Early PC Architecture

Rémi Audebert

2015-07-18

Introduction

The question that drives us all.

How does a PC work?

General ideas about how a PC works.

- The CPU fetches instructions from memory and executes them.
- The CPU starts fetching instructions at static address.
- Devices can be memory-mapped, I/O mapped or both, depending on the CPU architecture.

These assumptions are **true**, but they **hide** complexity.

Investigating and testing is near impossible:

- Current motherboards are black boxes.
- SoC and SoM are incredibly integrated and not publicly documented.

What's required for full understanding?

In general:

- Go for an analytic approach
- Understanding the role of each component in the system.
- Understand how they are connected and how they interact.

Our approach:

- Take a computer system and understand each of its components down to the signal level.
- Because doing this on a "modern" system would be too complex, restrain us to an understandable computer system: the IBM PC/AT.

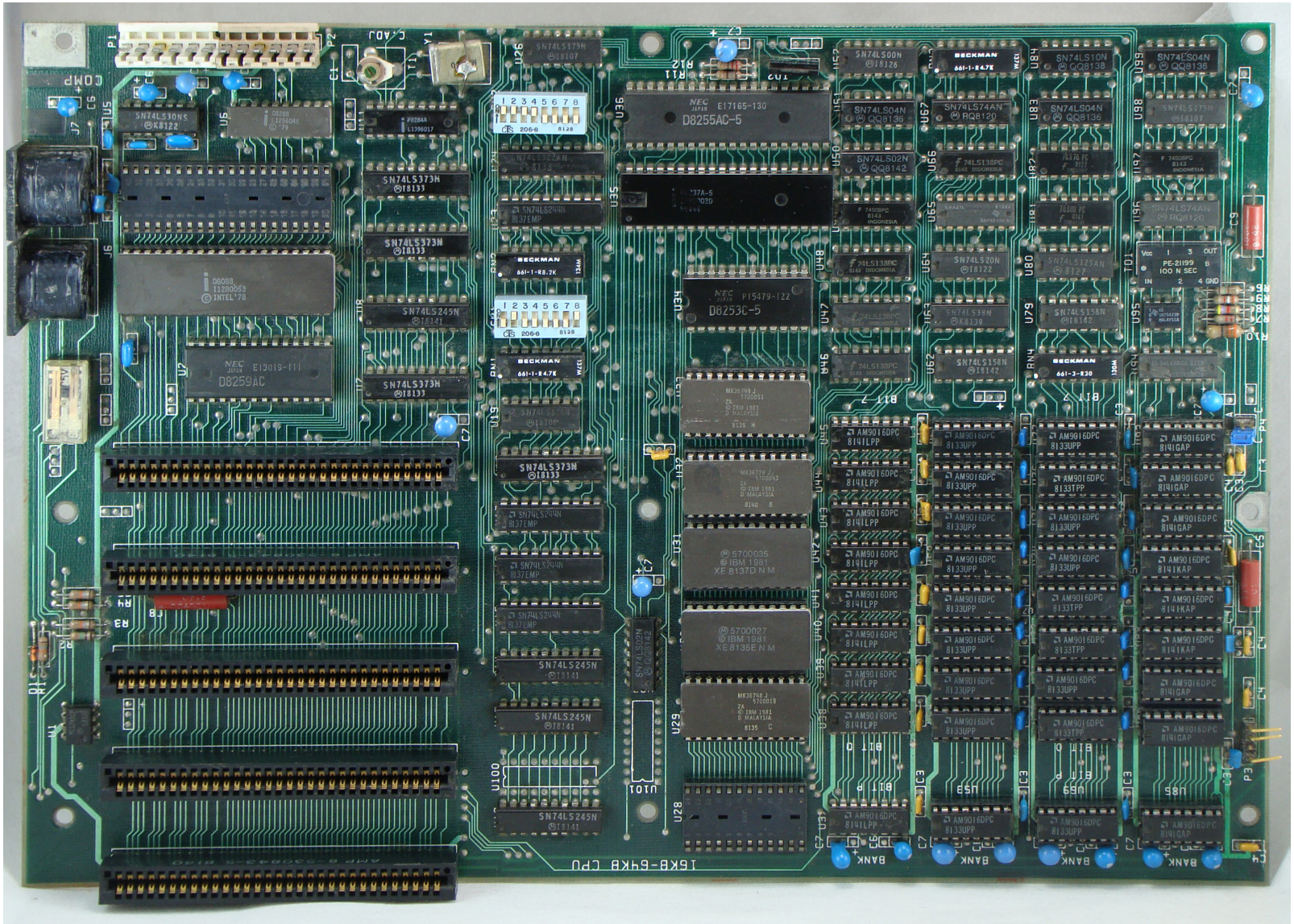
This talk

1. IBM PC, IBM PC/XT, IBM PC/AT and compatibles
2. CPU interface's to the rest of the components
3. On-board peripherals and subsystems
4. The ISA bus, and fun experiments
5. Conclusion and references

