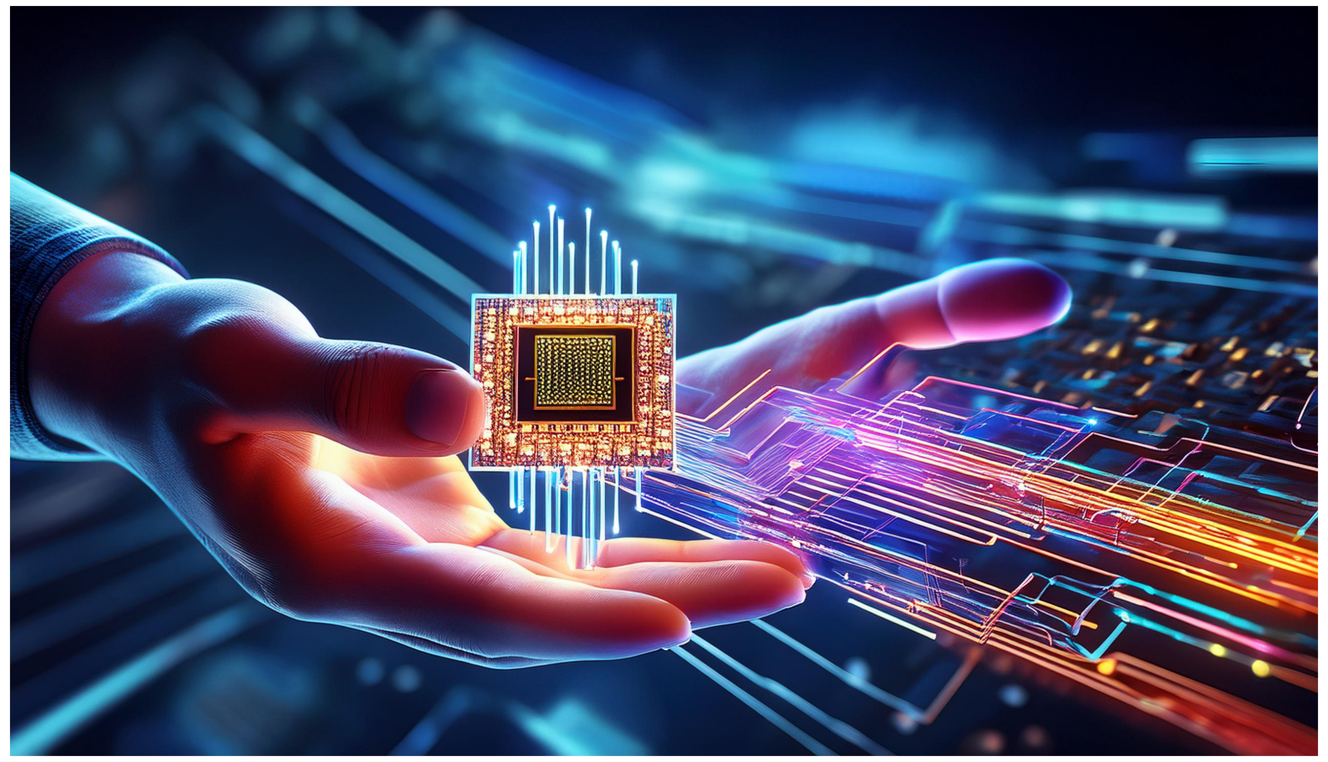
26.06.25 16:03

CSCI 250
Introduction to Computer Organisation
Lecture 3: CPU Architecture VI



Jetic Gū

Overview

- Focus: Course Introduction
- Architecture: CPU and Computer Architecture
- Core Ideas:
    1. Supplement: Intel 8086 CPU

