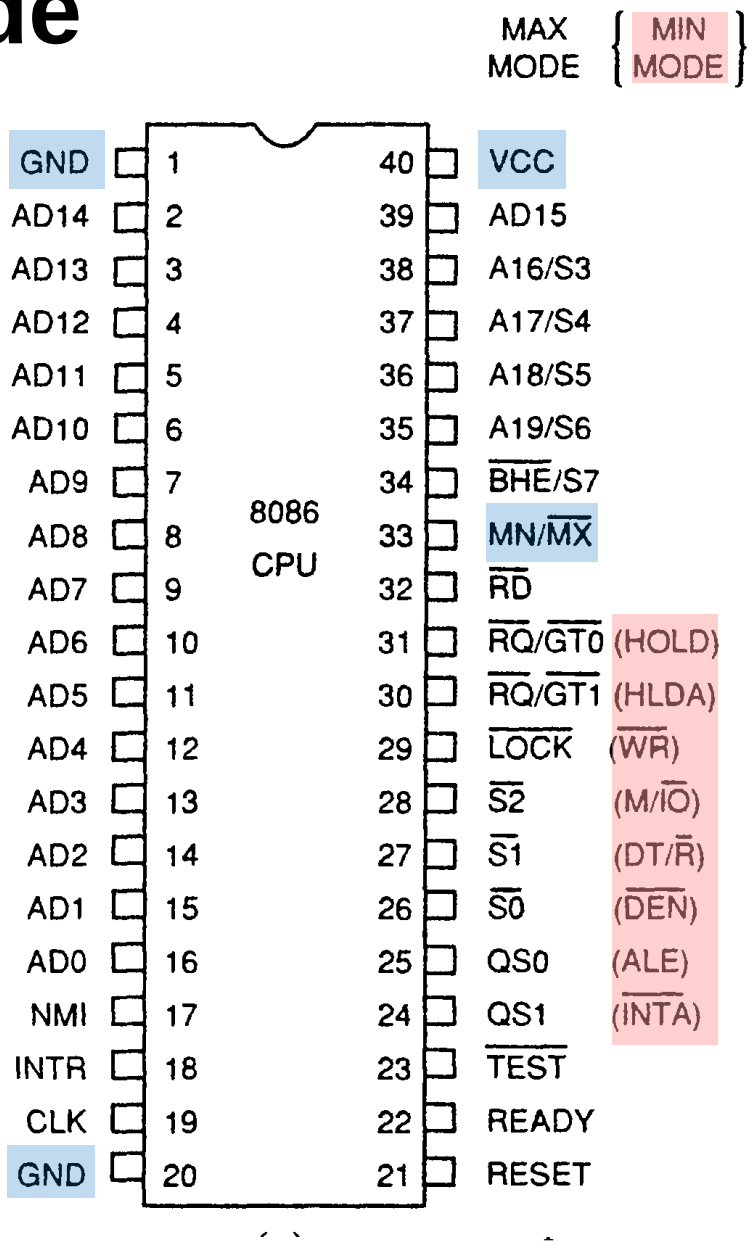
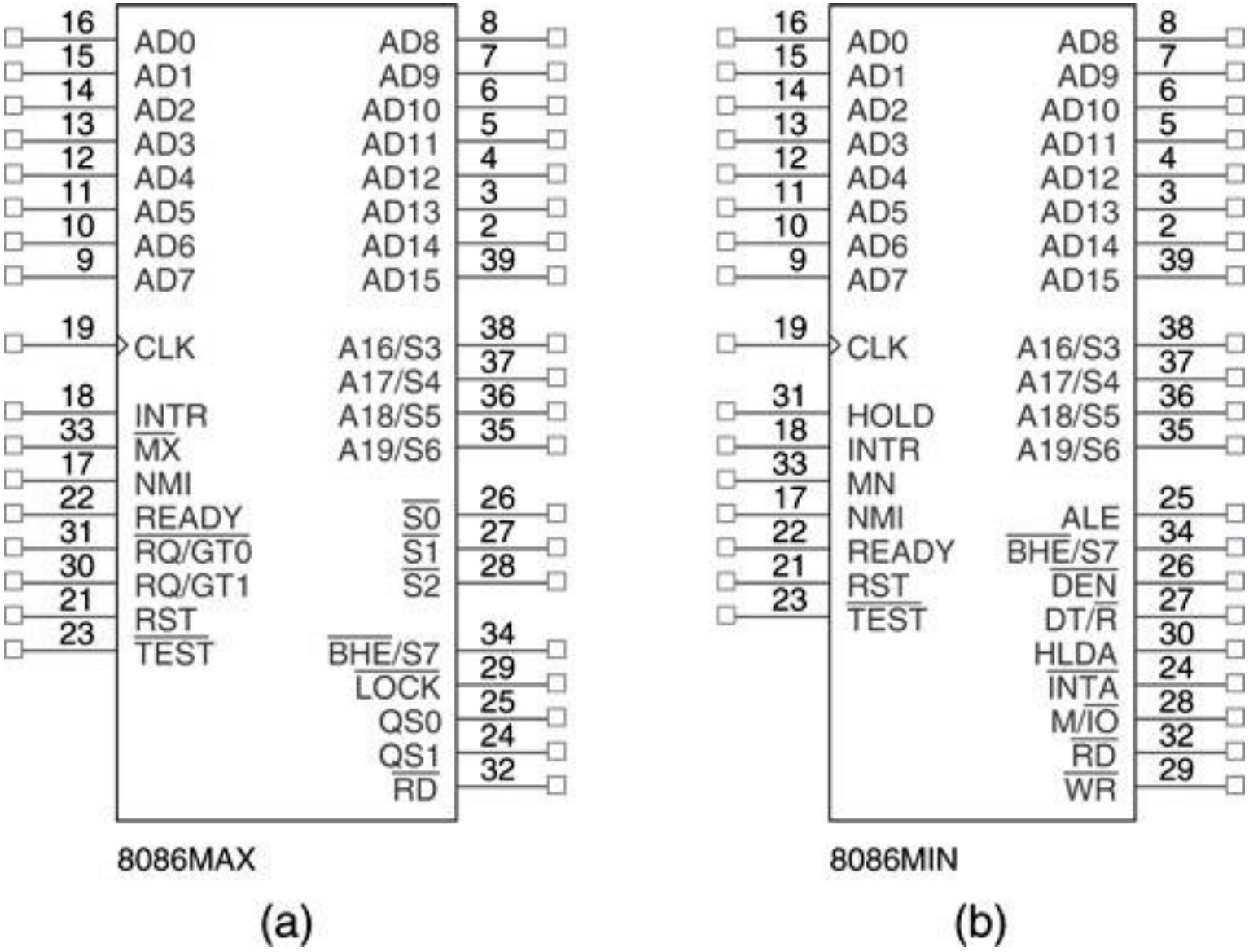