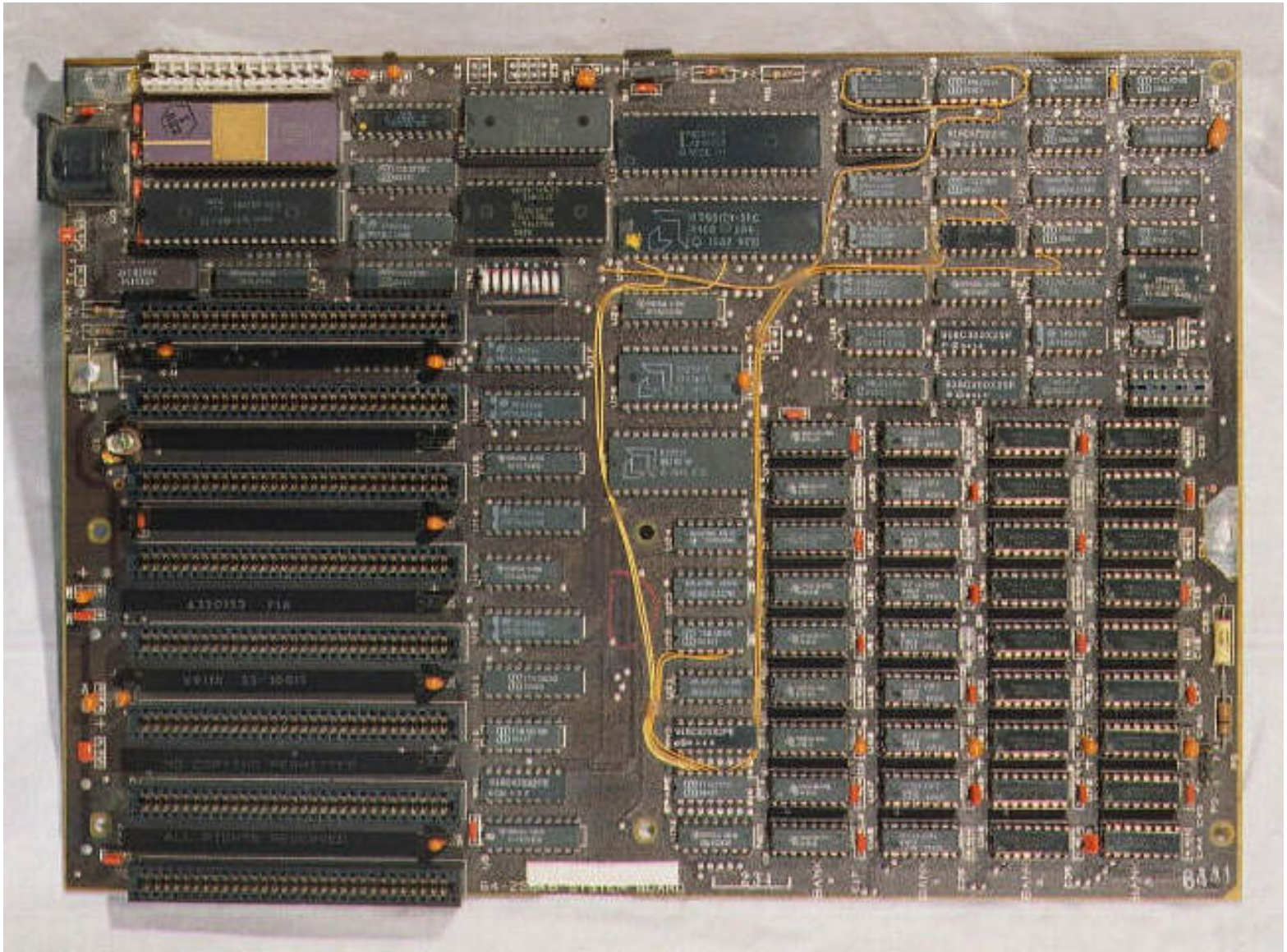
History

IBM PC (IBM 5150) - 1981

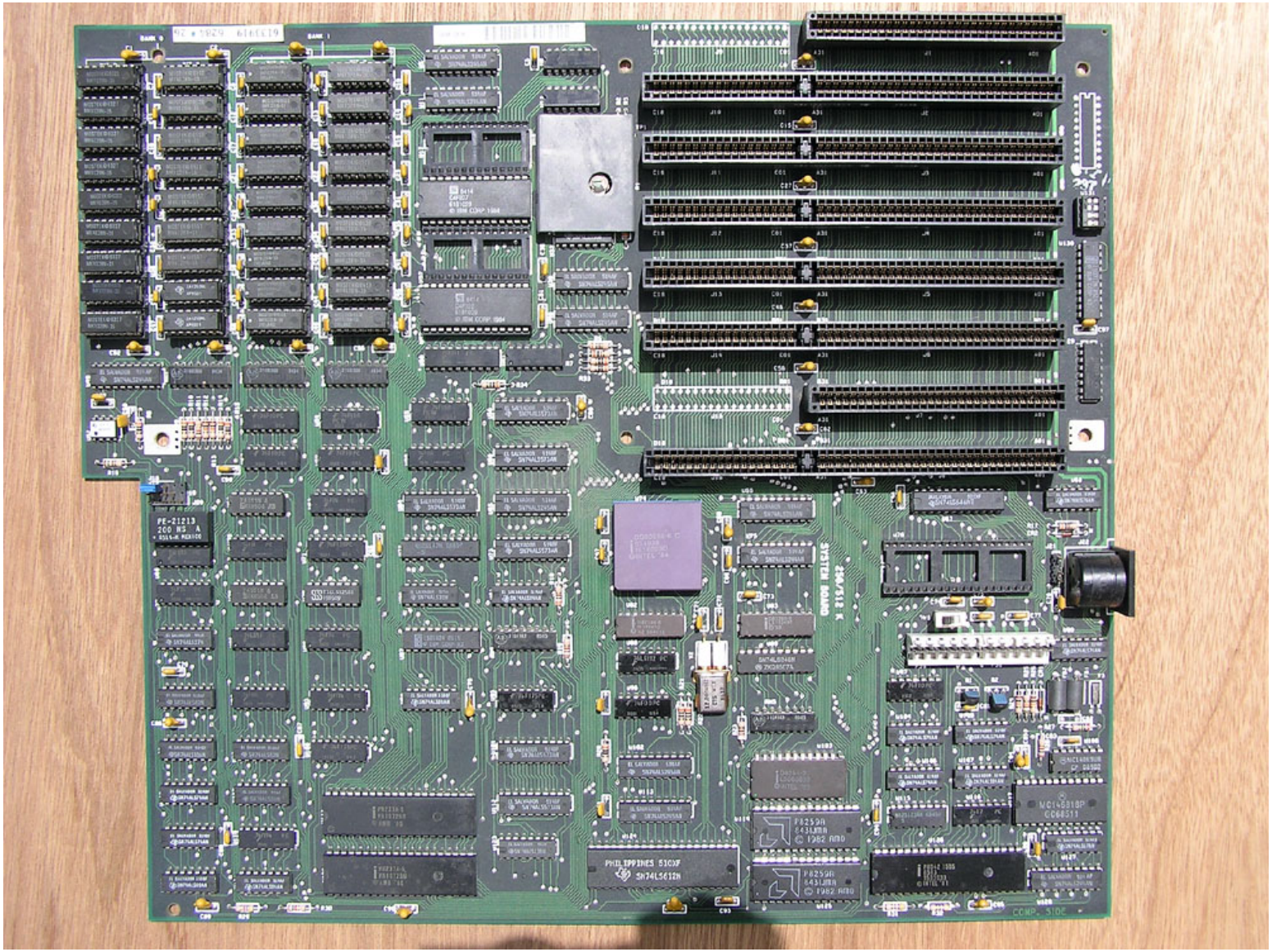
IBM PC Motherboard



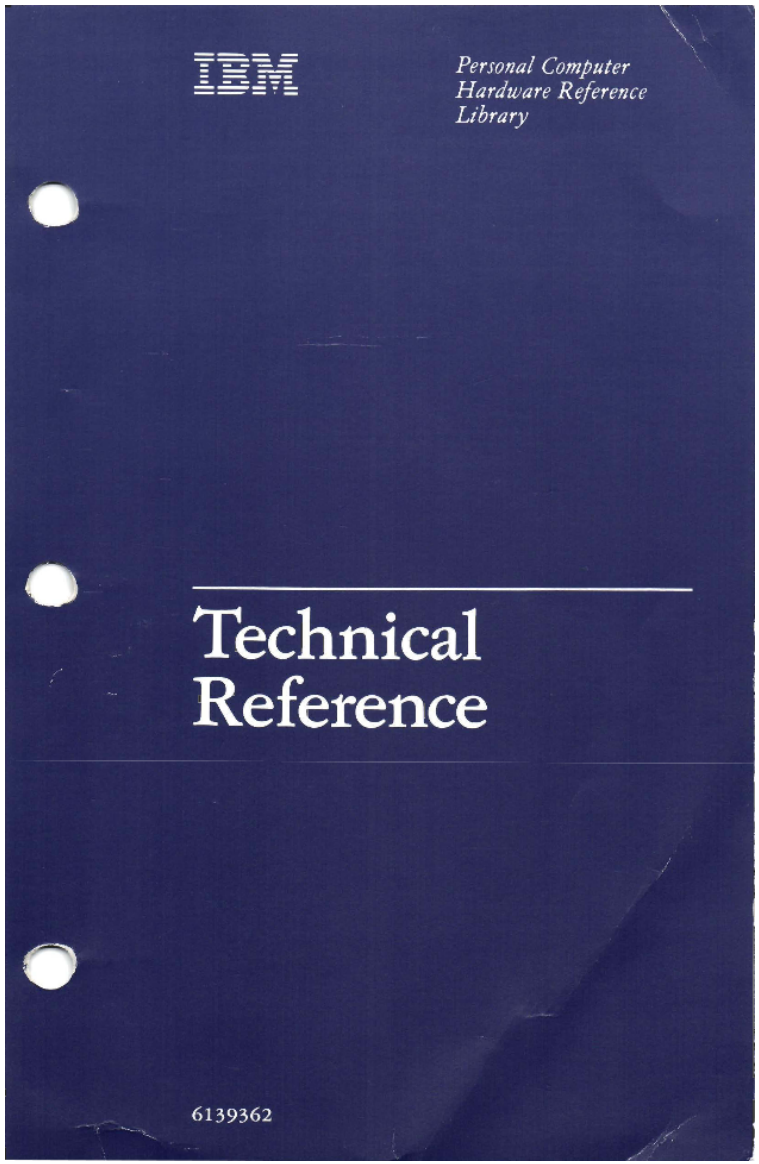
IBM PC/XT (IBM 5160) - 1983



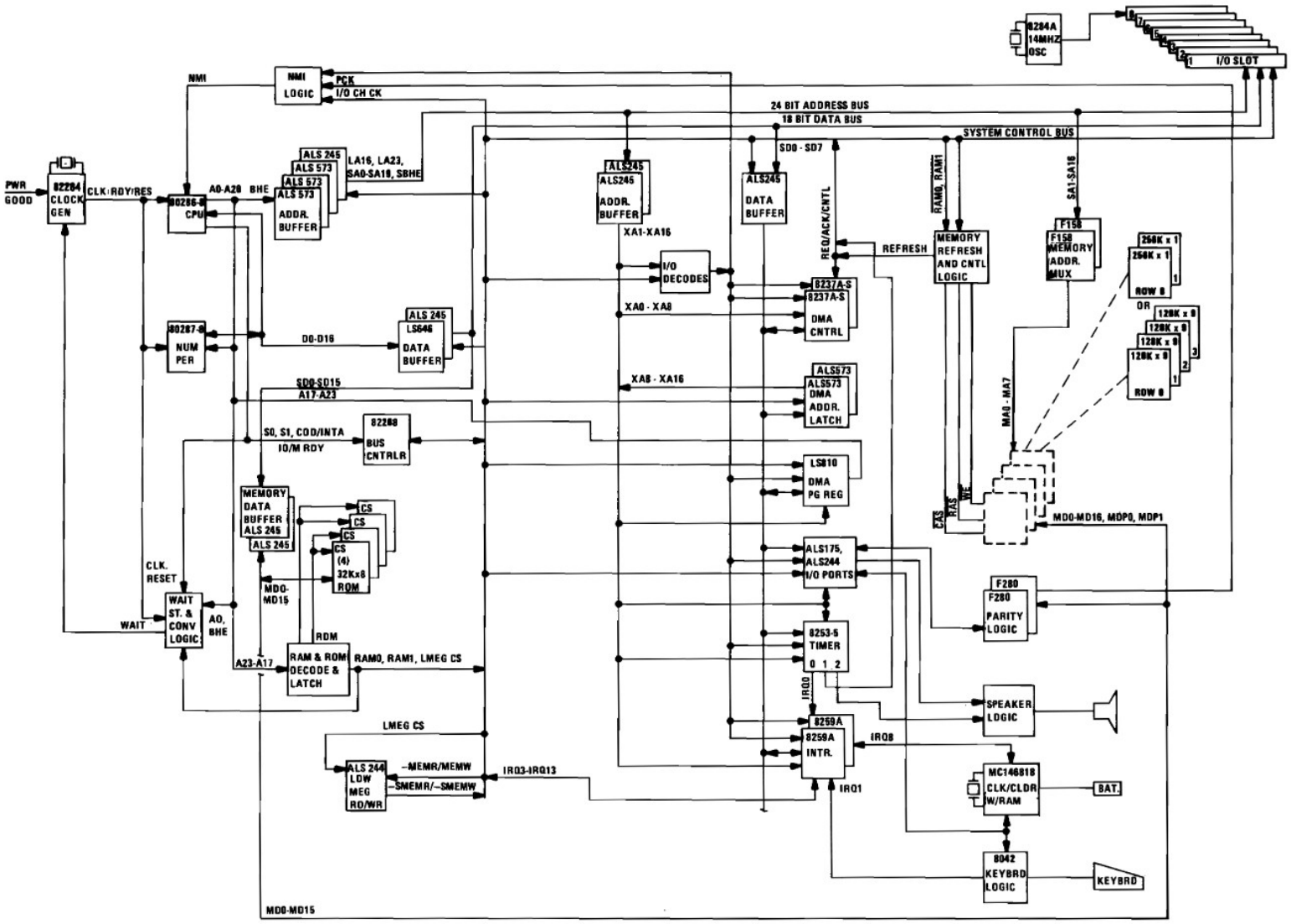
IBM PC/AT (IBM 5170) - 1984



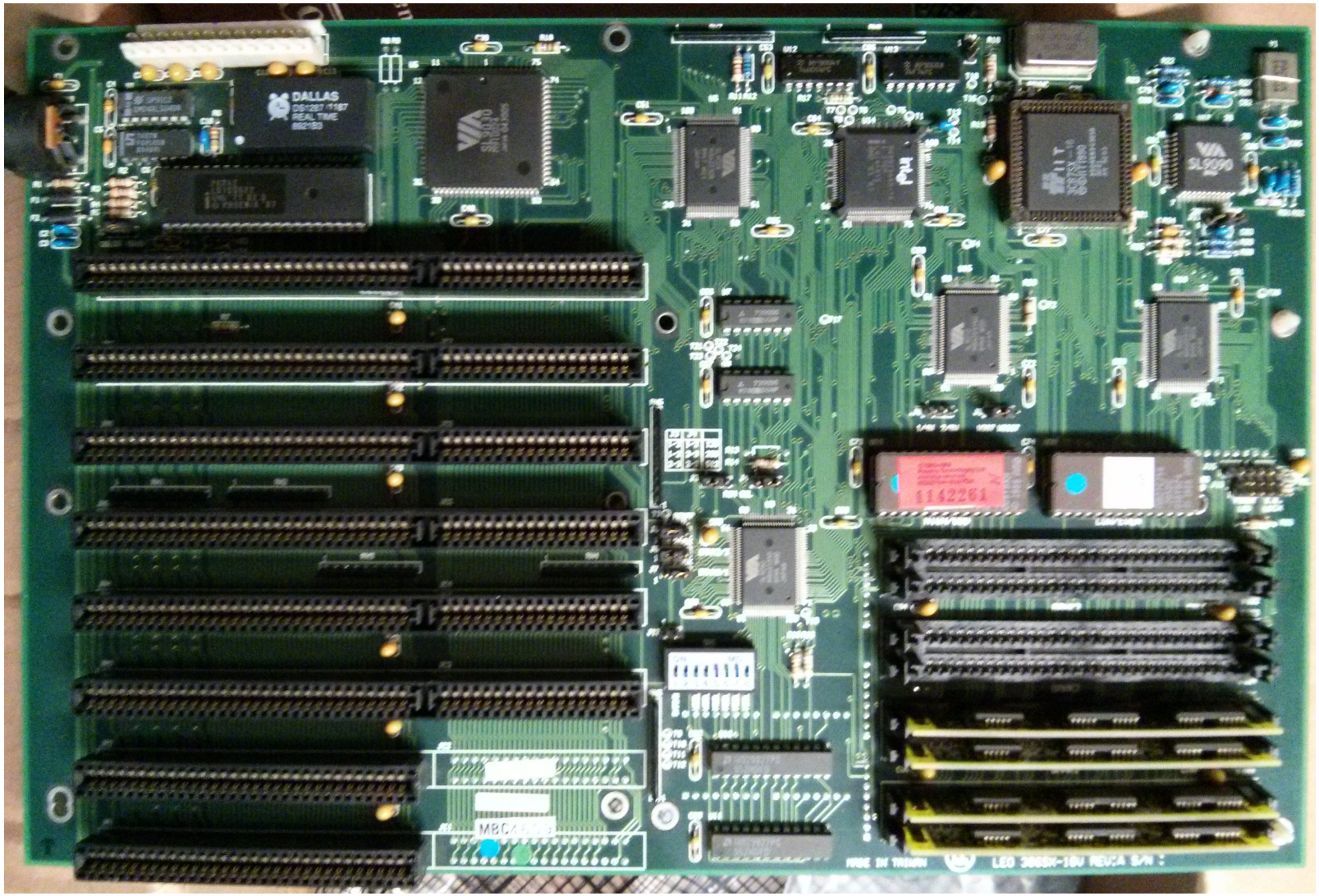
IBM PC Hardware and Technical references



IBM PC/AT Block Diagram



LEO-PC Motherboard - 1990



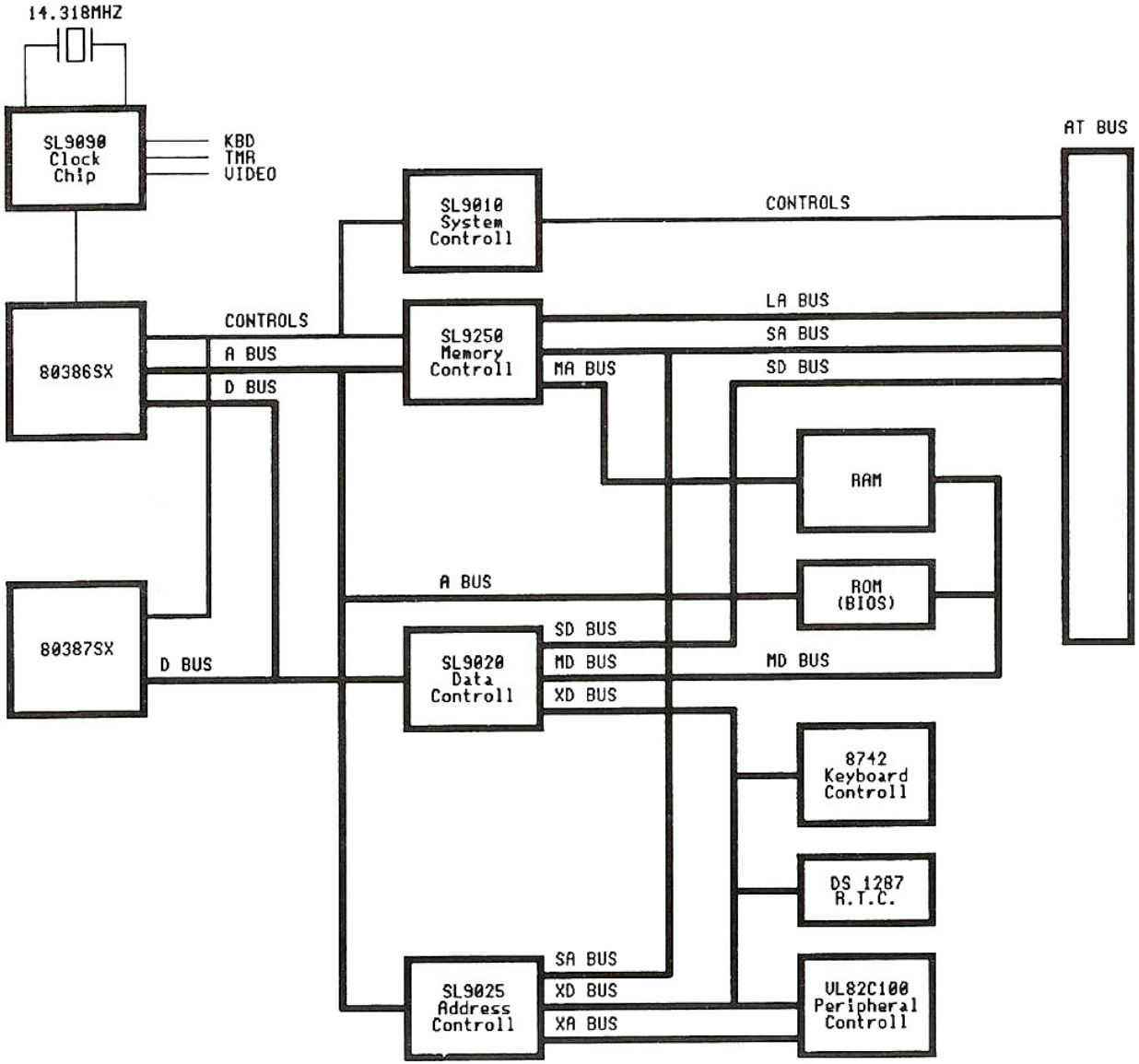
LEO-PC in PC Magazine

PC MAGAZINE FACT FILE

Leo 386/16
Computer Systems Corp.
229 West Grand Ave.
Bensenville, IL 60106
(312) 595-2950
(800) 284-7746
List Price: With 2MB RAM, 1.2MB 5¼-inch floppy disk drive, $2,499; with 80MB hard disk, monochrome monitor, DOS 3.3, $3,565; with VGA monitor, $4,125; with 40MB hard disk, $3,825. 150MB hard disk, $1,999; 60MB tape backup, $815.