1. https://intranetssn.github.io/www.ssn.net/twiki/pub/CseIntranet/CseAEC6504/8086PinDiagram.pdf
2. https://ece-research.unm.edu/jimp/310/slides/8086_chipset.html
3. https://www.righto.com/2023/02/8086-interrupt.html
4. https://drsunbeam.wordpress.com/wp-content/uploads/2021/04/8086-pin-functions-au-2021-2.pdf
5. https://archive.org/details/The_8086_Book_Russell_Rector_George_Alexy

P1
BG

Background

- Intel 4004: one of the first consumer microprocessor CPU, 4bit
- Intel 8008: 1972, 8bit, up to 16KB memory, 500-800kHz
- Intel 8080: 1974, 8bit, up to 64KB memory, 2-3.125MHz
- Intel 8086: 1978, 16bit, up to 1MB Memory, 5-10MHz
    - Used in the original IBM PC (Personal Computer)
    - x86-16 instruction set, the 64bit version of which still used today

Concept

P2
CPU Spec

Intel 8086

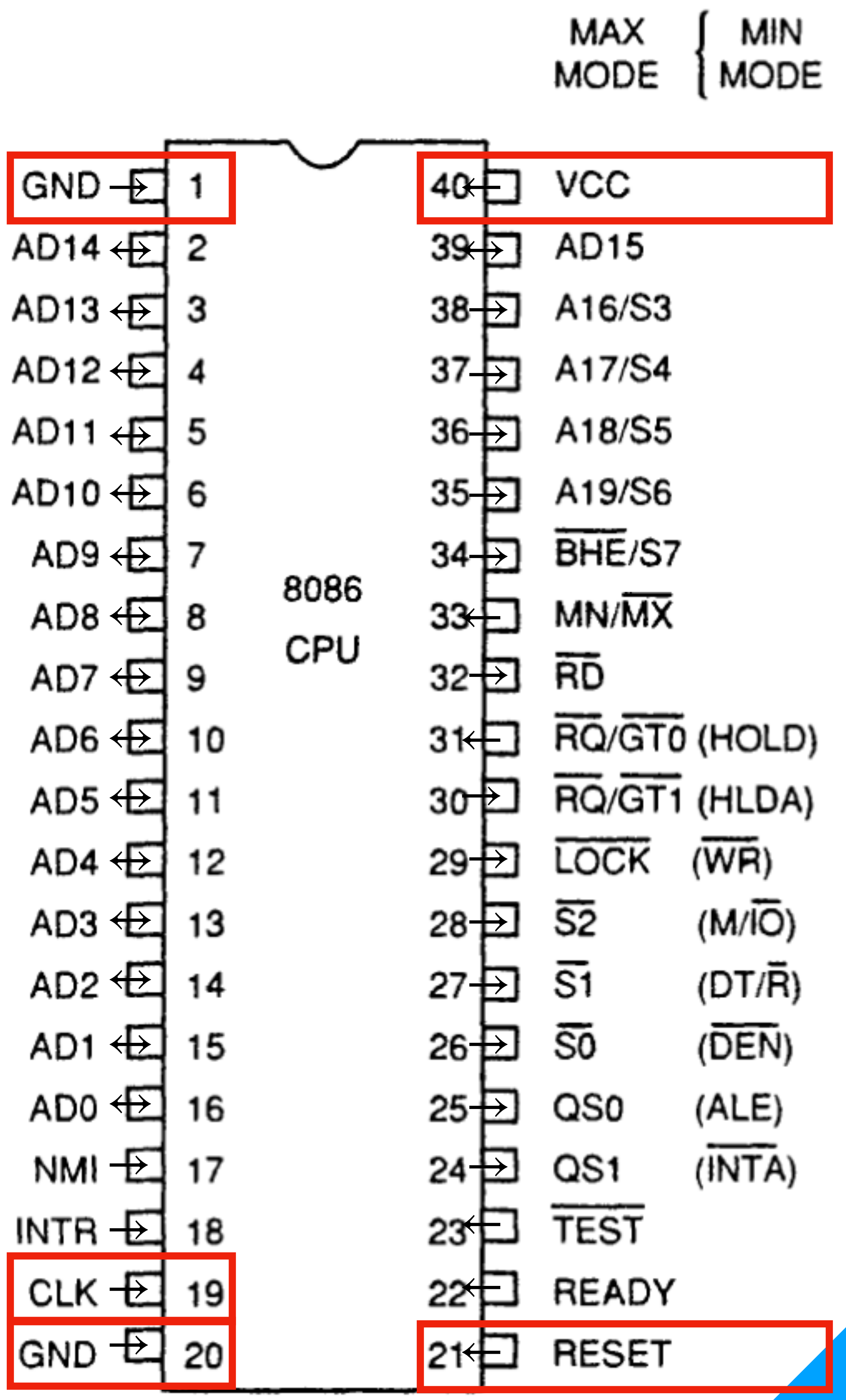
- Complex Instruction Set Computing (CISC)
- 20bit Multiplexed Address Bus
    - Address pins are also data pins
    - Address/Data? Controlled by a multiplexer
- Not really comprehensible Opcodes
- 8 General(-ish) Purpose Registers, PC not included
- 40pins