Figure 9–1 (a) The pin-out of the 8086 in maximum mode; (b) the pin-out of the 8086 in minimum mode.



Pin Connections AD_{15} - AD_0

- 8086 **address/data bus** lines are time multiplexed address and data bus lines:
  - ALE is active (logic 1): contain 16 bits memory address or I/O port number
  - ALE is inactive (logic 0): contain data

Minimum Mode Pins ALE

- **Address latch enable** shows the 8086/8088 address/data bus contains an address.
  - can be a memory address or an I/O port number
  - ALE signal doesn't float during hold acknowledge

Minimum Mode Pins IO/$\overline{M}$ or M/$\overline{IO}$

- The **IO/$\overline{M}$** (8088) or **M/$\overline{IO}$** (8086) pin selects memory or I/O.
  - indicates the address bus contains either a memory address or an I/O port address.
  - high-impedance state during hold acknowledge

BHE

- The **bus high enable** pin is used in 8086 to enable the most-significant data bus bits (D_{15}–D_8) during a read or a write operation.

Pin Connections $\overline{RD}$

- When **read signal** is logic 0, the data bus is receptive to data from memory or I/O devices
  - pin floats high-impedance state during a hold acknowledge

Minimum Mode Pins $\overline{WR}$

- **Write line** indicates 8086/8088 is outputting data to a memory or I/O device.
  - during the time WR is a logic 0, the data bus contains valid data for memory or I/O
  - high-impedance during a hold acknowledge

Pin Connections INTR

- **Interrupt request** is used to request a hardware interrupt.
  - If INTR is held high when IF = 1, 8086/8088 enters an interrupt acknowledge cycle after the current instruction has completed execution

NMI

- The **non-maskable interrupt** input is similar to INTR.
  - does not check IF flag bit for logic 1
  - if activated, uses interrupt vector 2