In Short: The Leo 386/16 zero-wait-state tower system showed some promise in benchmark testing, but technical shortcomings and design quirks held it back from real competition with the rest of the pack.
CIRCLE **405** ON READER SERVICE CARD

The Leo 386/16, from Computer Systems Corp., is an affordably priced, zero-wait-state tower system. The use of mostly standard components in a sturdy case is commendable, but the front-panel turbo and reset buttons look alike and are dangerously close to each other.

LEO-PC Block Diagram

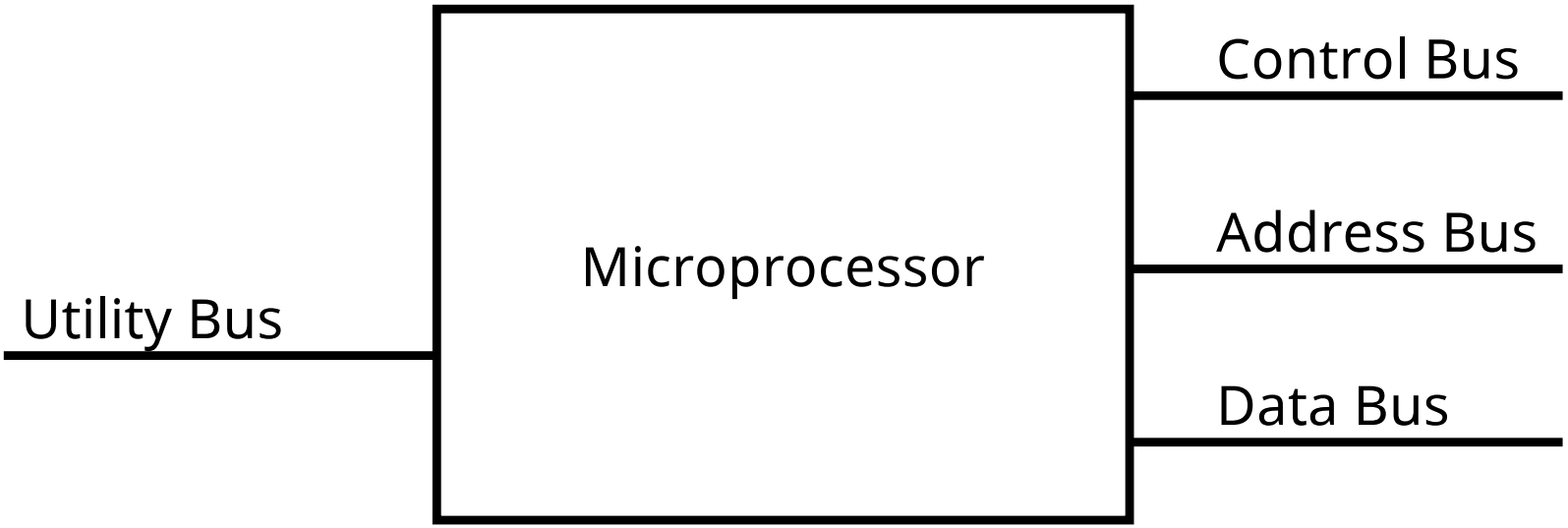


The 80386SX interface to the system

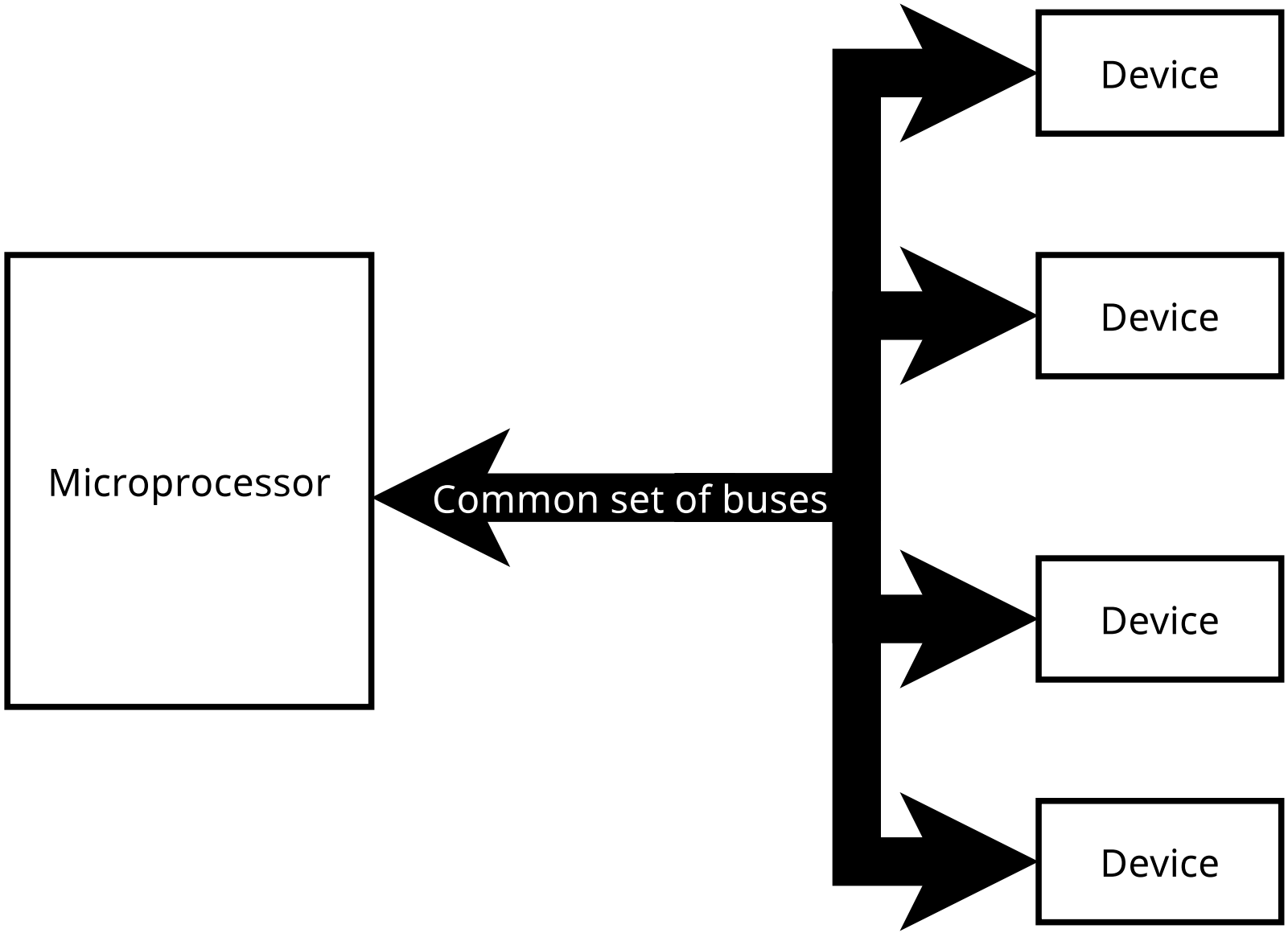
The 80386SX

Buses

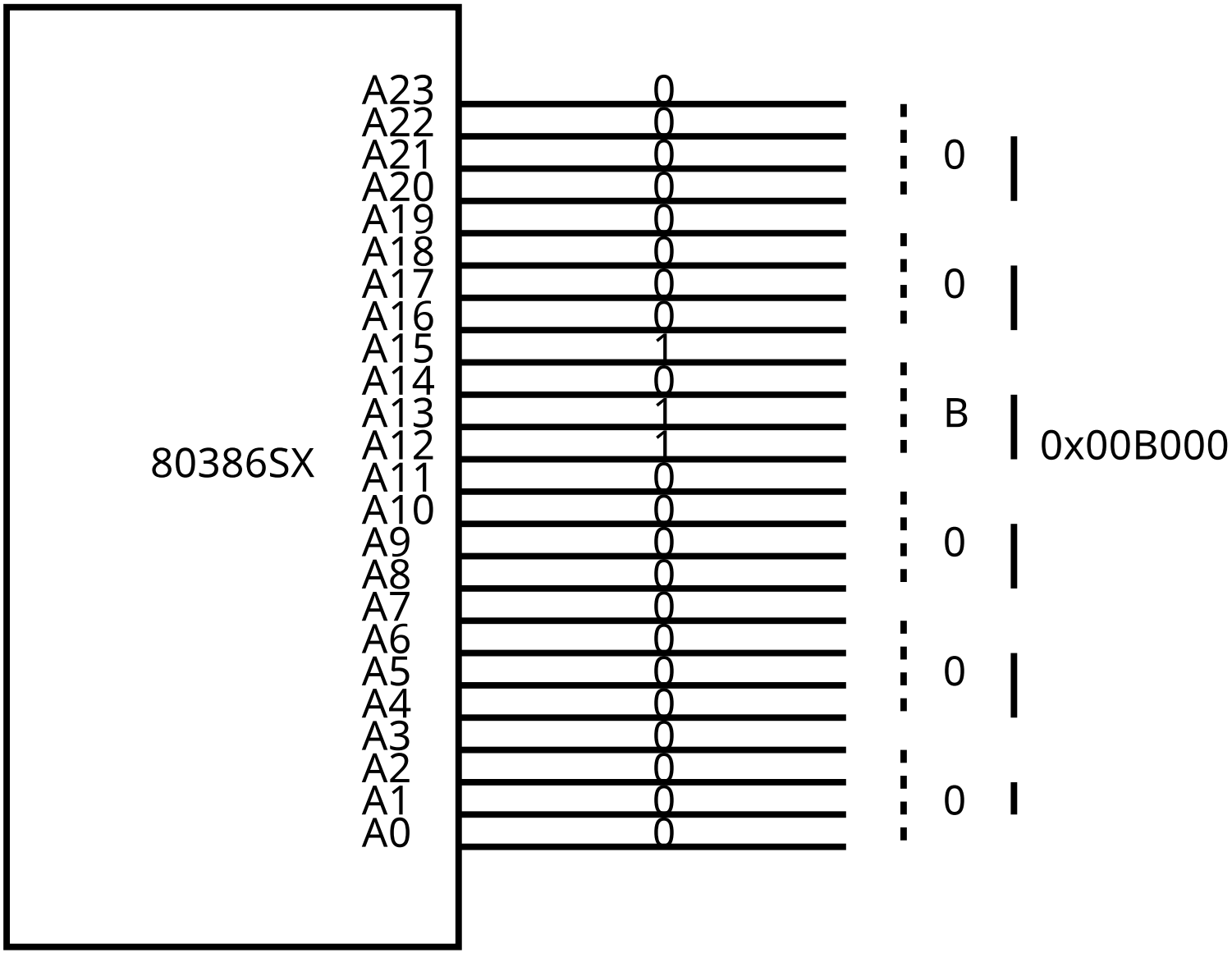
A bus is a list of signals coming in or out of the processor.