Technical

P2
CPU Spec

Power

- 5v (+-10%) Vcc
- 2 x GND pins
- Reset
  - Resets all registers, including PC
- CLK
  - 4 CLKs per machine cycle
    Memory Read/Write takes 1 machine cycle
    One instruction takes multiple machine cycles

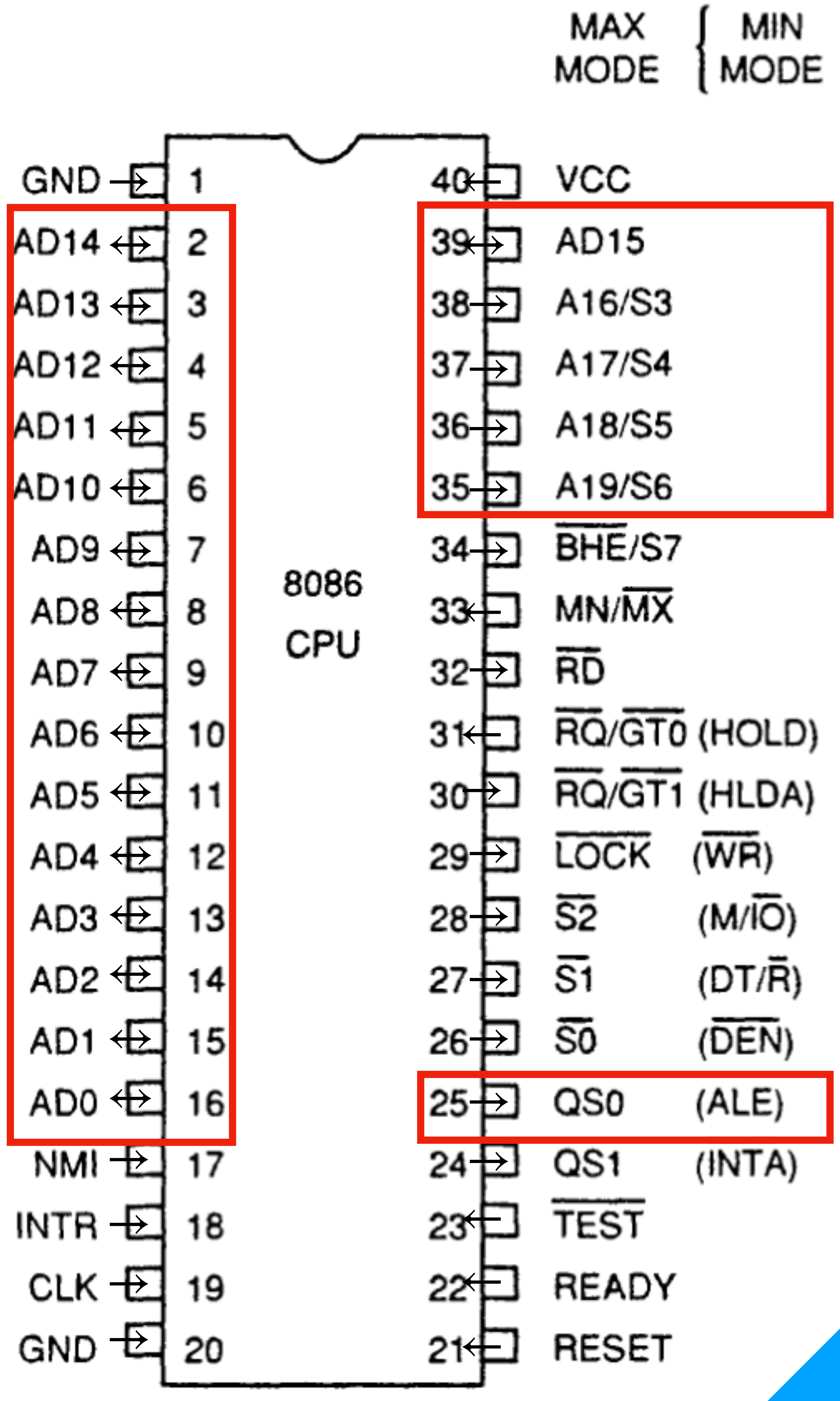


Technical

P2