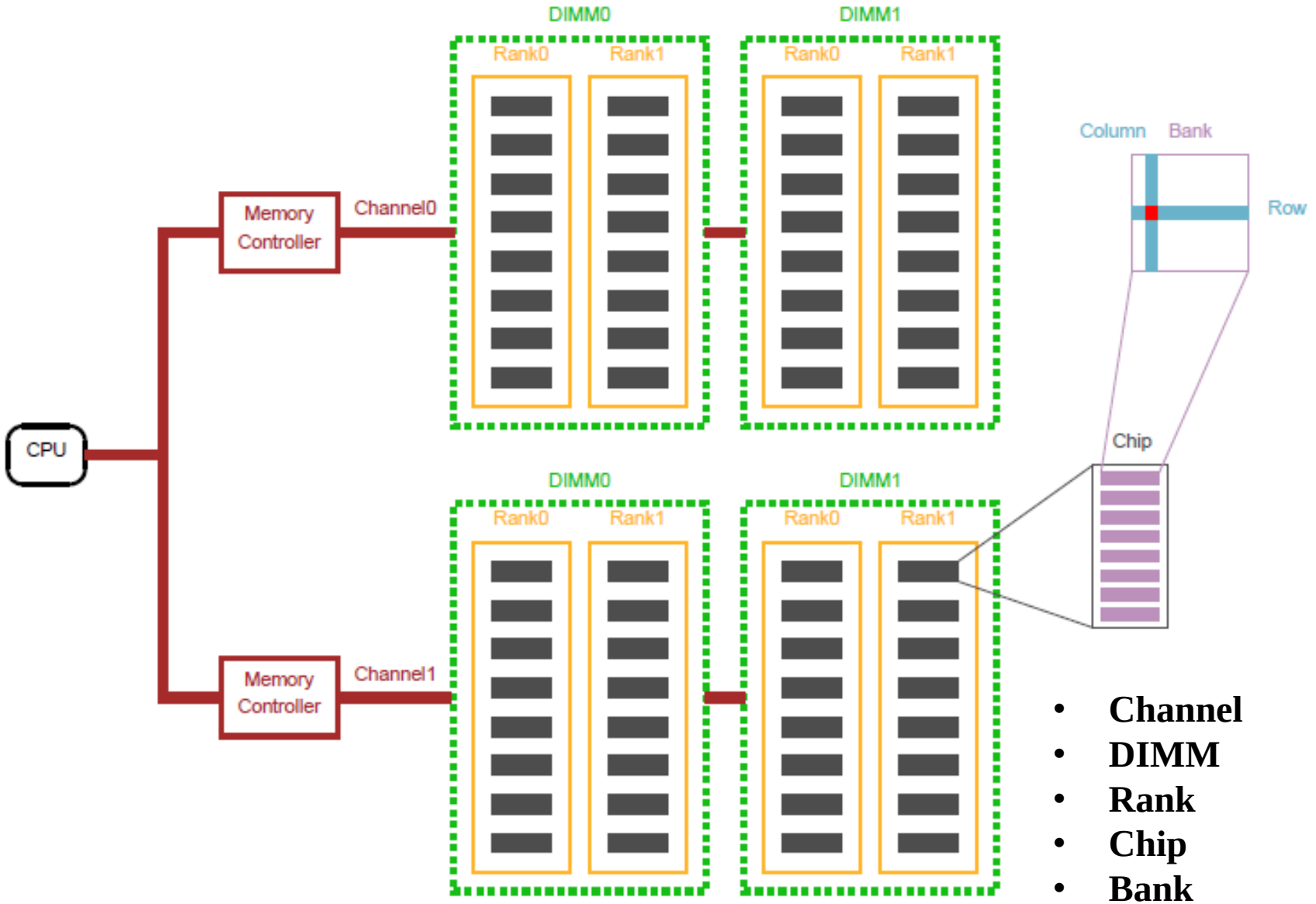
Minimum Mode Pins $\overline{\text{INTA}}$

- The **interrupt acknowledge** signal is a response to the INTR input pin.
  - normally used to gate the interrupt vector number onto the data bus in response to an interrupt

Maximum Mode Pins $\overline{\text{LOCK}}$

- The **lock** output is used to lock peripherals off the system. This pin is activated by using the LOCK: prefix on any instruction.