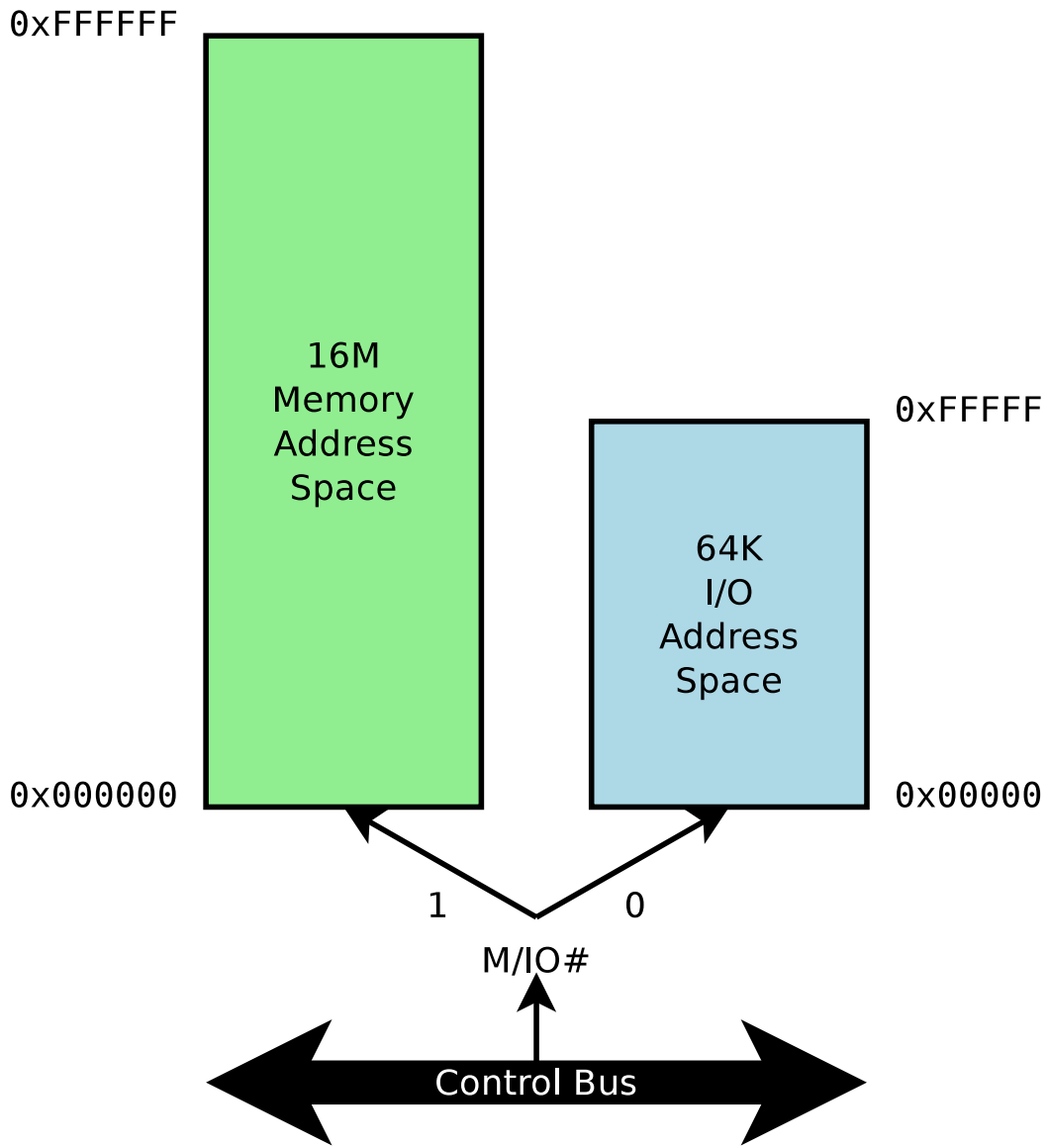
CPU and Devices



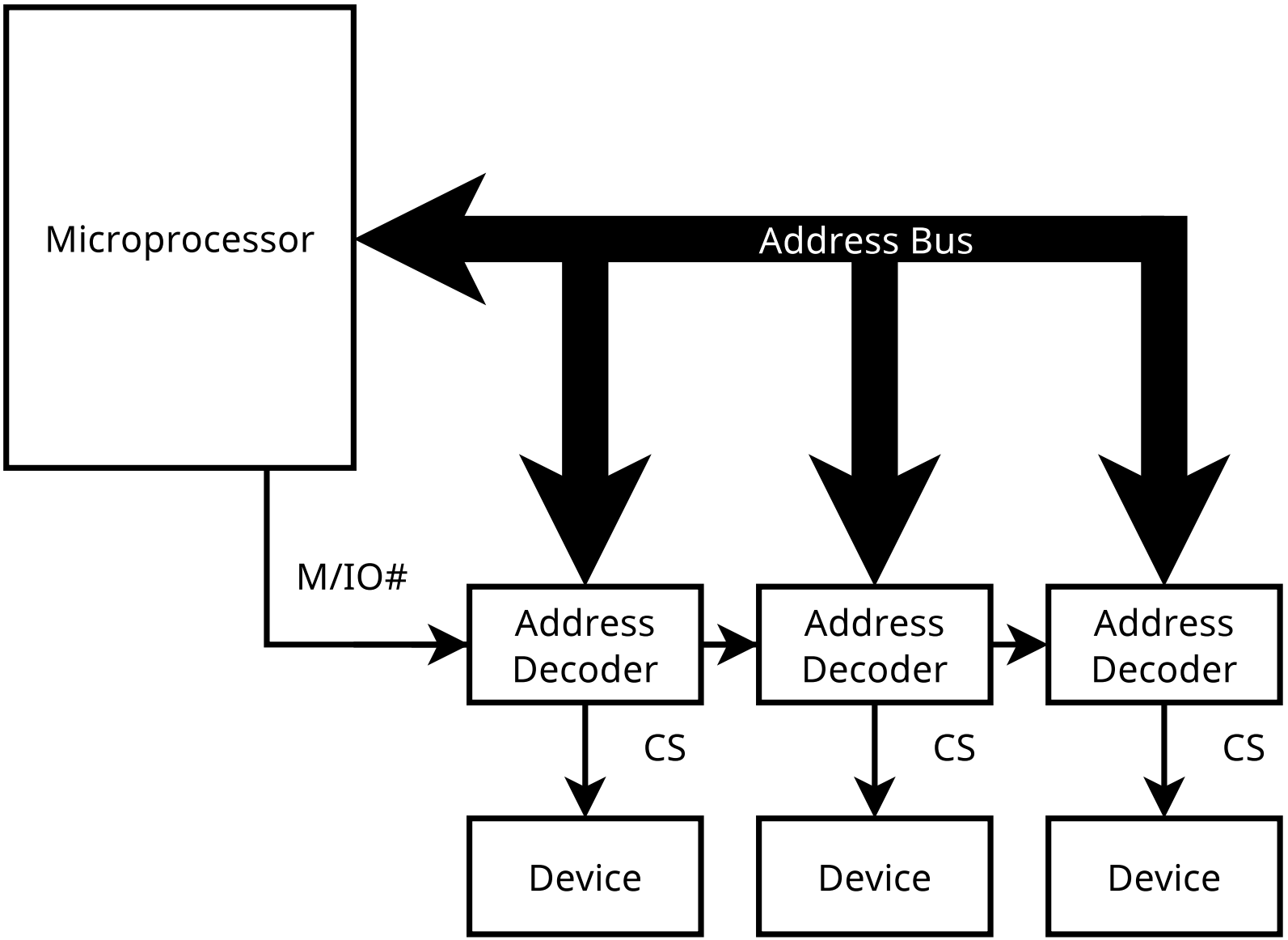
80386SX: Address Bus



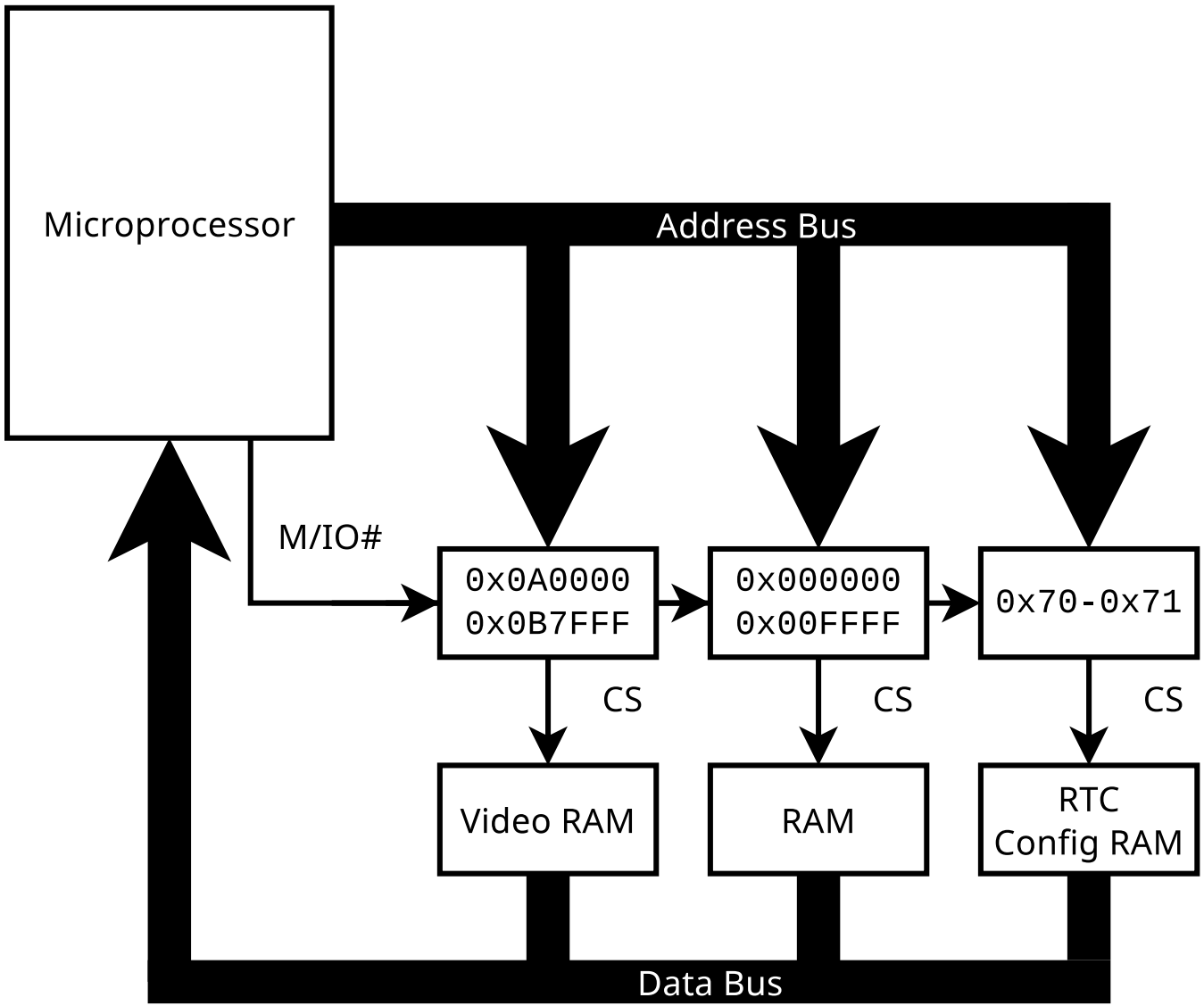
80386SX: Address Spaces



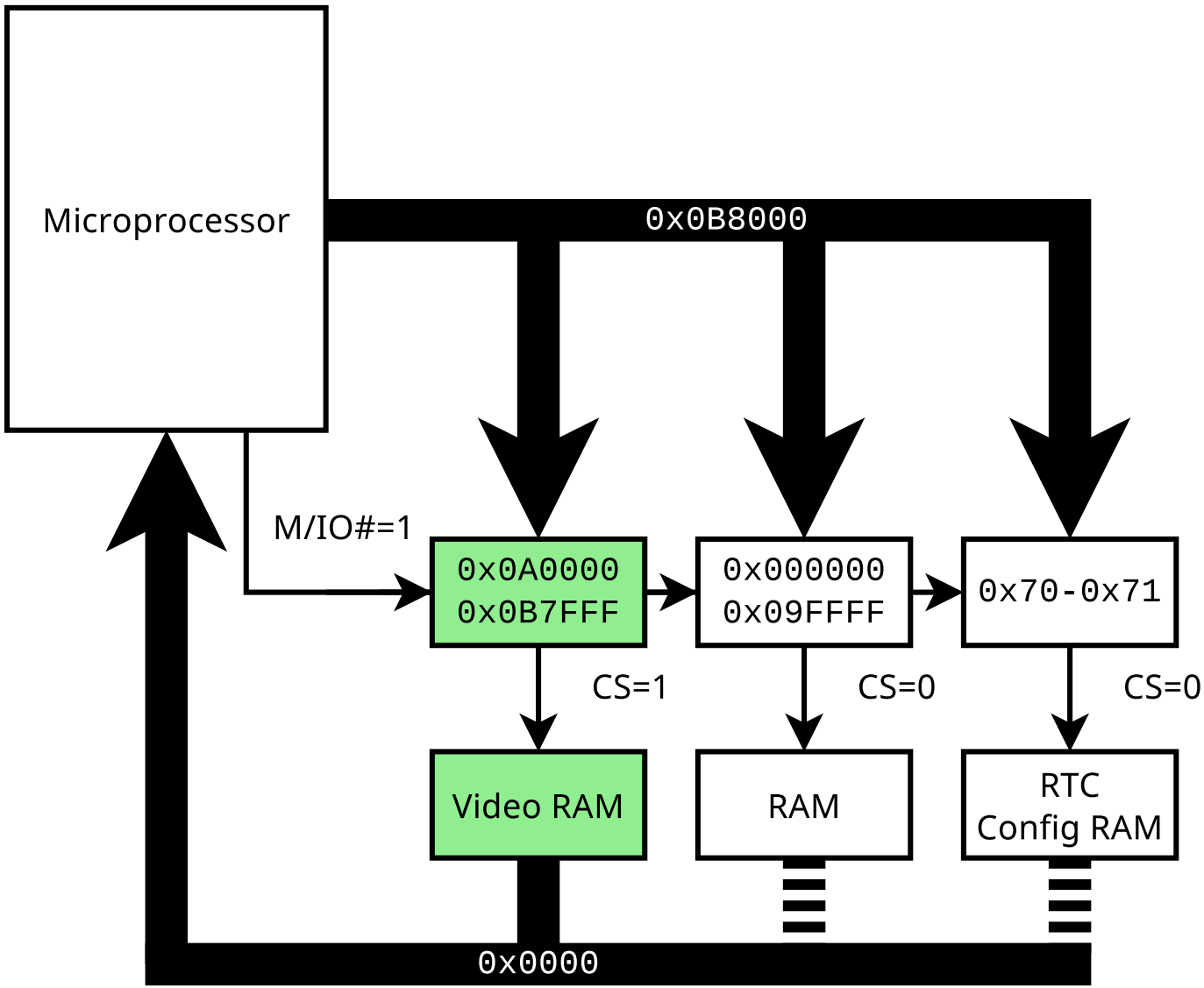
Address Decoders



Address Decoders: Read to Video RAM



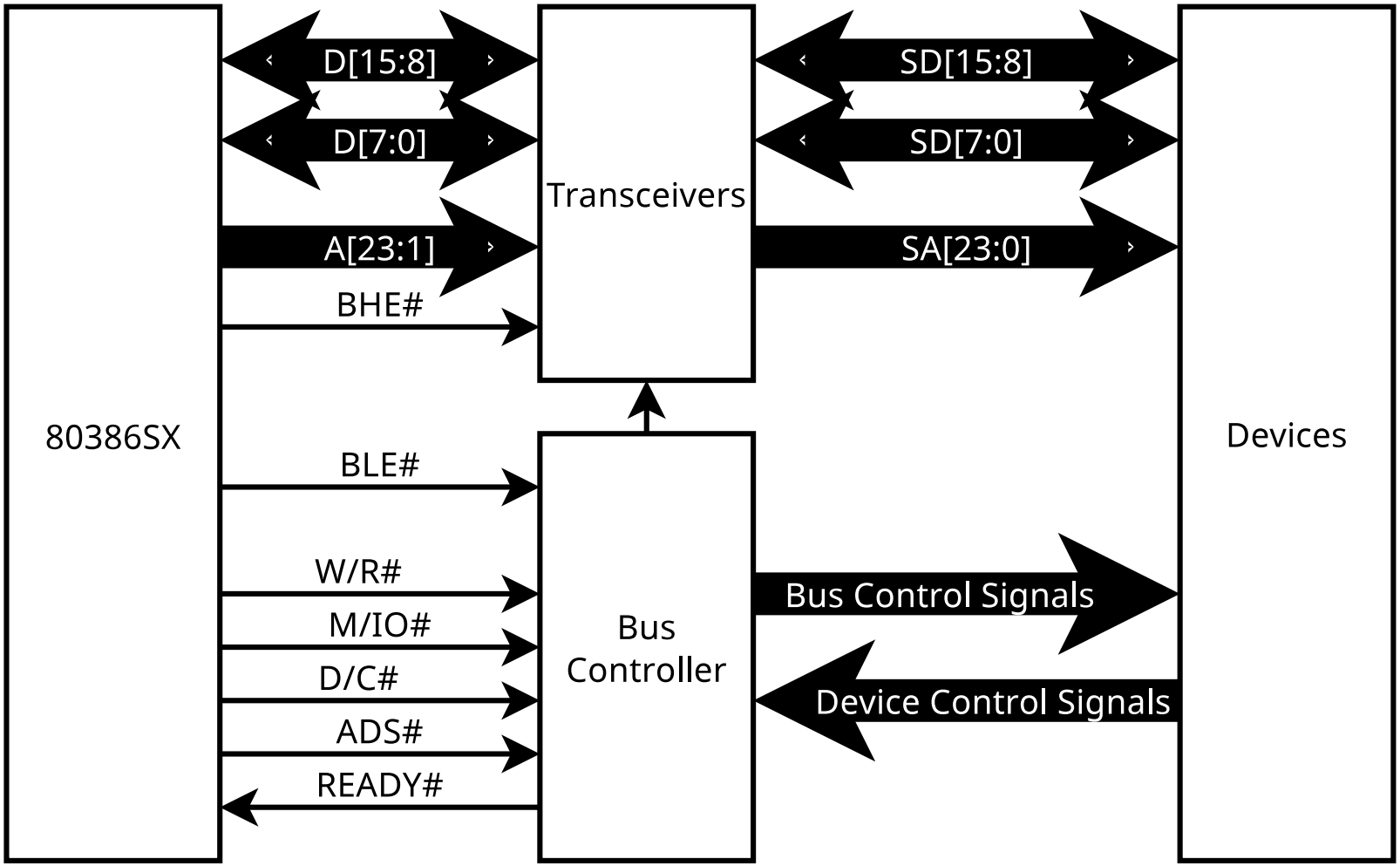
Address Decoders: Read to Video RAM



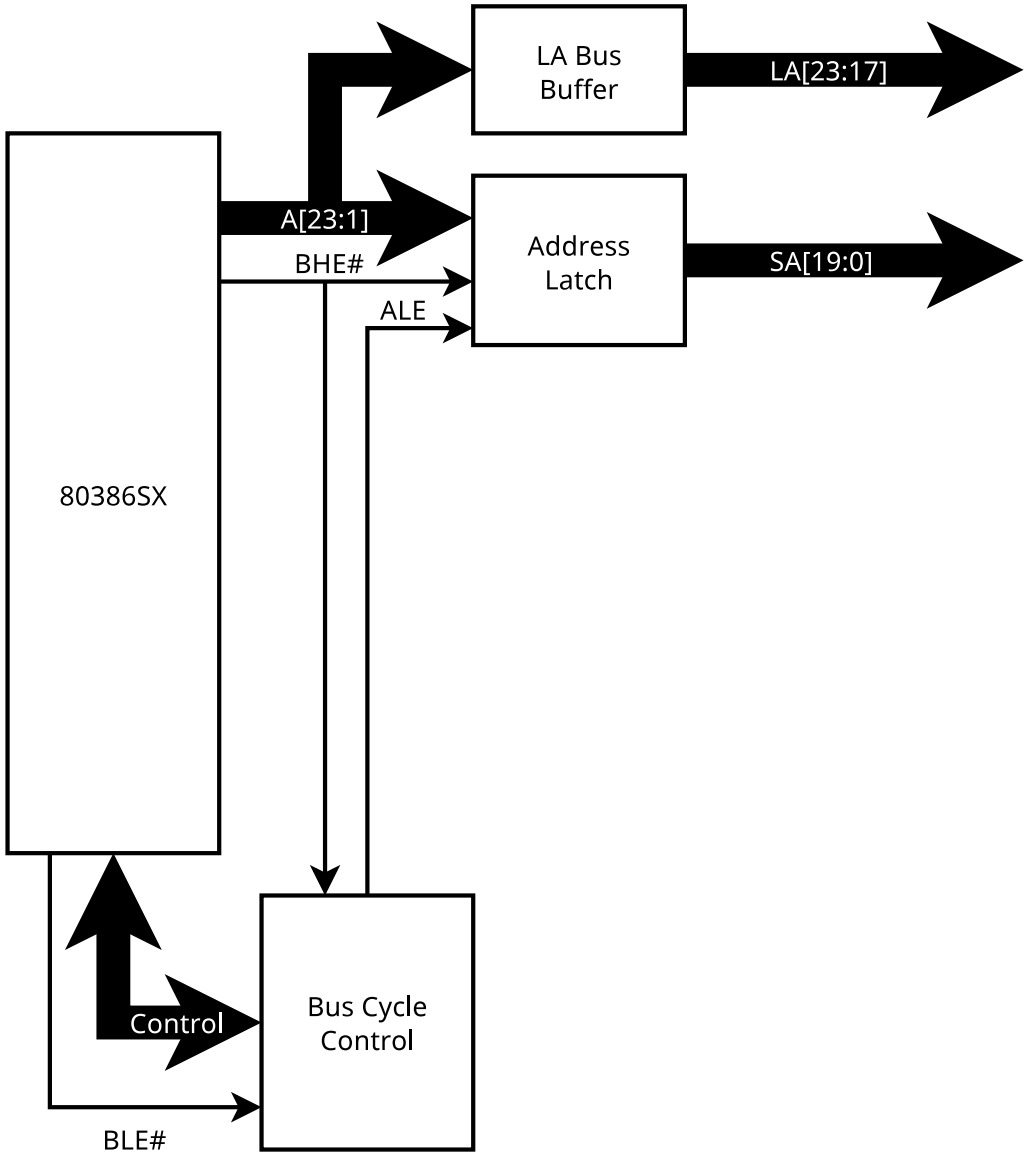
80386SX: Interface Rules

1. Every memory and I/O storage location contains one byte of information, no more, no less.
2. Every memory and I/O address is considered to be either an even address or an odd address.
3. When the 80386SX reads from or writes to an even address, the data is transfered over the lower data path: `D[7:0]`. When reading or writing to an odd address, the data is transferred over the upper data path: `D[15:8]`.