CPU Spec

Address/Data bus

- If `ALE` = 1; `A(0:19)` for Address (→)
- If `ALE` = 0; `D(0:15)` for Data (↔)
  S3-S6 for status (→)
  - S6 (when not used for address) is always 0



Technical

Control Signals

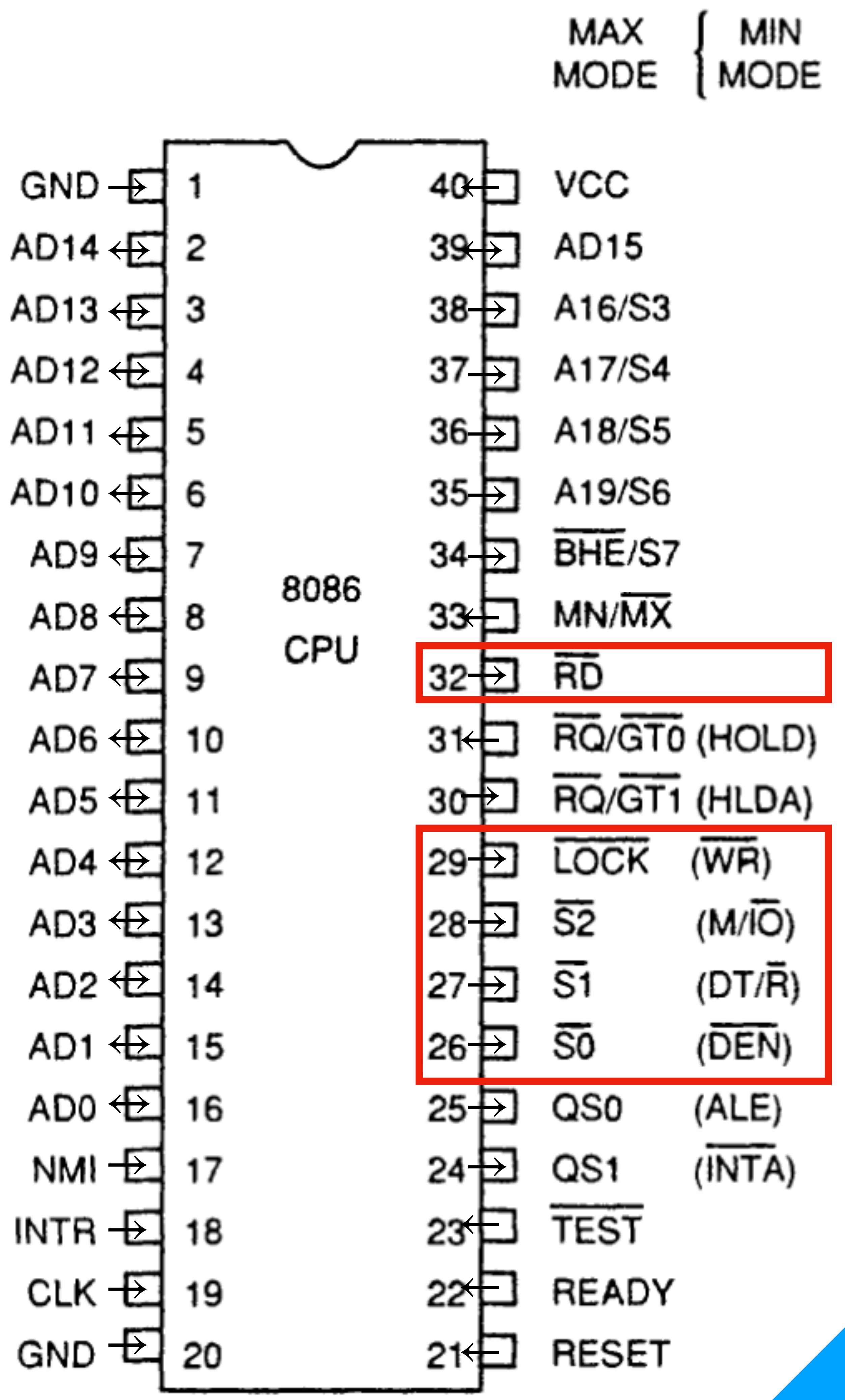
- → RD = Read Signal
  Read to Memory or IO