DRAM Organization

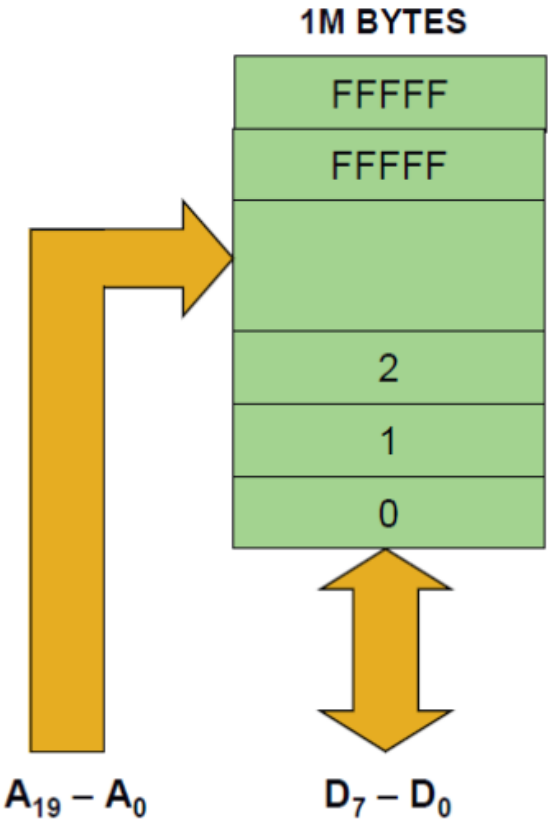


- **Channel**
- **DIMM**
- **Rank**
- **Chip**
- **Bank**
- **Row / Column**

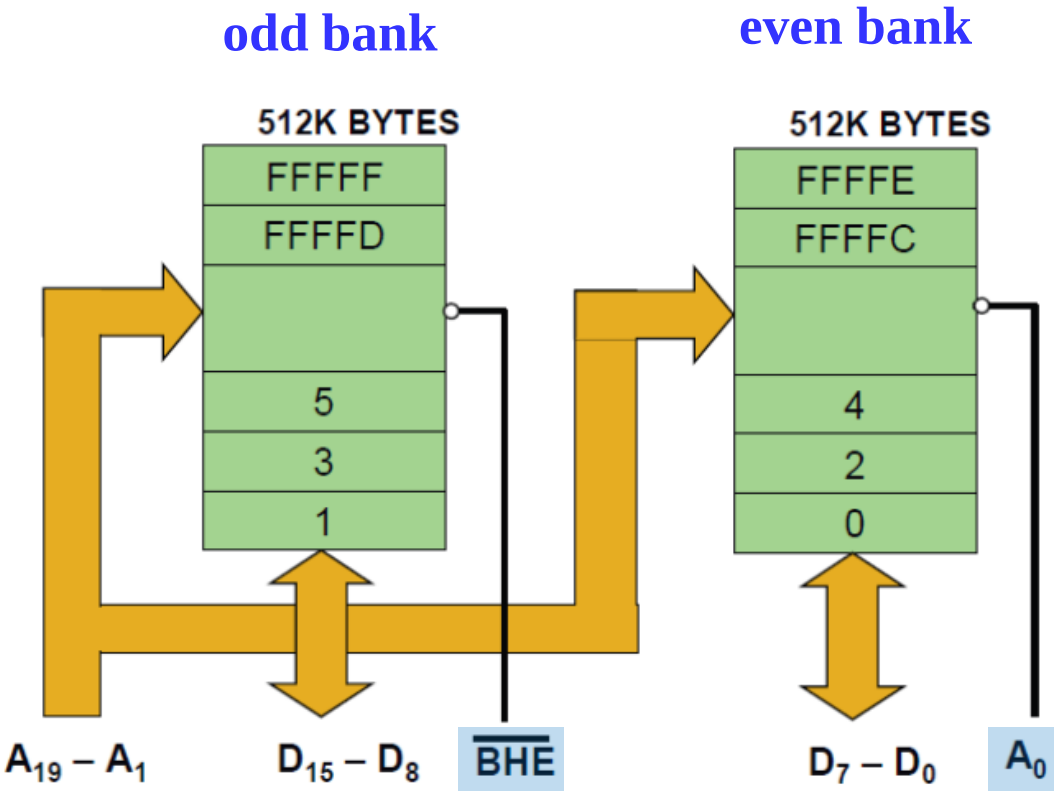
Memory Bank

- X86 uses memory banks to support one byte transfer or unaligned memory accesses.
- A "bank" refers to 8-bit wide memory. For example:
  - The 8088 has an 8-bit data bus and the memory address space is implemented as single 1 Mbyte memory bank.
  - While 8086 has a 16-bit data bus and the memory address space implemented as two independent 512 Kbyte banks.