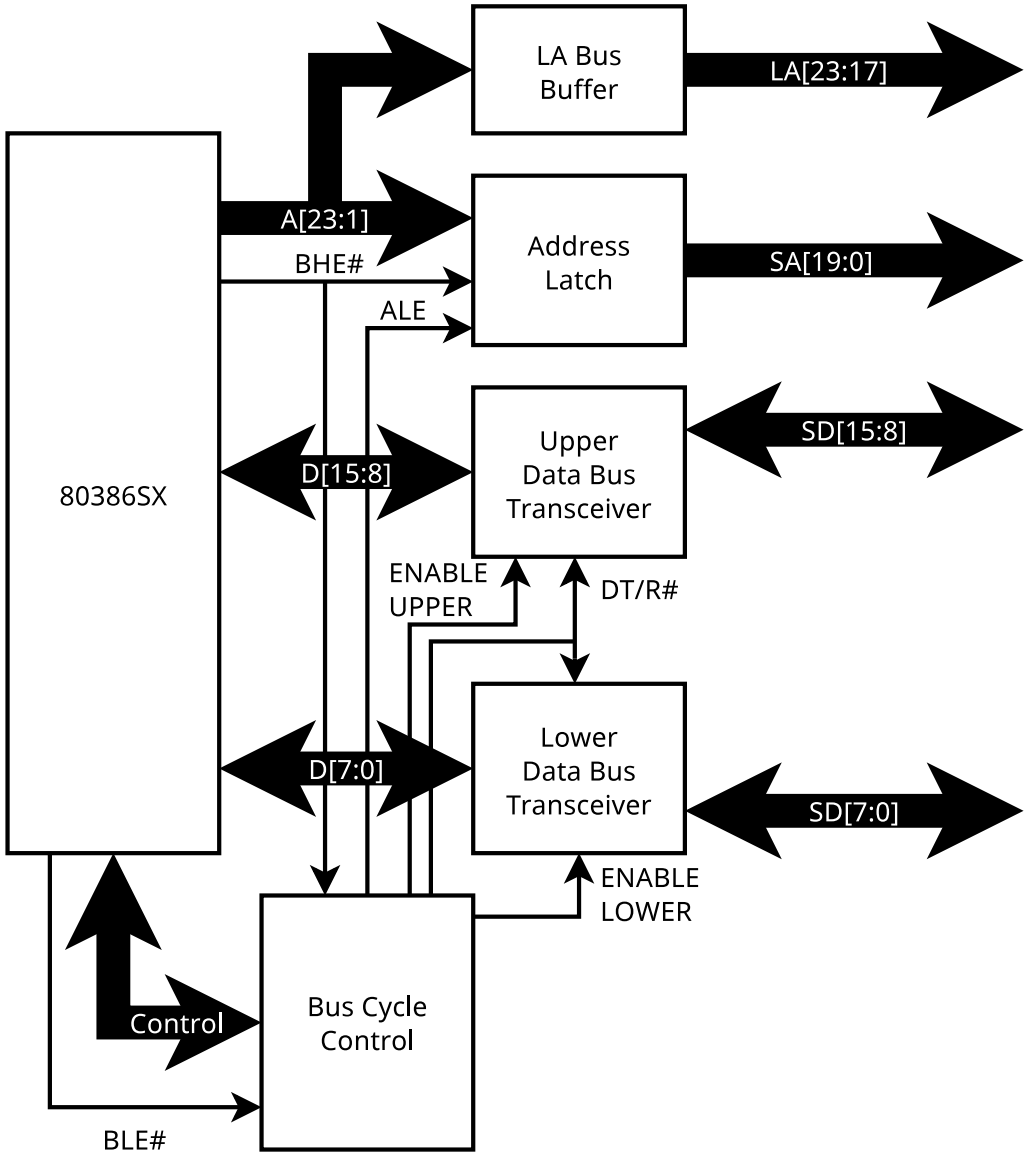
80386SX: Interface diagram



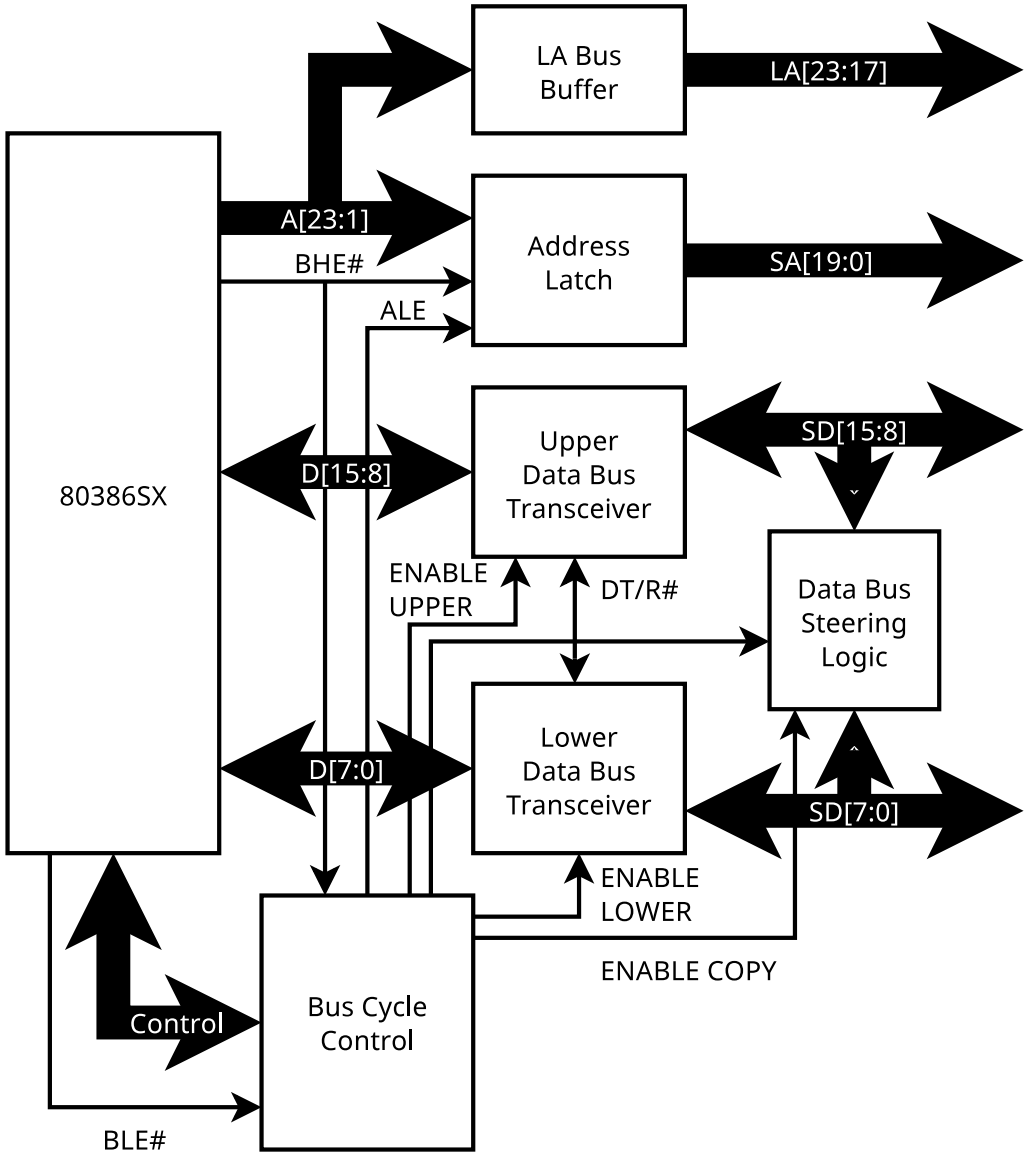
Address Latch



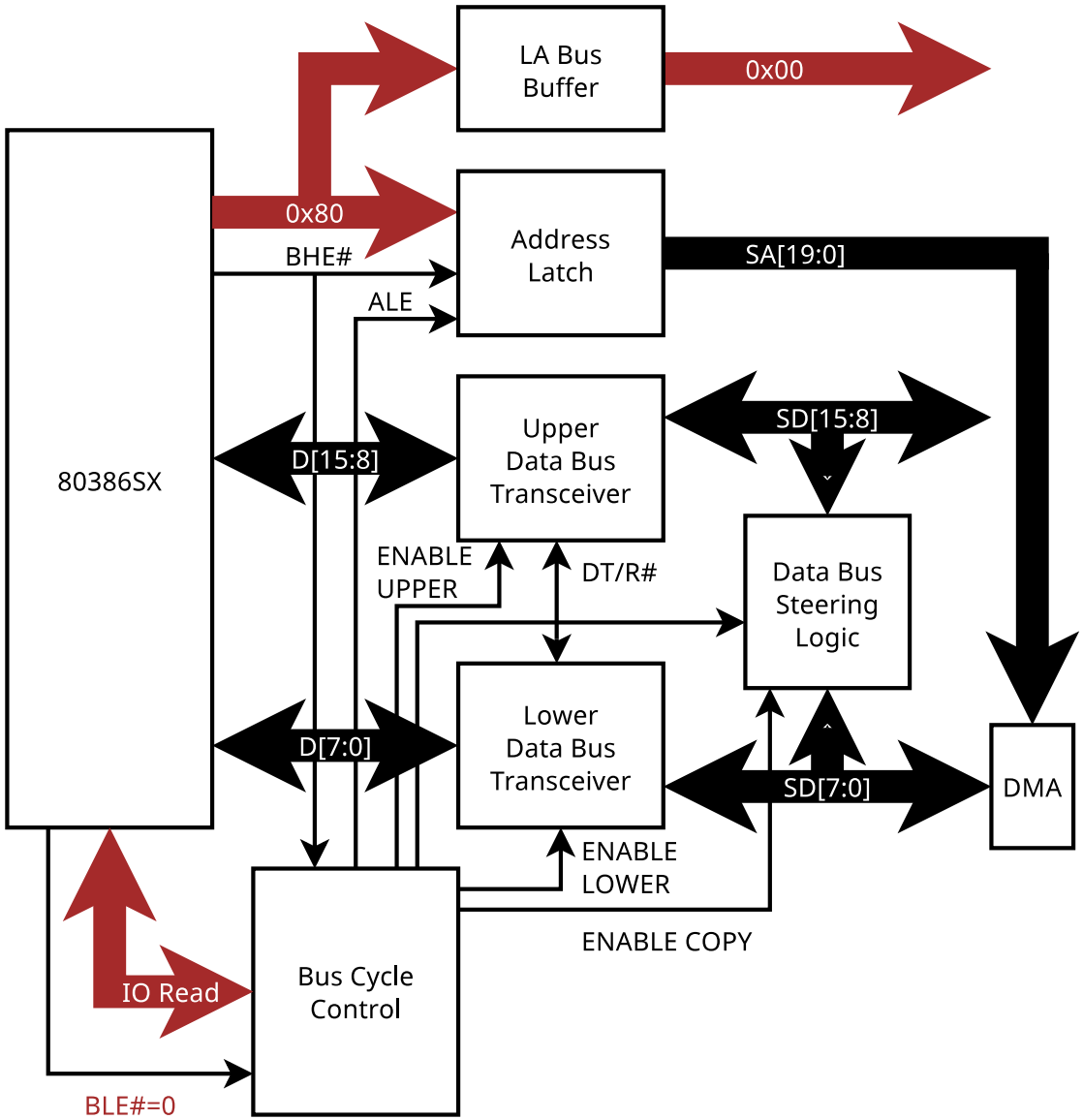
Data bus transceivers



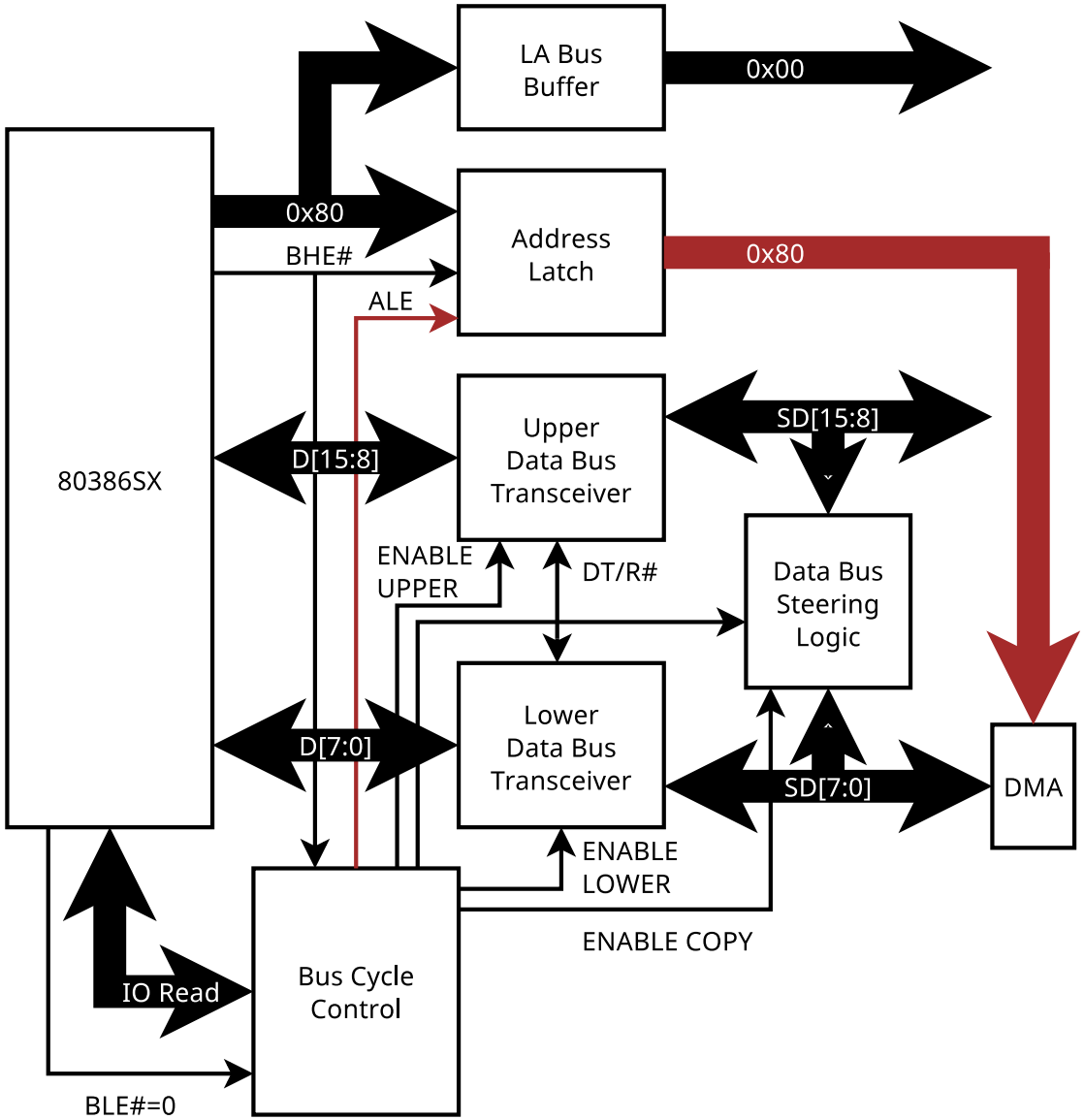
Data bus steering



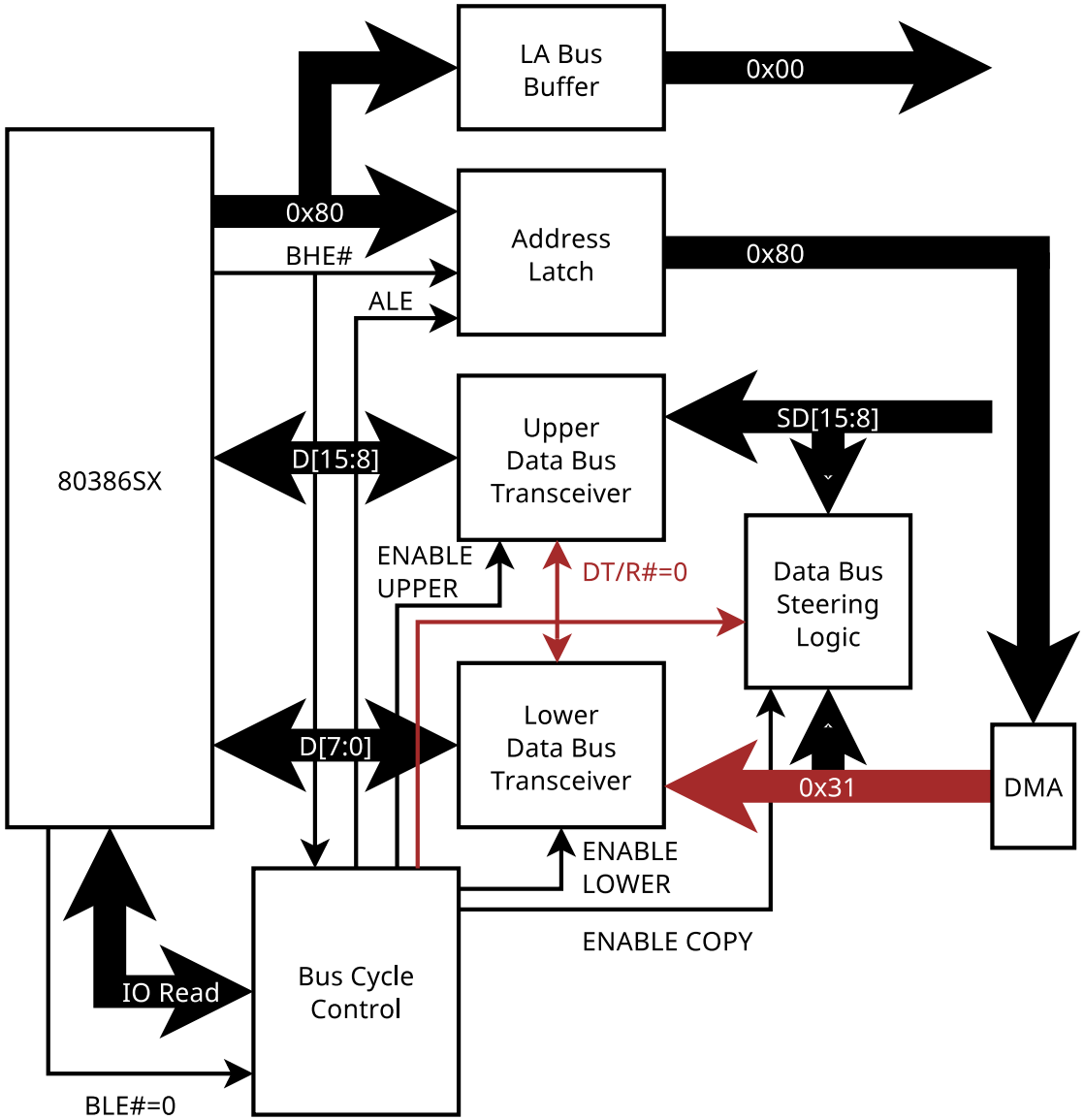
in al, 0x80 (1/4)



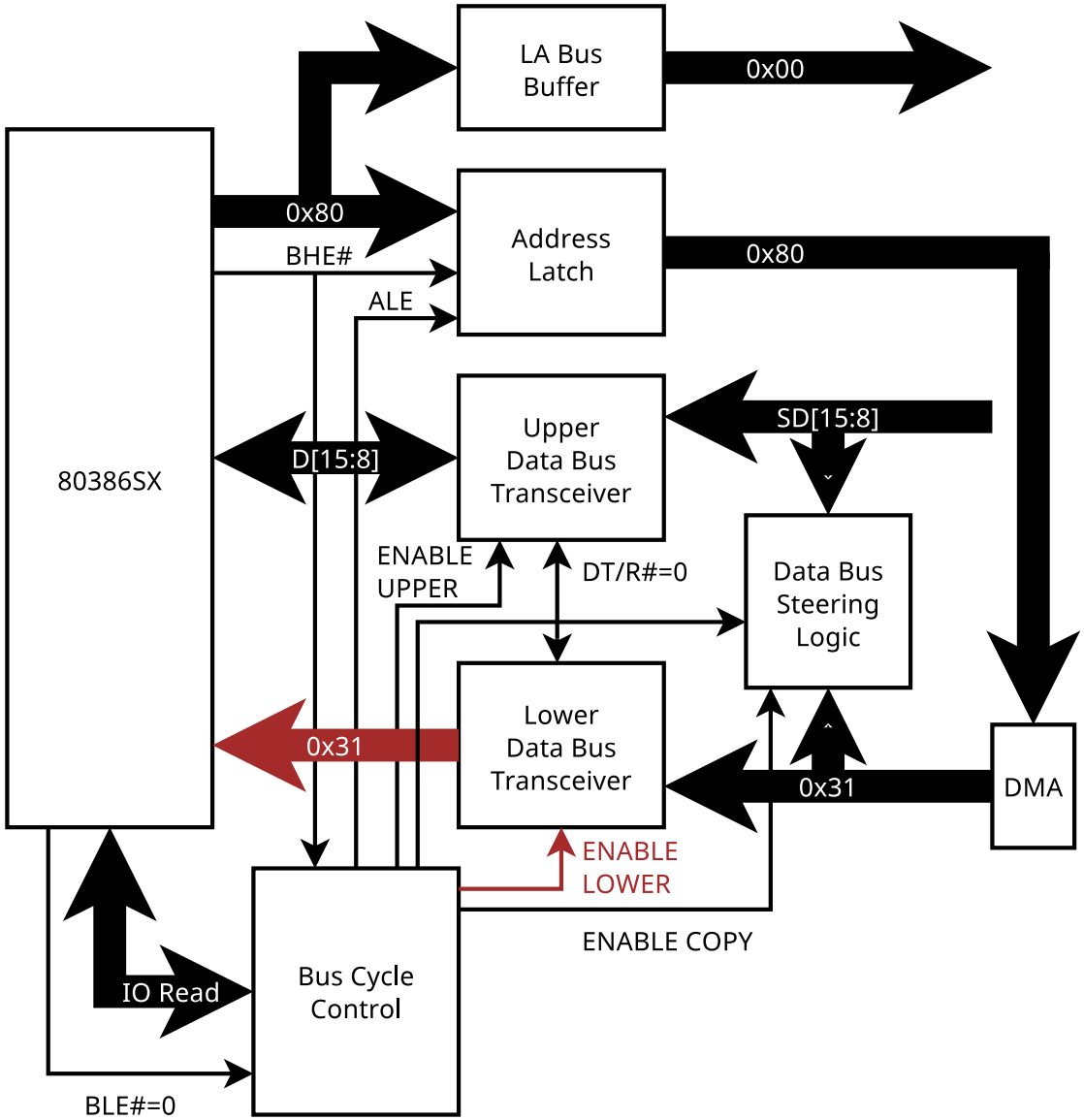
in al, 0x80 (2/4)



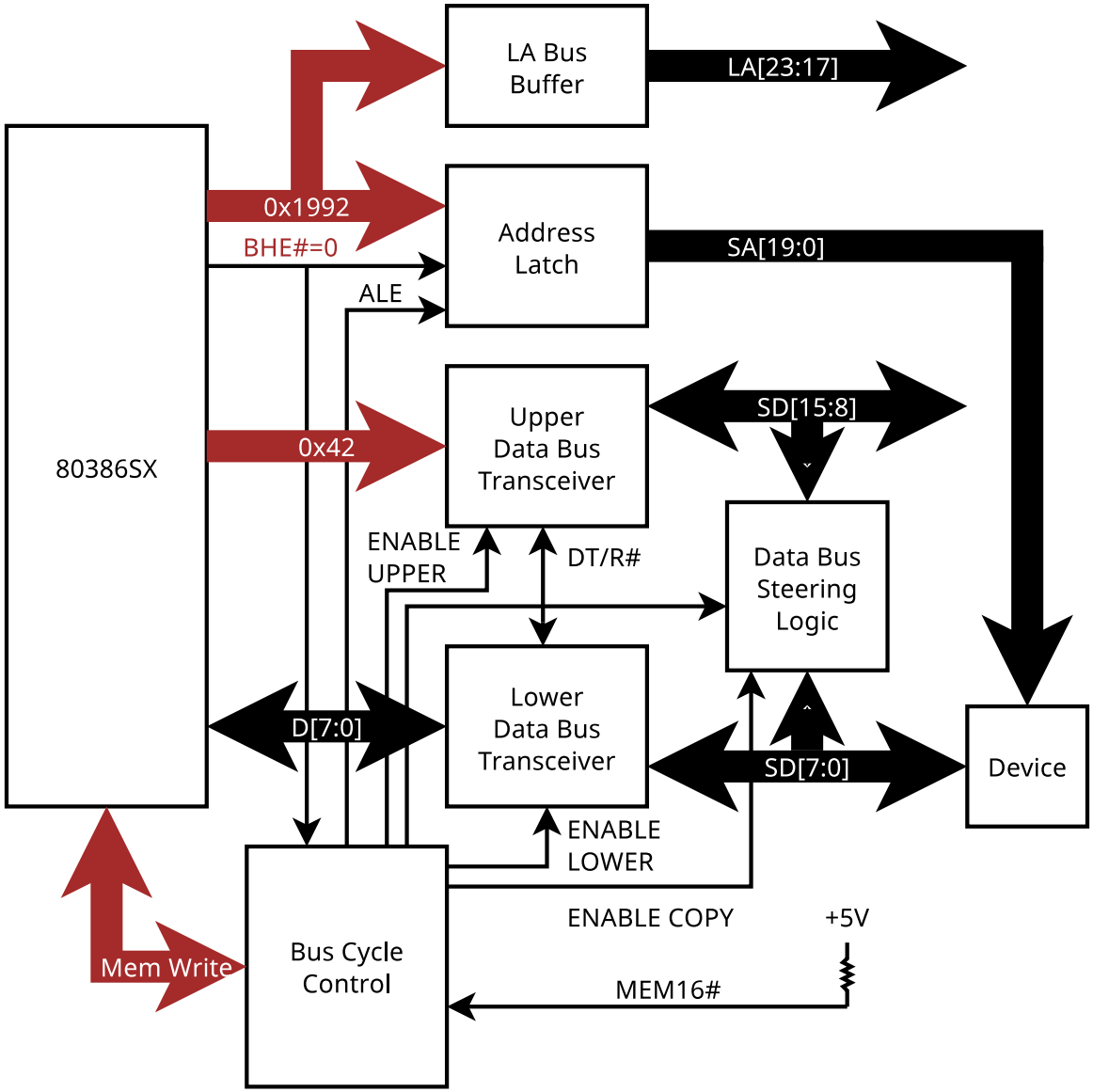
in al, 0x80 (3/4)