- → WR(LOCK) = Write Signal
  Write to Memory or IO

- → M/IO = Memory or IO
  1: memory operation; 0: IO operation

- → DT/R = Data Transmission / Receive
  1: data goes out; 0: data comes in

- → DEN = Data Bus Enable
  Enables/Disables Transreceiver 8286
  a device used to separate data from the address/data bus



Technical

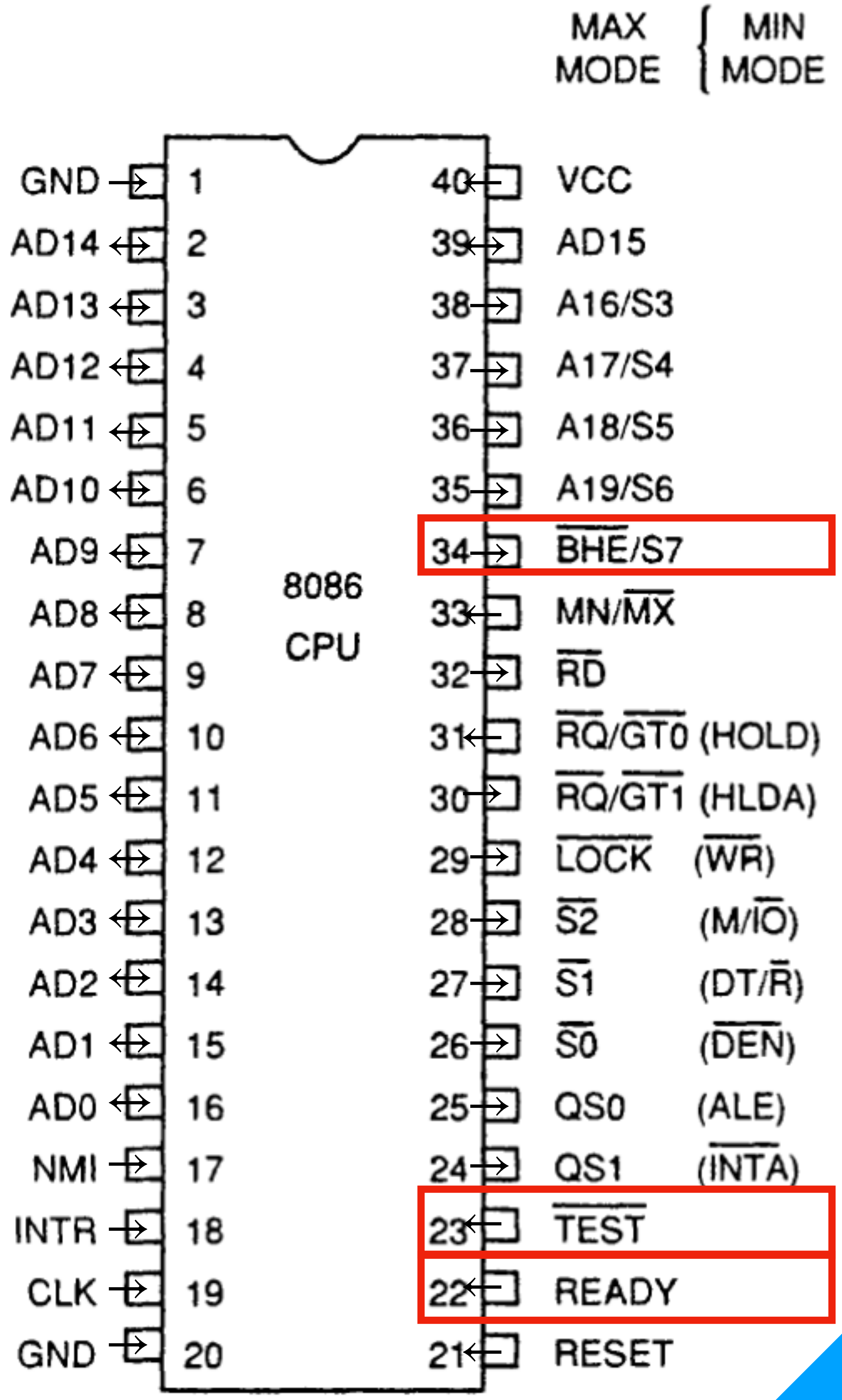
P2
CPU Spec

Control Signals

- `TEST`: Input used to test the CPU
  CPU executing the `WAIT` instruction will wait for `TEST` to turn into 0, while doing nothing

- `READY`
  `READY` = 1: CPU functions normally
  `READY` = 0: CPU idles and wait for signal to turn back to 1

- `BHE/S7`
  Outputs 0 during Read/Write/Interrupt Ack Cycles, in which data are transferred in `AD(8:15)`
  Otherwise always 1