Single Memory Bank vs. Dual Memory Bank



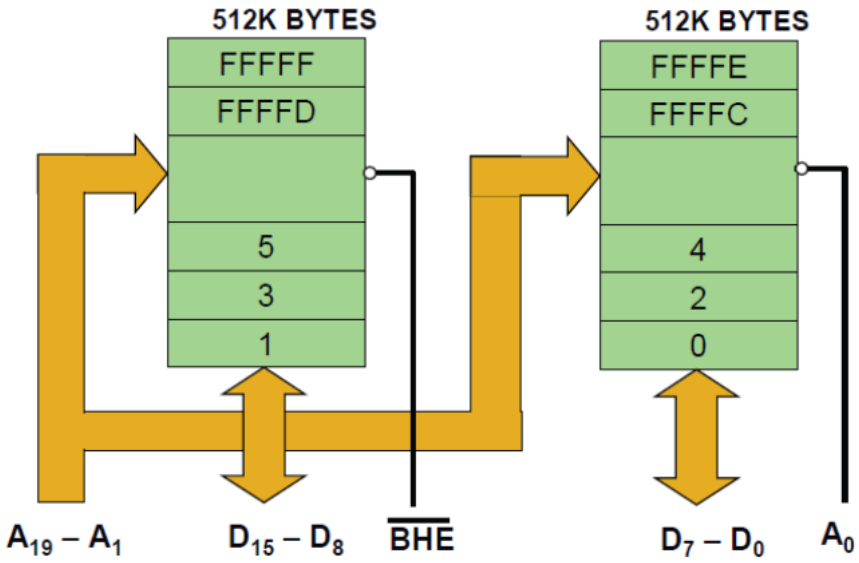
1Mx8 memory bank of the 8088



High and low memory banks of the 8086

Memory Bank Selection

- Bank high enable ($\overline{BHE}$) and bank low enable ($\overline{BLE}$/A0) are used as bank-select signals:
  - $\overline{BHE}$ = 0 enables the high/odd bank.
  - $\overline{BLE}$/A0 = 0 enables the low/even bank.
- Address bits A1-A19 select the location.