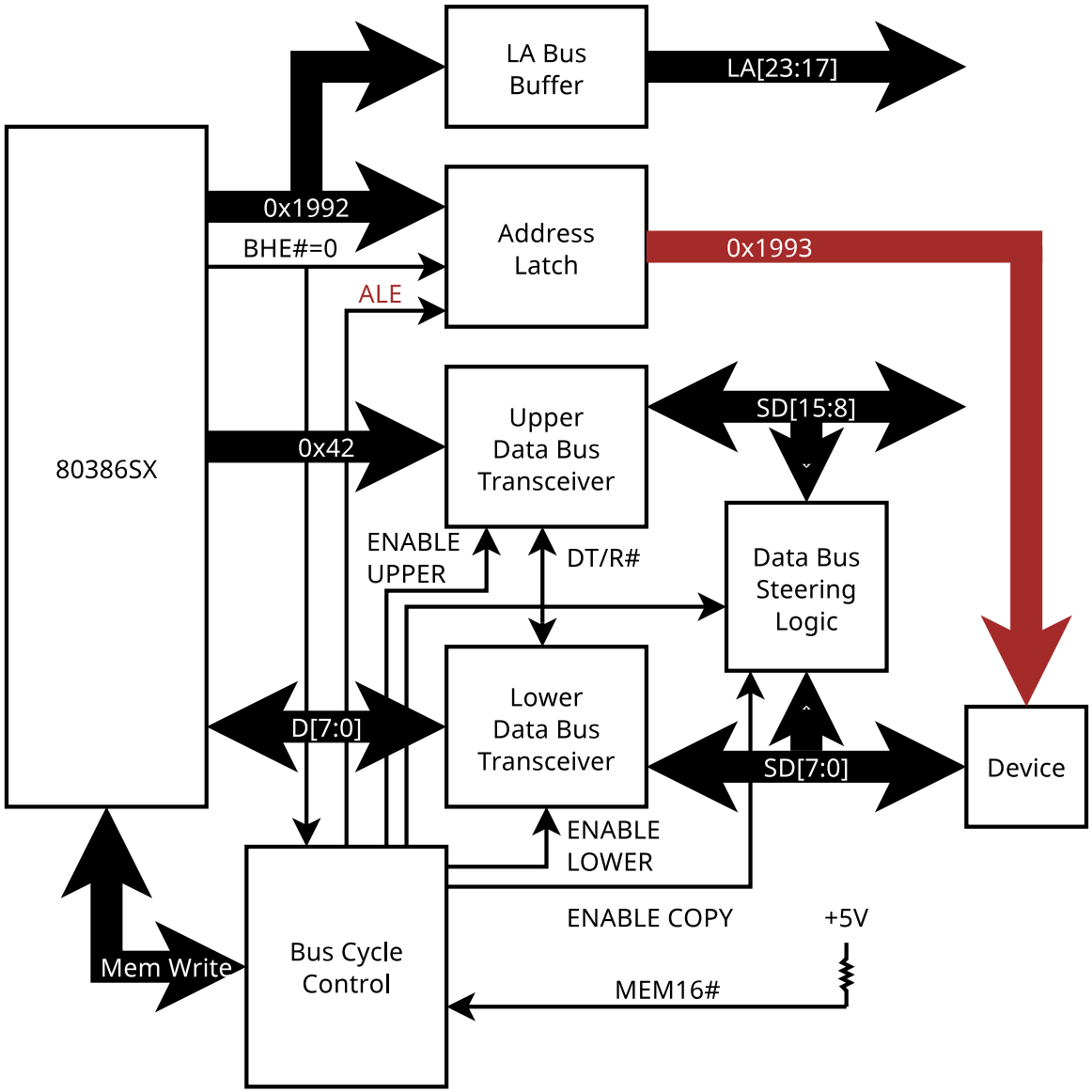
in al, 0x80 (4/4)



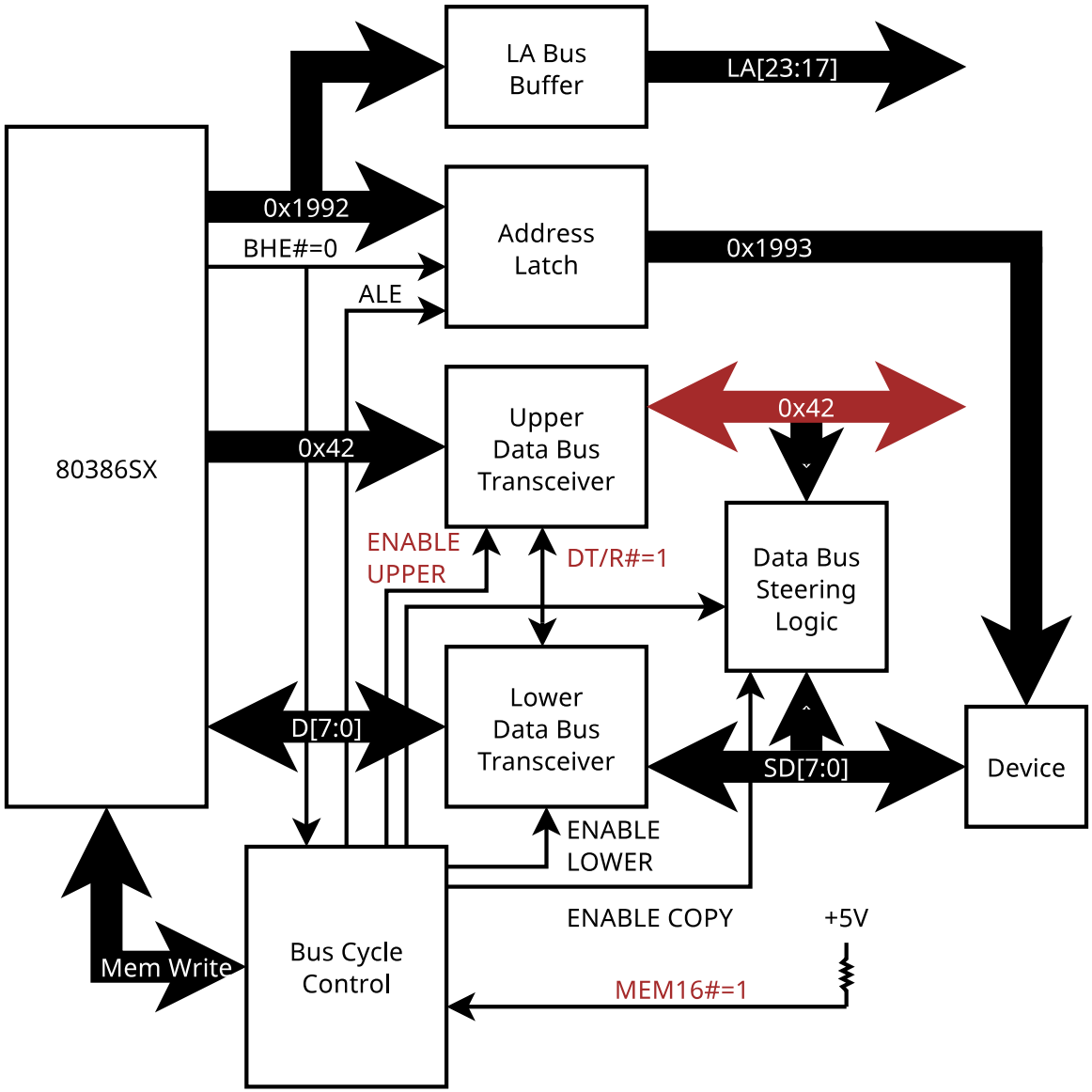
mov [0x1993], al (1/4)



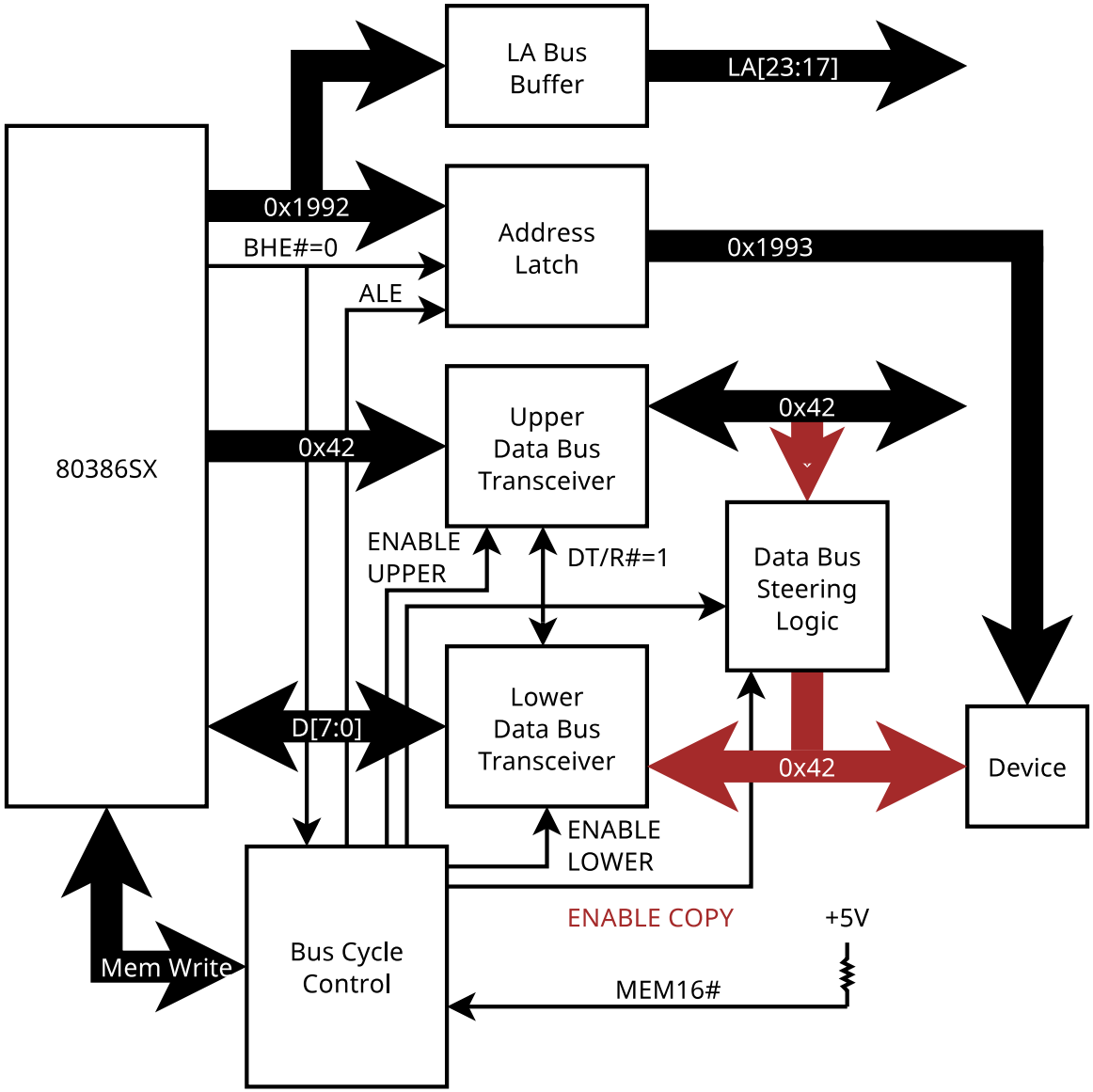
mov [0x1993], al (2/4)



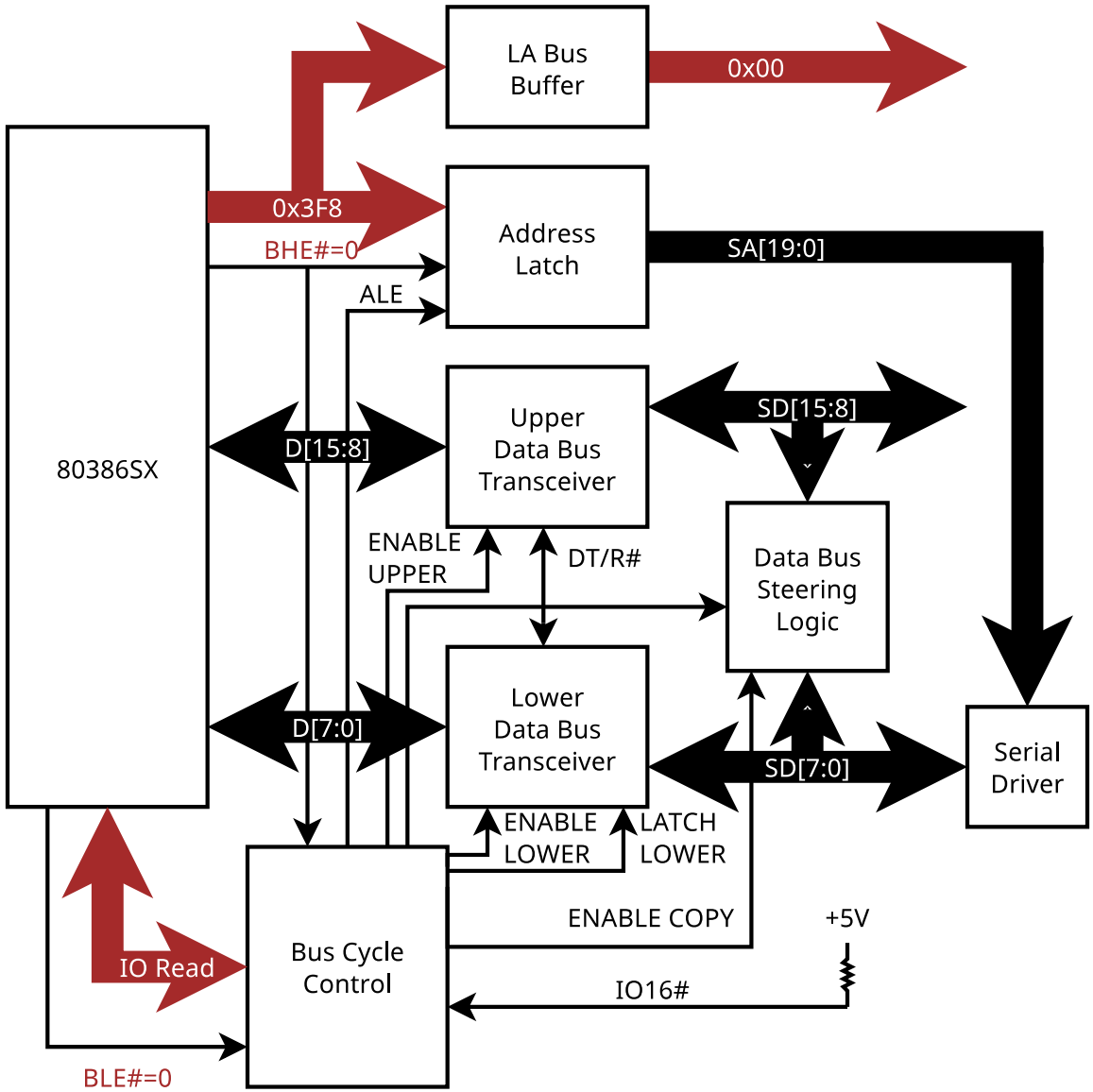
mov [0x1993], al (3/4)



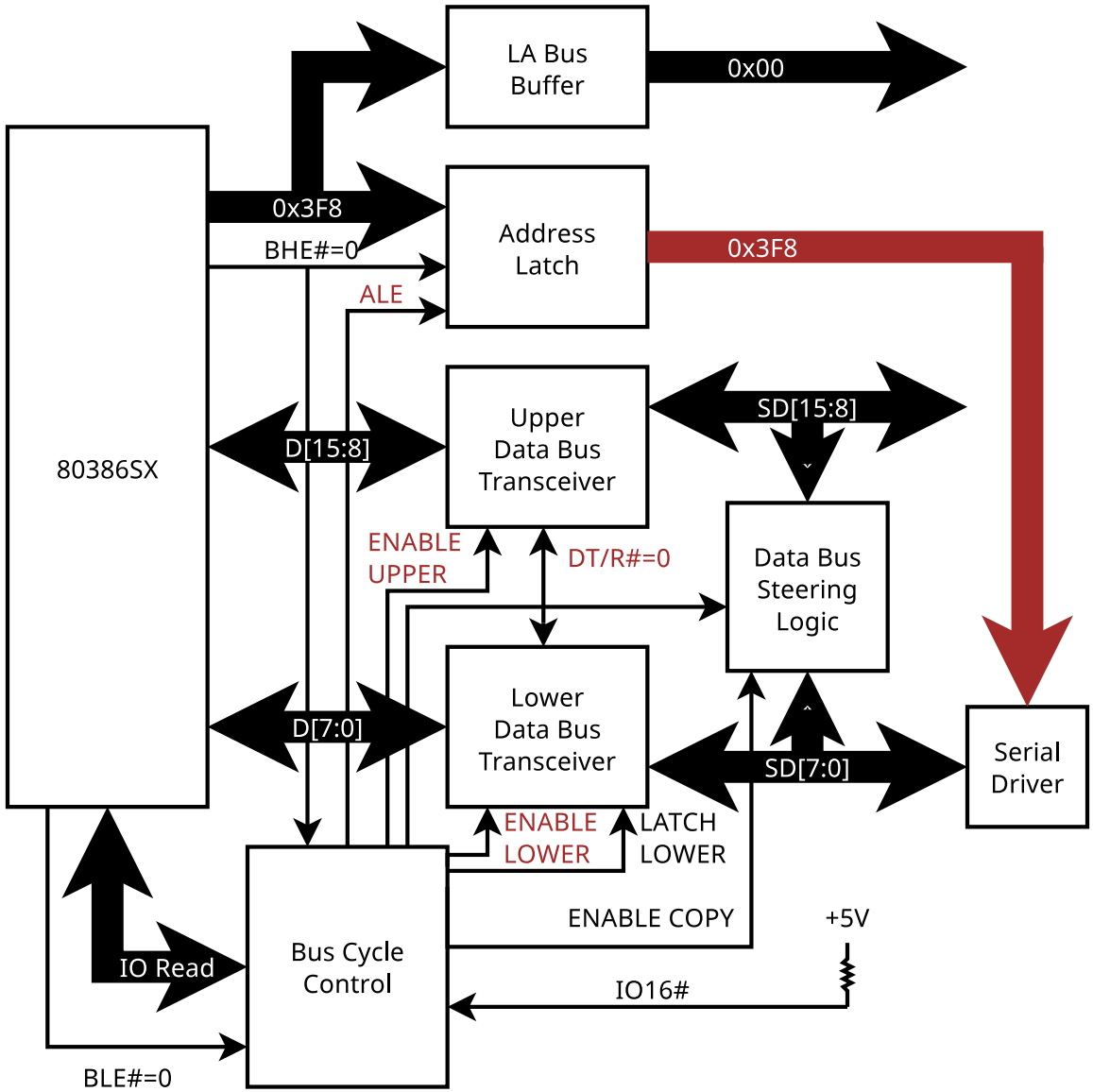
mov [0x1993], al (4/4)