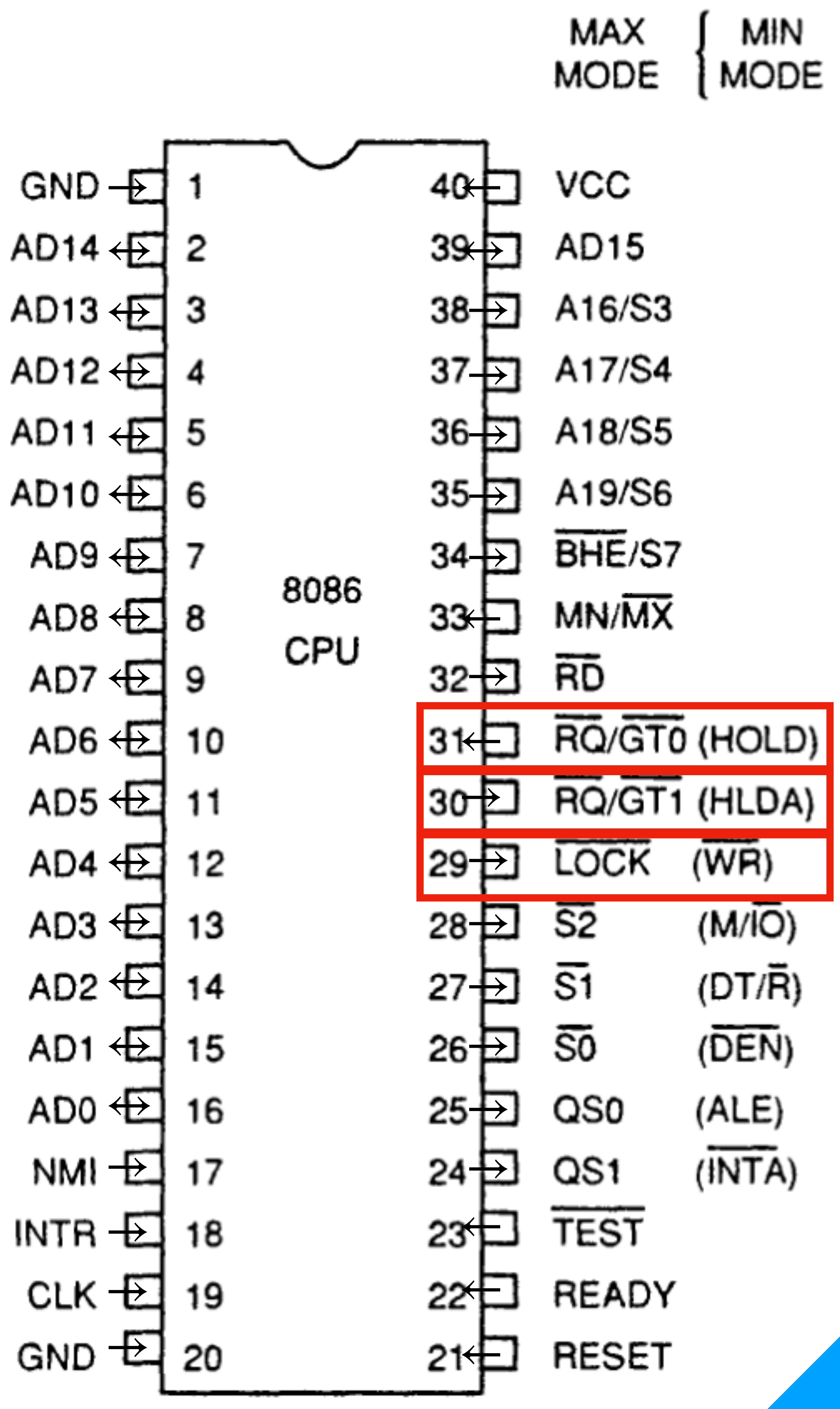
Technical

Bus Control (RQ: Request)

- ←`HOLD`
  1: Another controller requests to take over the control of the AD bus. If granted, the CPU enters the `HOLD` state
  0: CPU controls the data bus

- →`HLDA`
  Outputs 1 when CPU has entered the `HOLD` state, acknowledging transferring of control

- →`LOCK` / `WR`
  `LOCK` / `WR` can be used to prevent other bus controllers from taking over