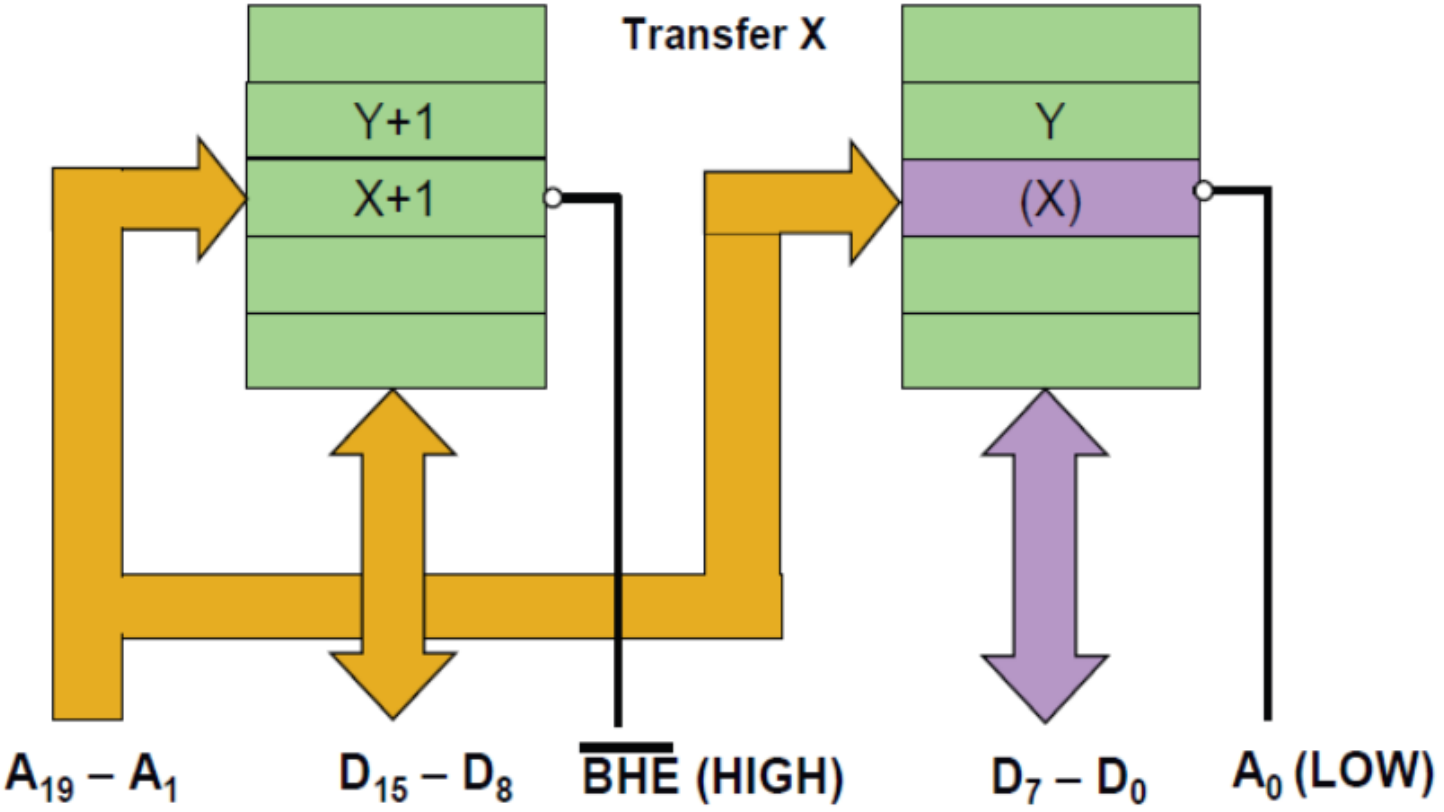


High and low memory banks of the 8086

$\overline{BHE}$	$\overline{BLE}$	Function
0	0	Both banks enabled for a 16-bit transfer
0	1	High bank enabled for an 8-bit transfer
1	0	Low bank enabled for an 8-bit transfer
1	1	No bank enabled

Memory bank selection using BHE and BLE (A_0)