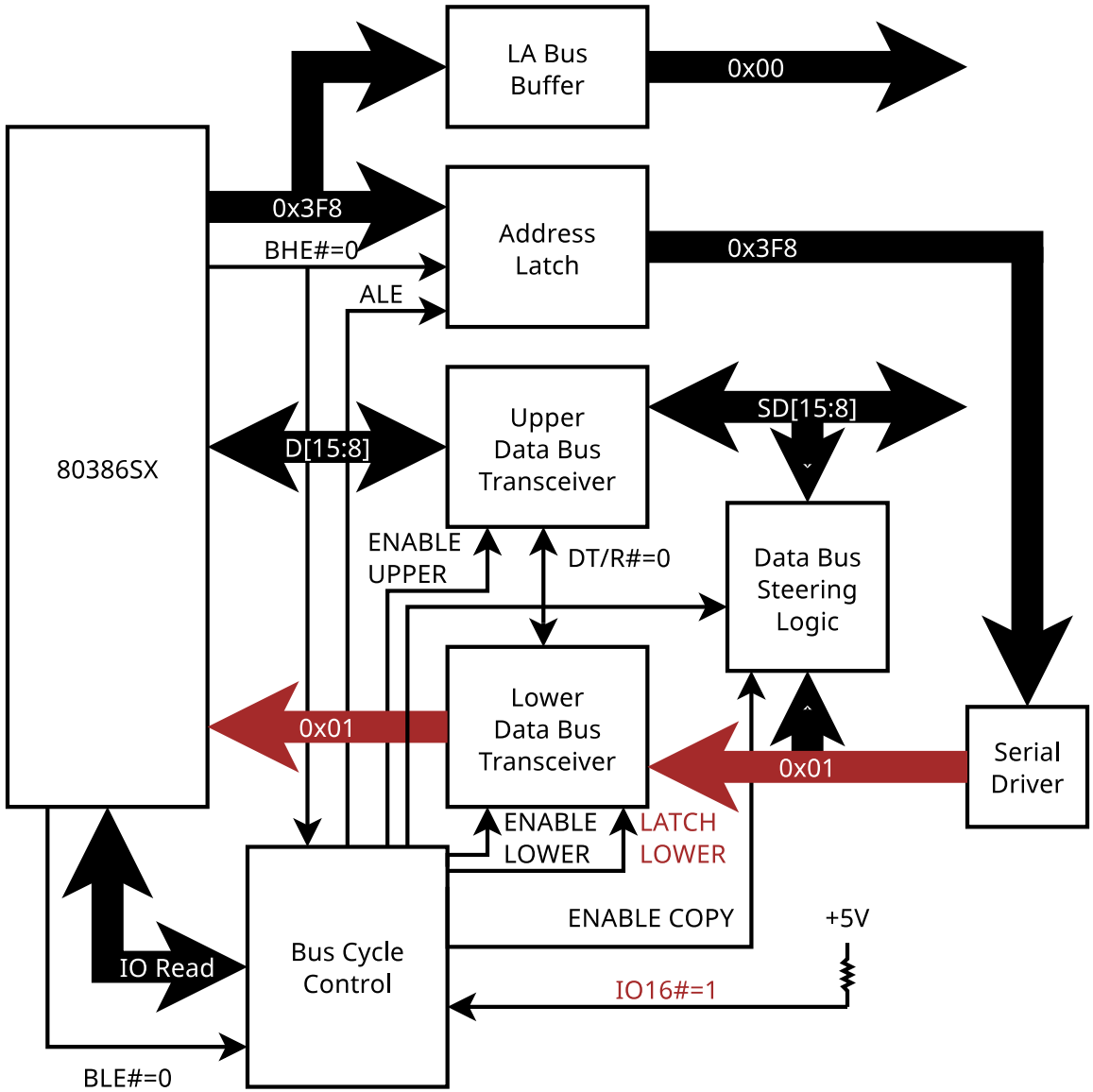
mov dx, 0x3f8; in ax, dx



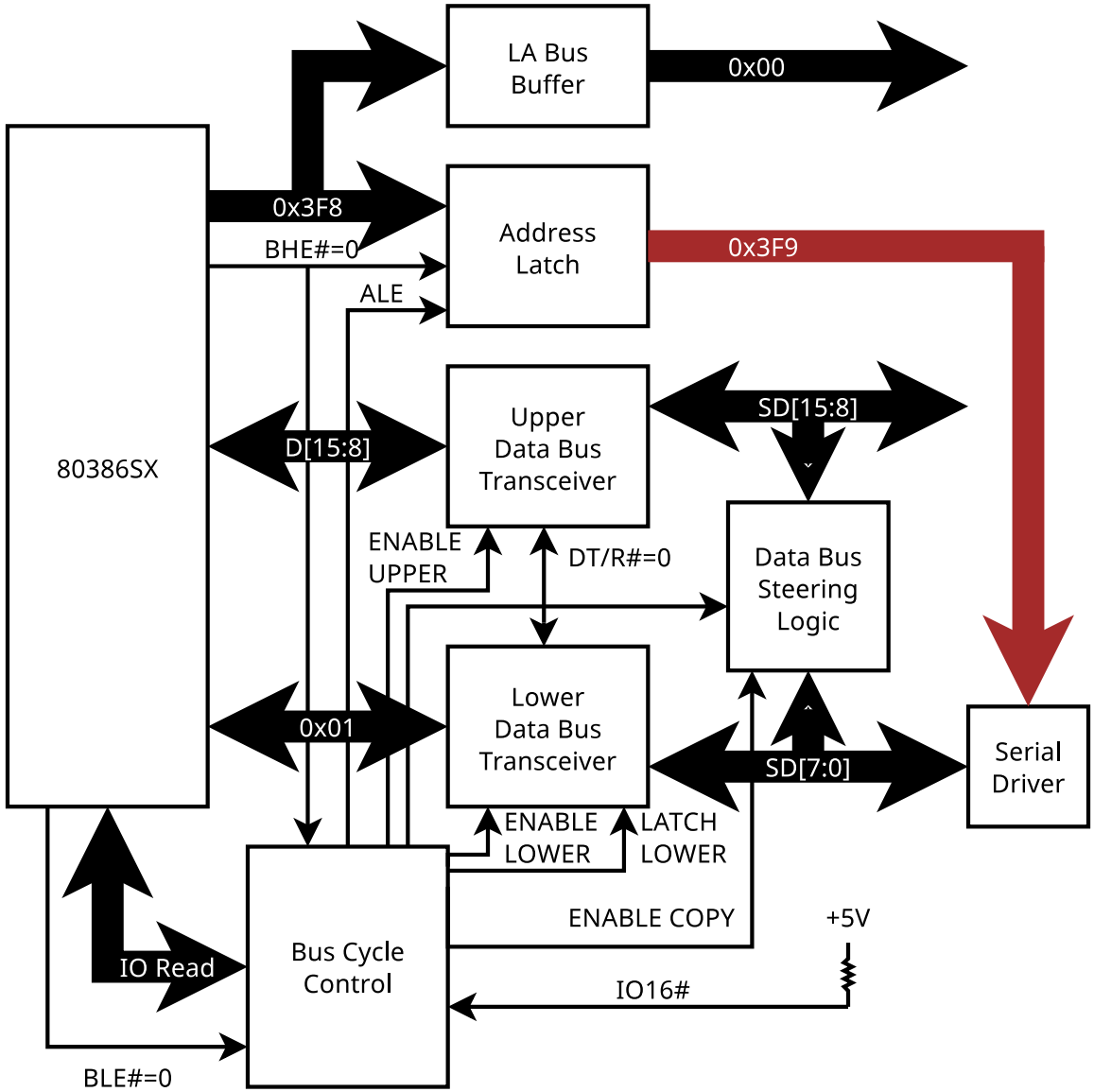
mov dx, 0x3f8; in ax, dx



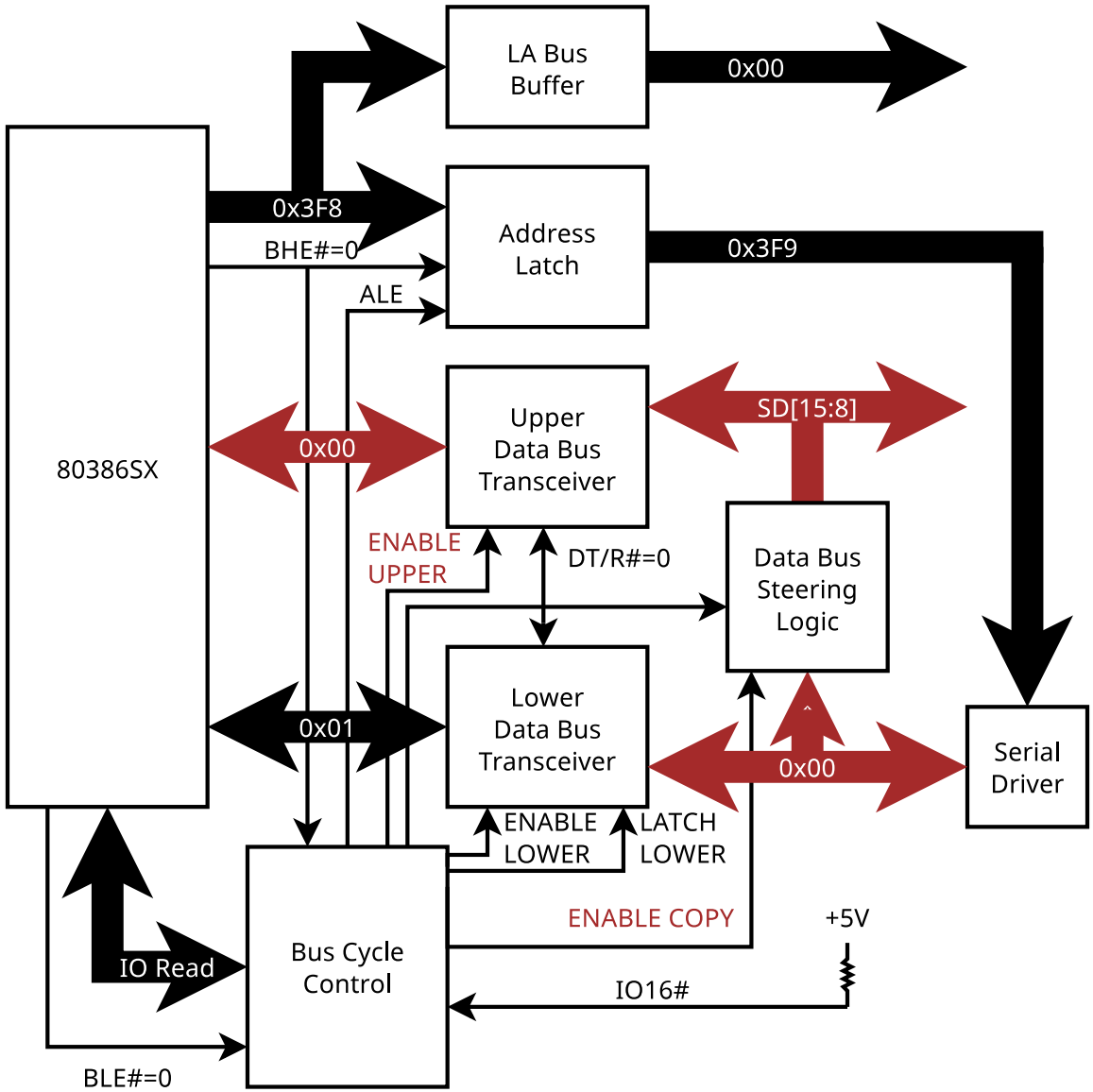
mov dx, 0x3f8; in ax, dx



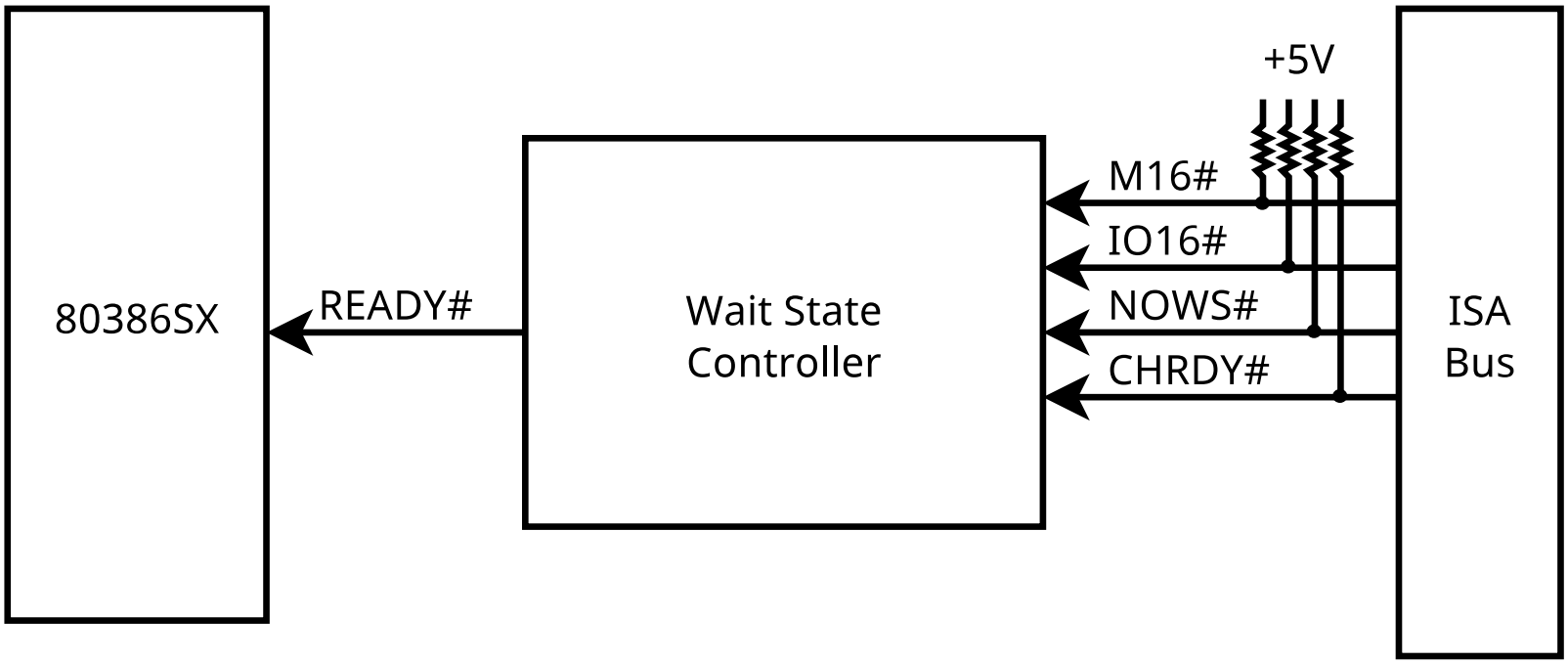
mov dx, 0x3f8; in ax, dx



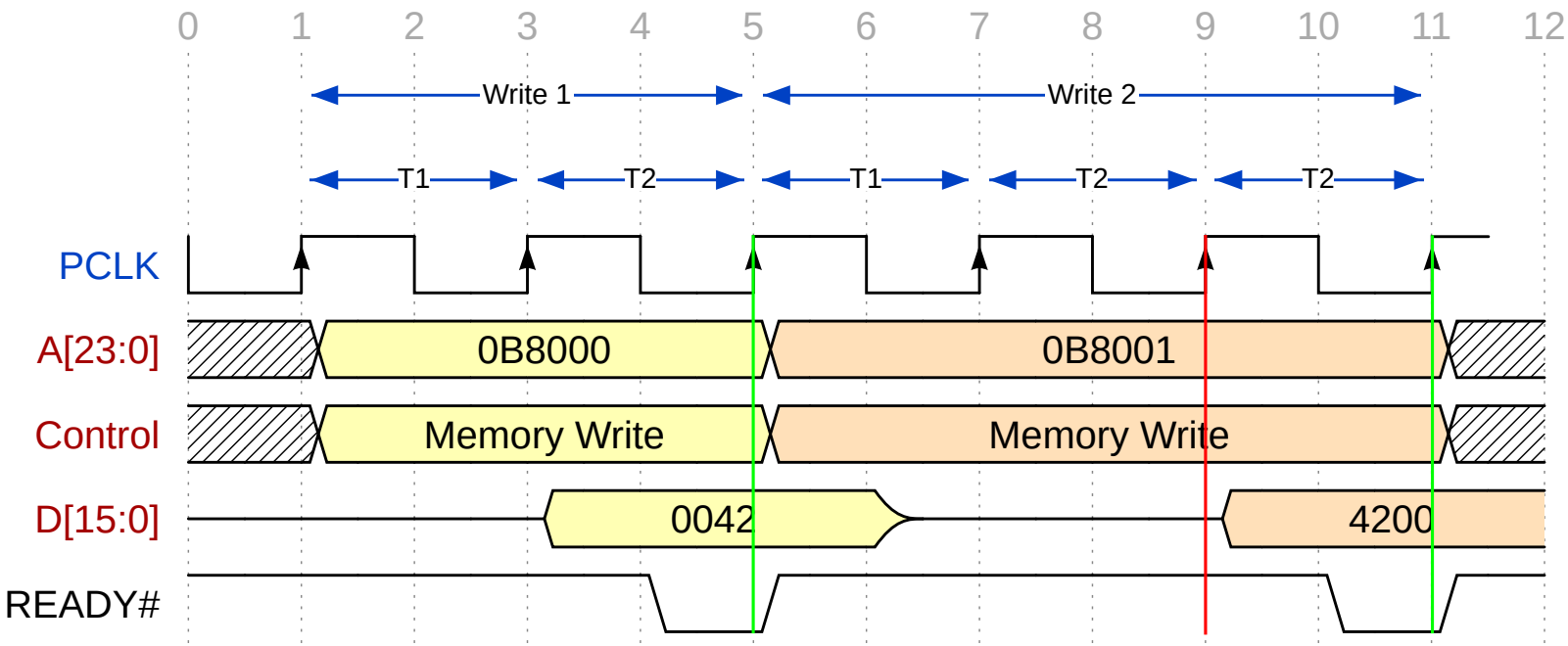
mov dx, 0x3f8; in ax, dx



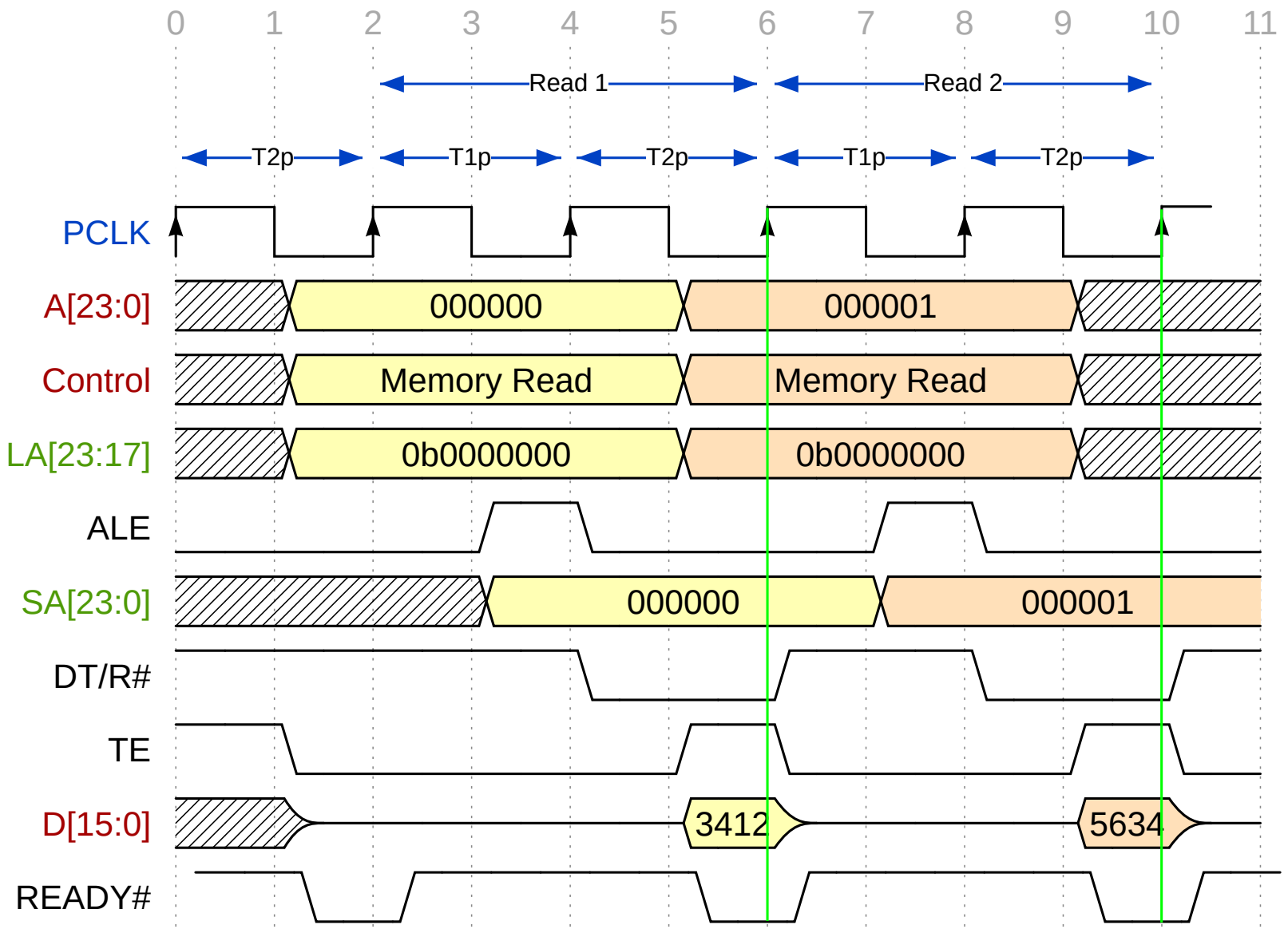
Wait states



Bus cycles and access time



0-Wait State with Pipelining



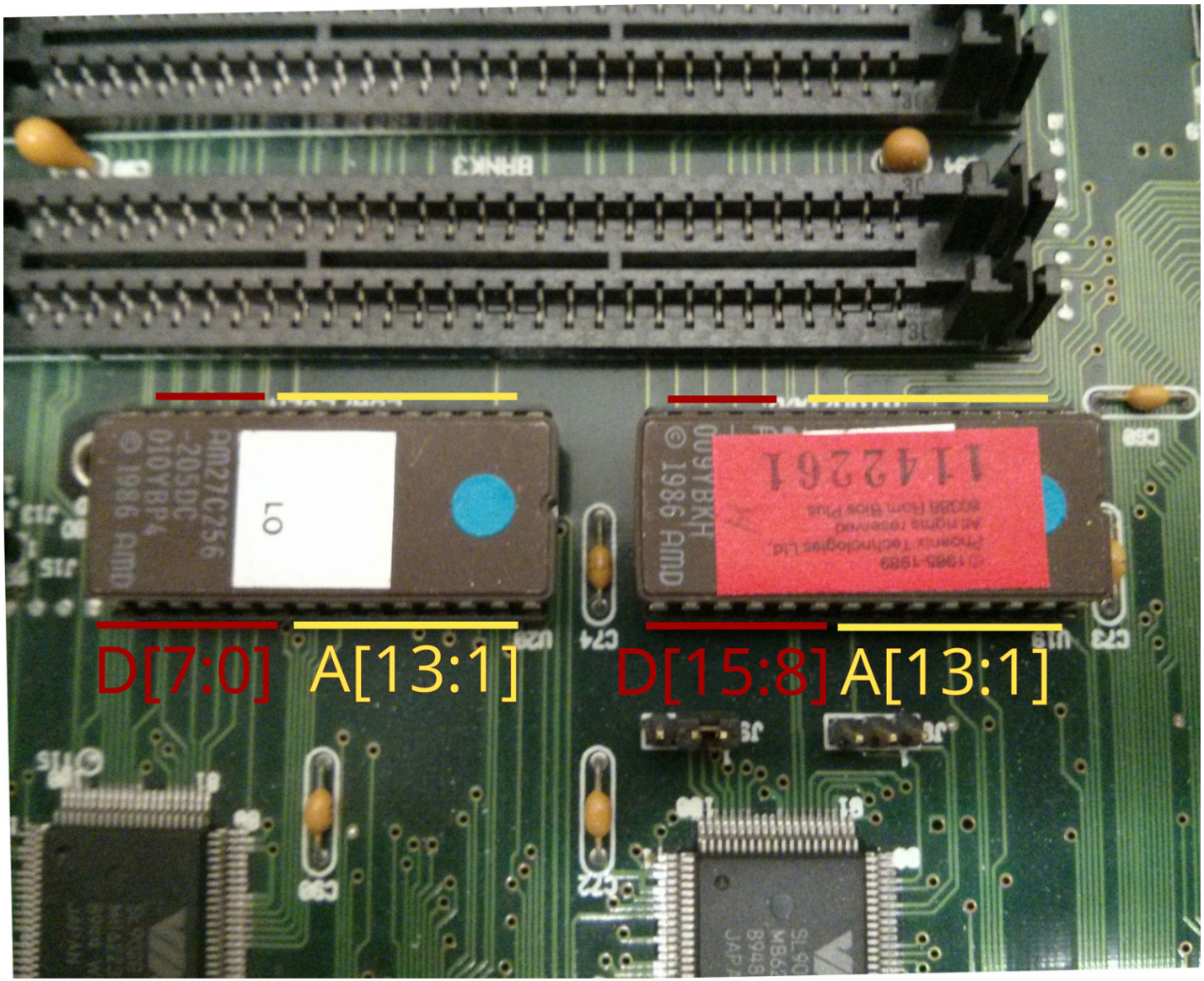
Subsystems

Legacy

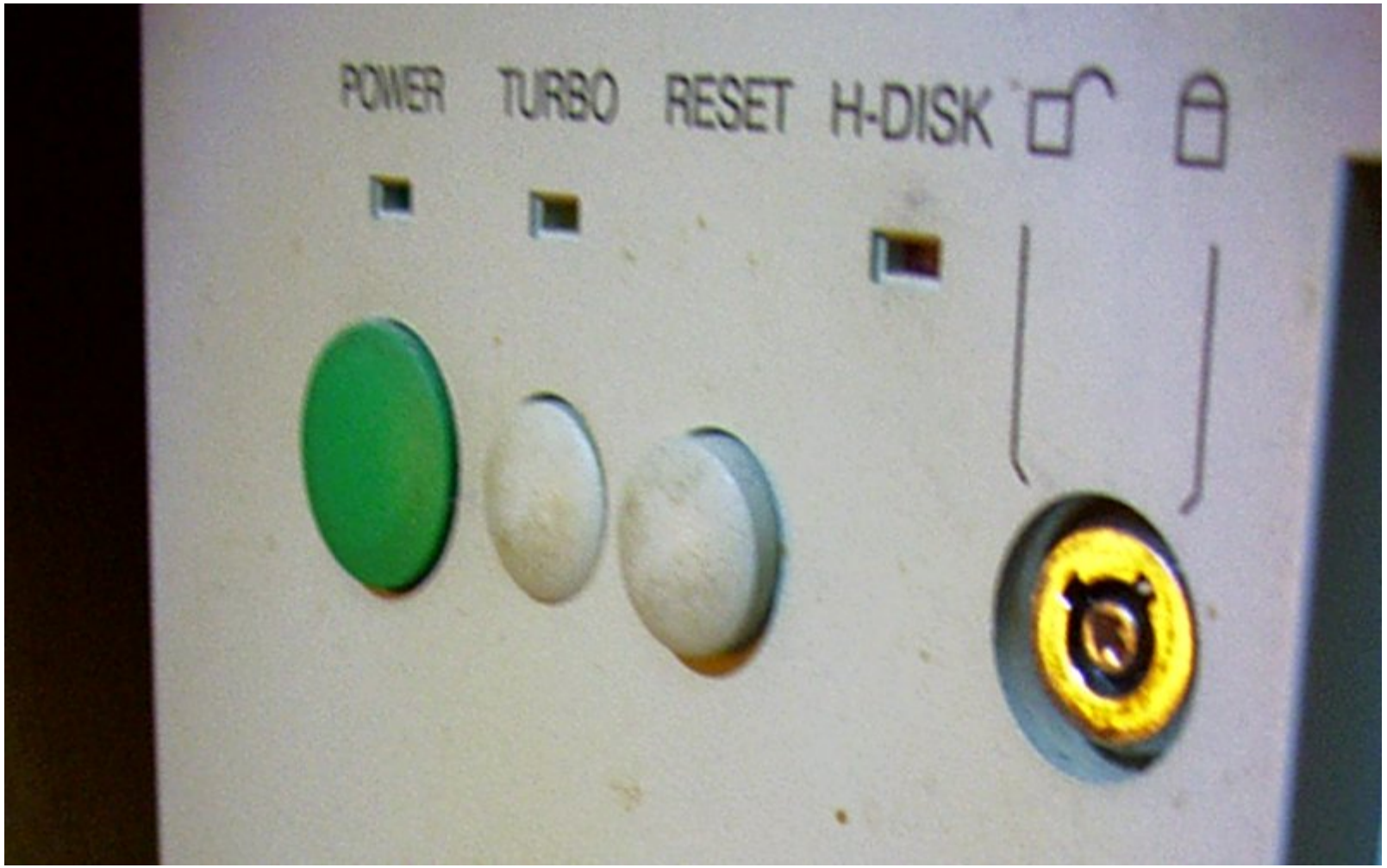