Technical

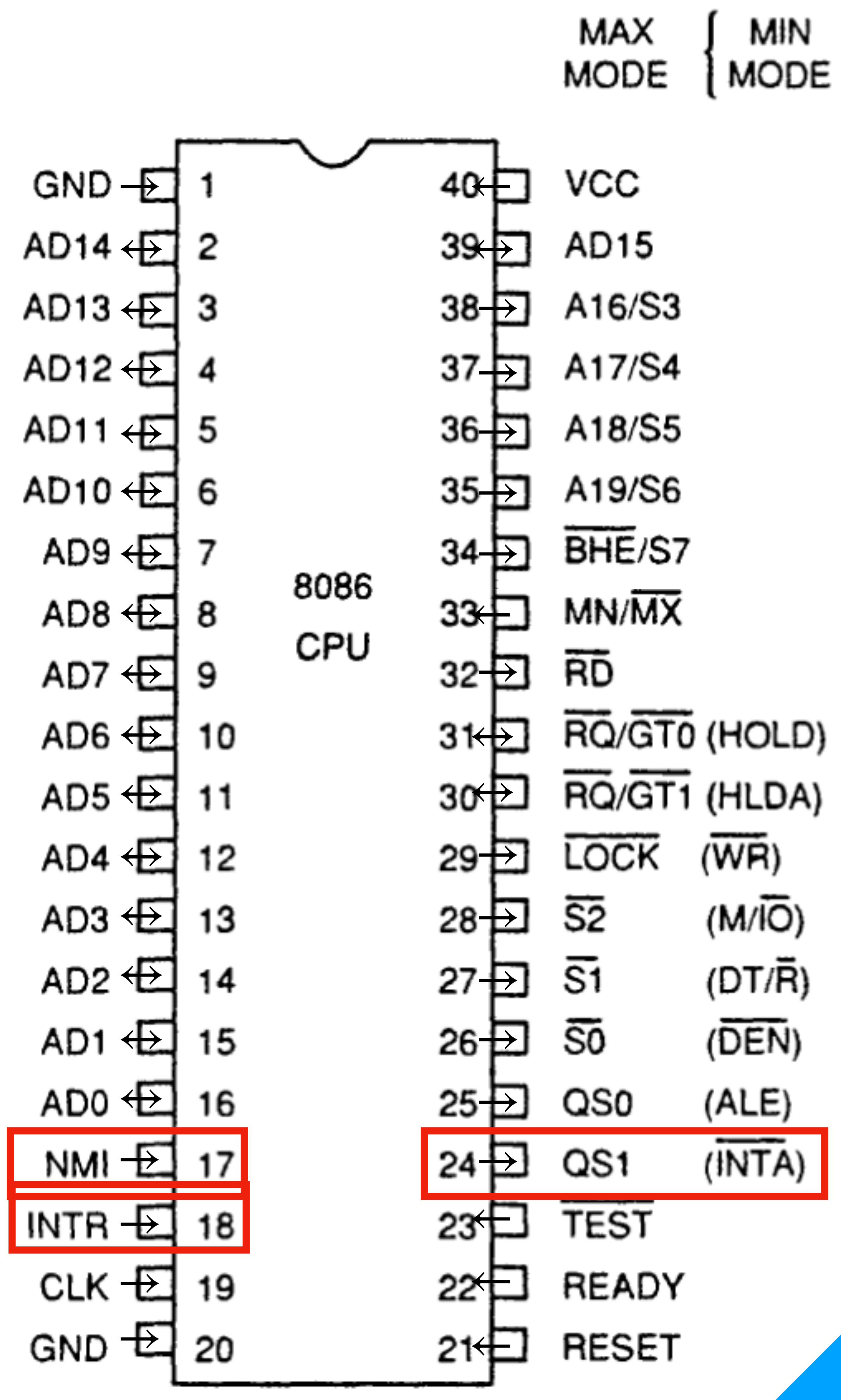
P2
CPU Spec

Interrupt

- `INTR`
  CPU receives interrupt if `INTR` = 1. If a special interrupt flag (`IF` flag) in the status register is also 1, CPU enters <u>Interrupt Acknowledgement Cycle</u>

- `NMI`
  Interrupts without checking flags like `INTR`

- `INTA`
  Signals that the CPU is in an <u>Interrupt Acknowledgement Cycle</u>



Technical

P2
CPU Spec

Interrupt Acknowledge Cycle

- **Review**: `Interrupt` means other controllers like IO wants the CPU to stop executing current instructions, and attend to them
- **Interrupt Acknowledgement Cycle (IAC)**
  - **1st machine cycle**
    Interrupt Register in the CPU is zeroed out;
    Outputs 1 at `INTA` (Interrupt Acknowledgement), sets `IF` to 1
  - **2nd machine cycle**
    Interrupt number is fetched into the Interrupt Register, just like memory read
  - **3rd machine cycle onward**
    CPU executes appropriate instructions depending on the interrupt number (Interrupt Service Routine), exits with `IRET` instruction (Interrupt Return)
  - **Exits** IAC, sets `IF` to 0

Technical