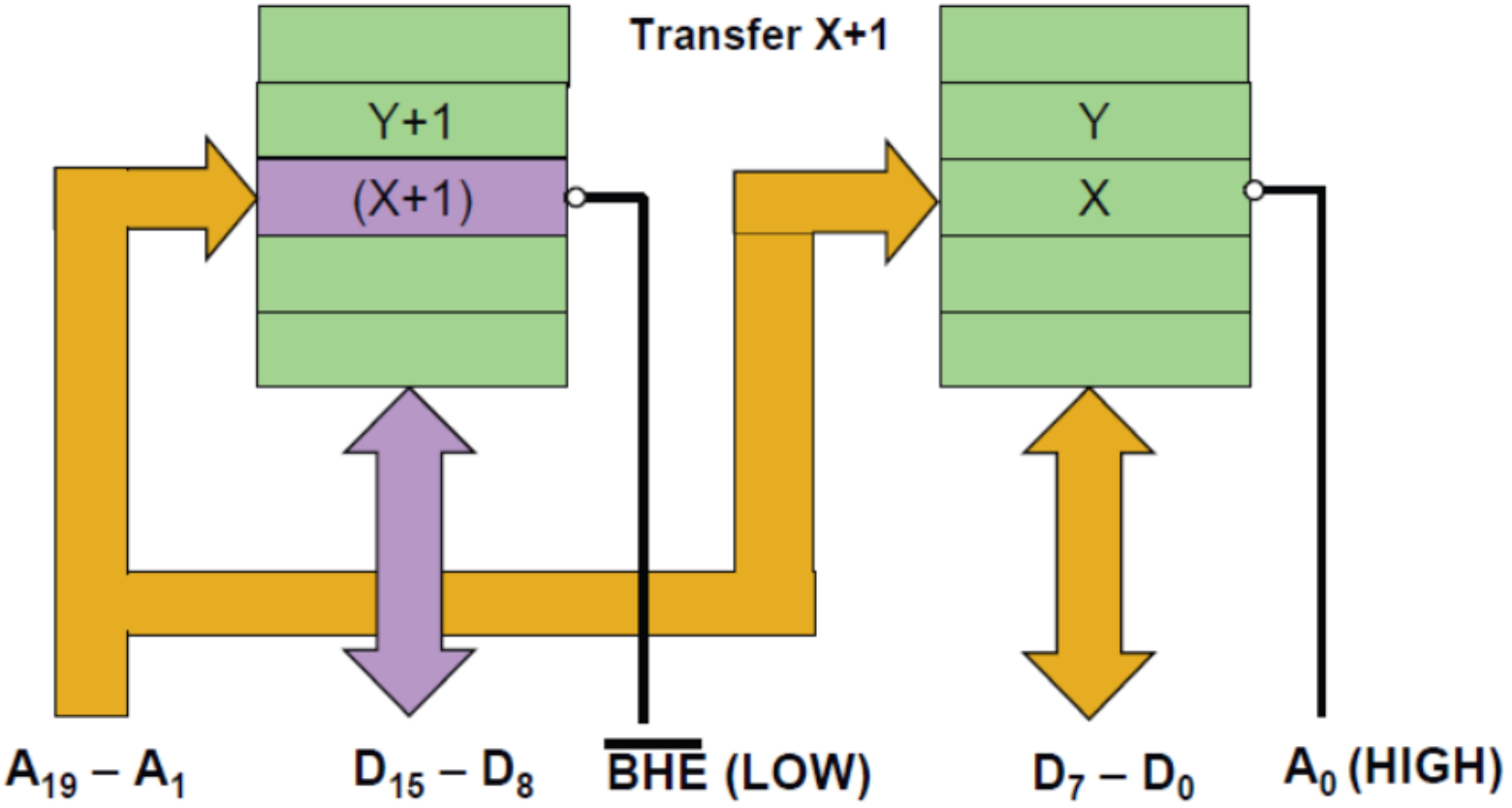
Even-addressed Byte Transfer



Even address byte transfer by the 8086

MOV [30H], AL

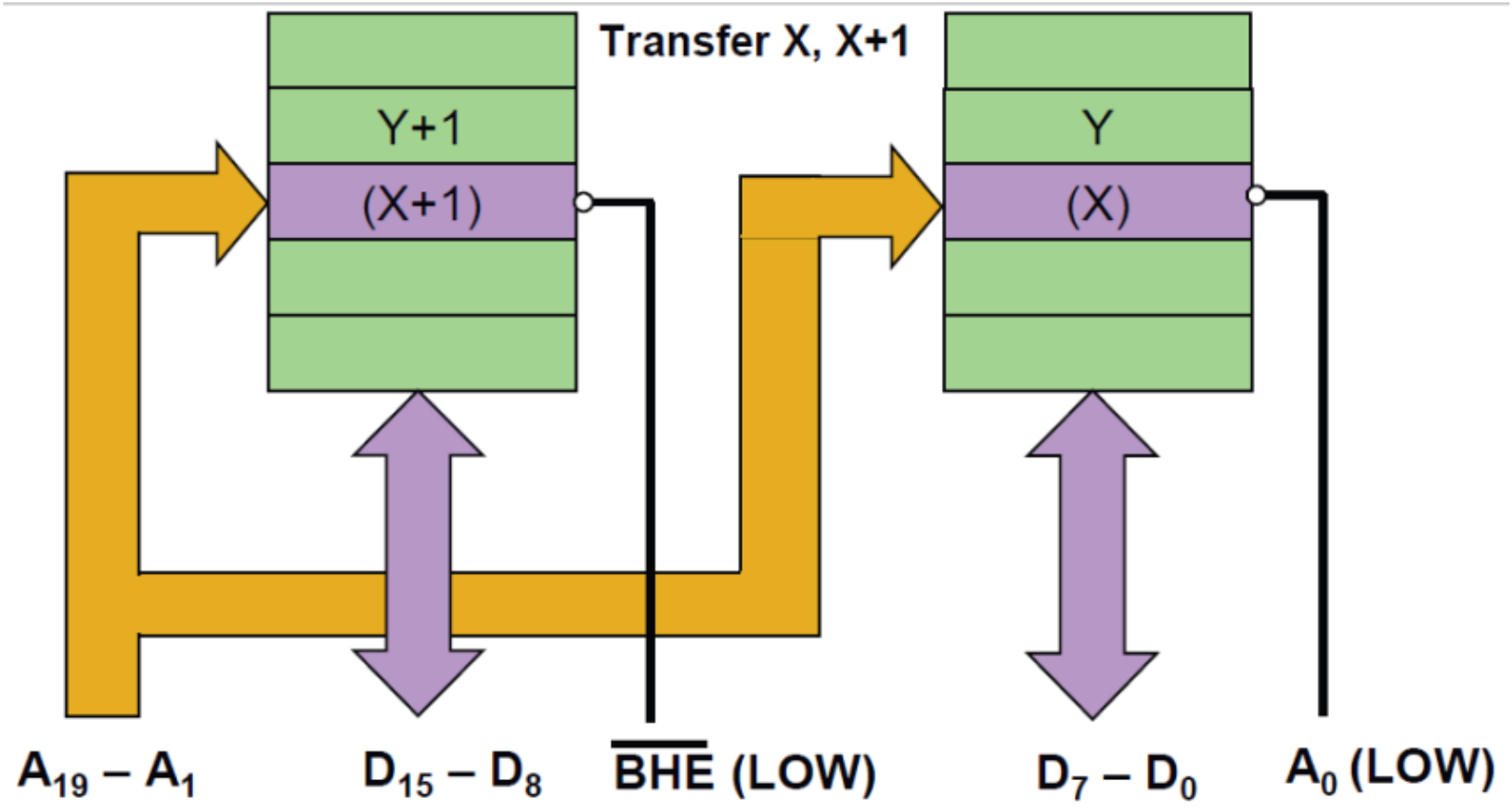
Odd-addressed Byte Transfer



Odd address byte transfer by the 8086

MOV [31H], AL

Even-addressed Word Transfer

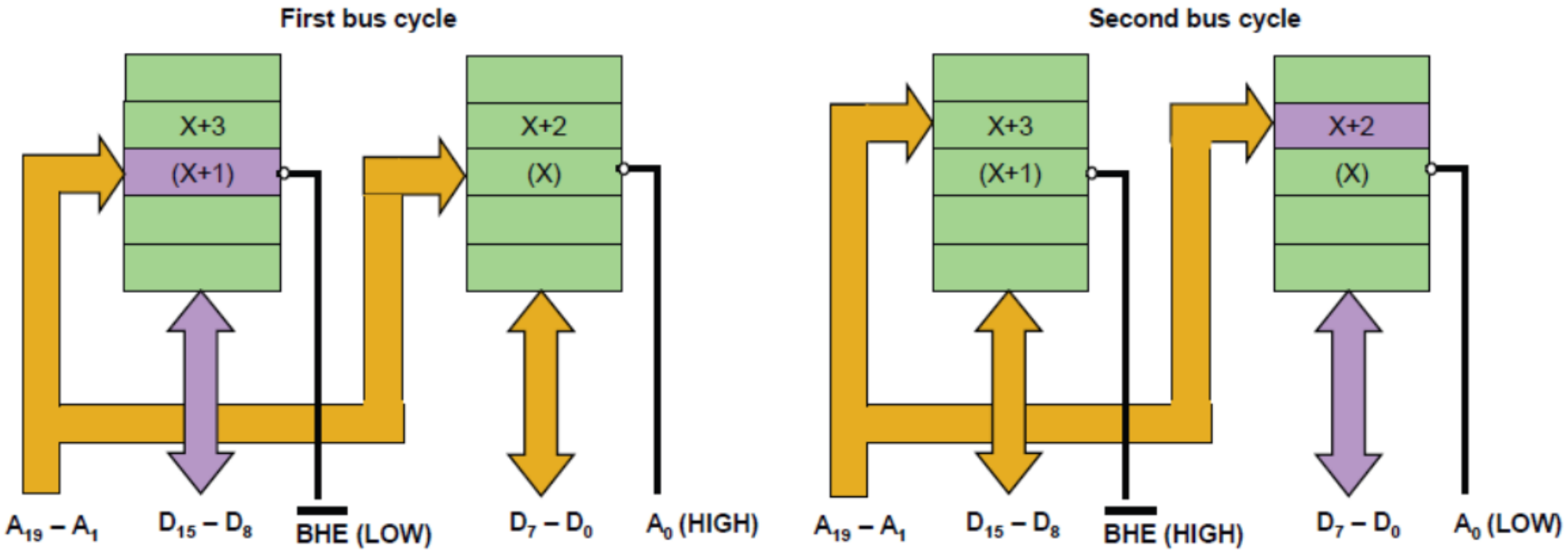


Even address word transfer by the 8086

MOV [30H], AX

Unaligned Memory Access

- Two bus cycles are needed for an unaligned memory access:
  - first cycle: data transfer uses D8-D15
  - second cycle: data transfer uses D0-D7



MOV [31H], AX

Memory Bank Summary

- Keep memory accesses aligned is of importance even on x86 machines.
- The x86 has memory and I/O space arranged in banks. Bank-select signals (BHE and BLE/A0) are used to reference byte-sized data.
- We often require separate write strobes (an upper and a lower) for memory and I/O write operations.

Appendix: Maximum Mode 8086 System