IBM PC, PC/XT and PC/AT systems have defined de-facto standards for:

- Memory mappings
- I/O ports functionality
- Interrupts mapping
- Peripherals behaviors
- BIOS APIs

The BIOS: Basic Input/Output System



The “Turbo” button



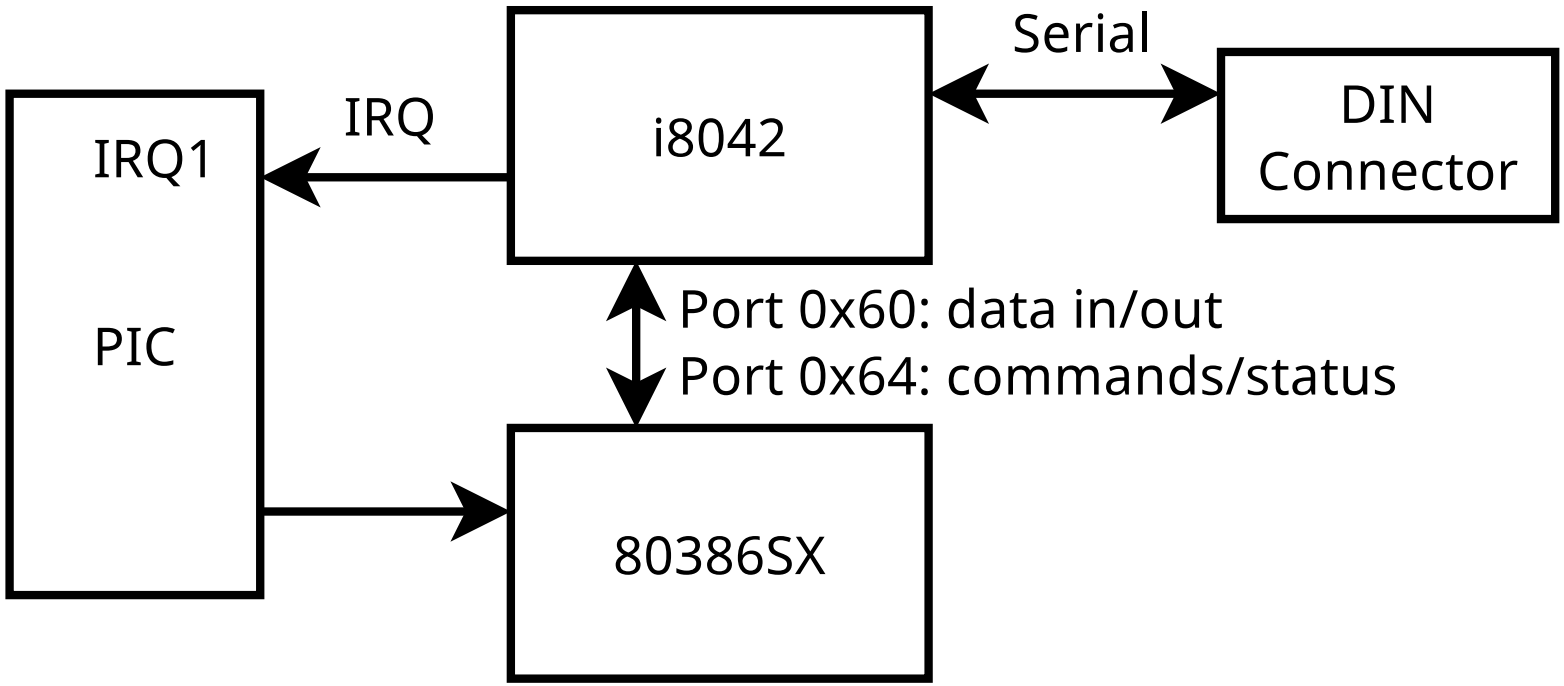
i8042

From Intel:

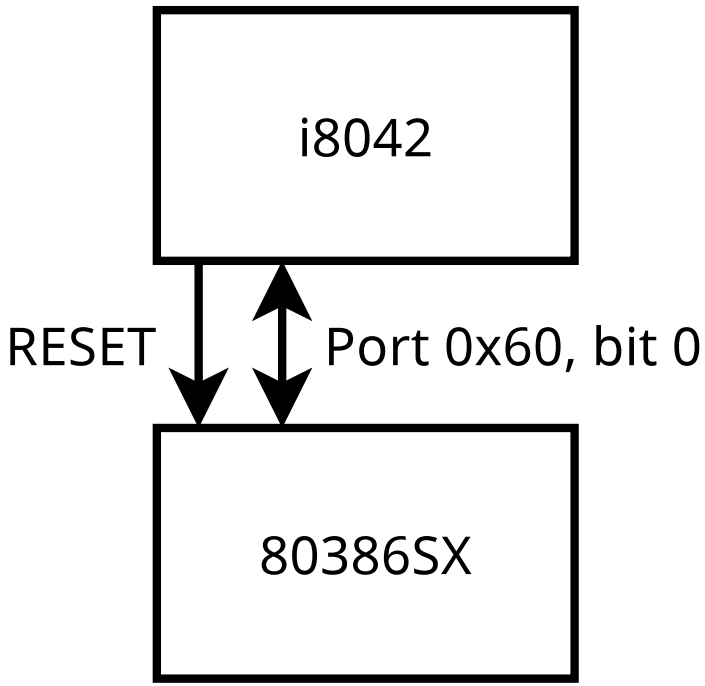
> *The 8042 is a general-purpose Universal Peripheral Interface that allows designers to grow their own customized solutions for peripheral device control.*

In the IBM PC, it's main purpose is Keyboard Control, but it also has other uses...

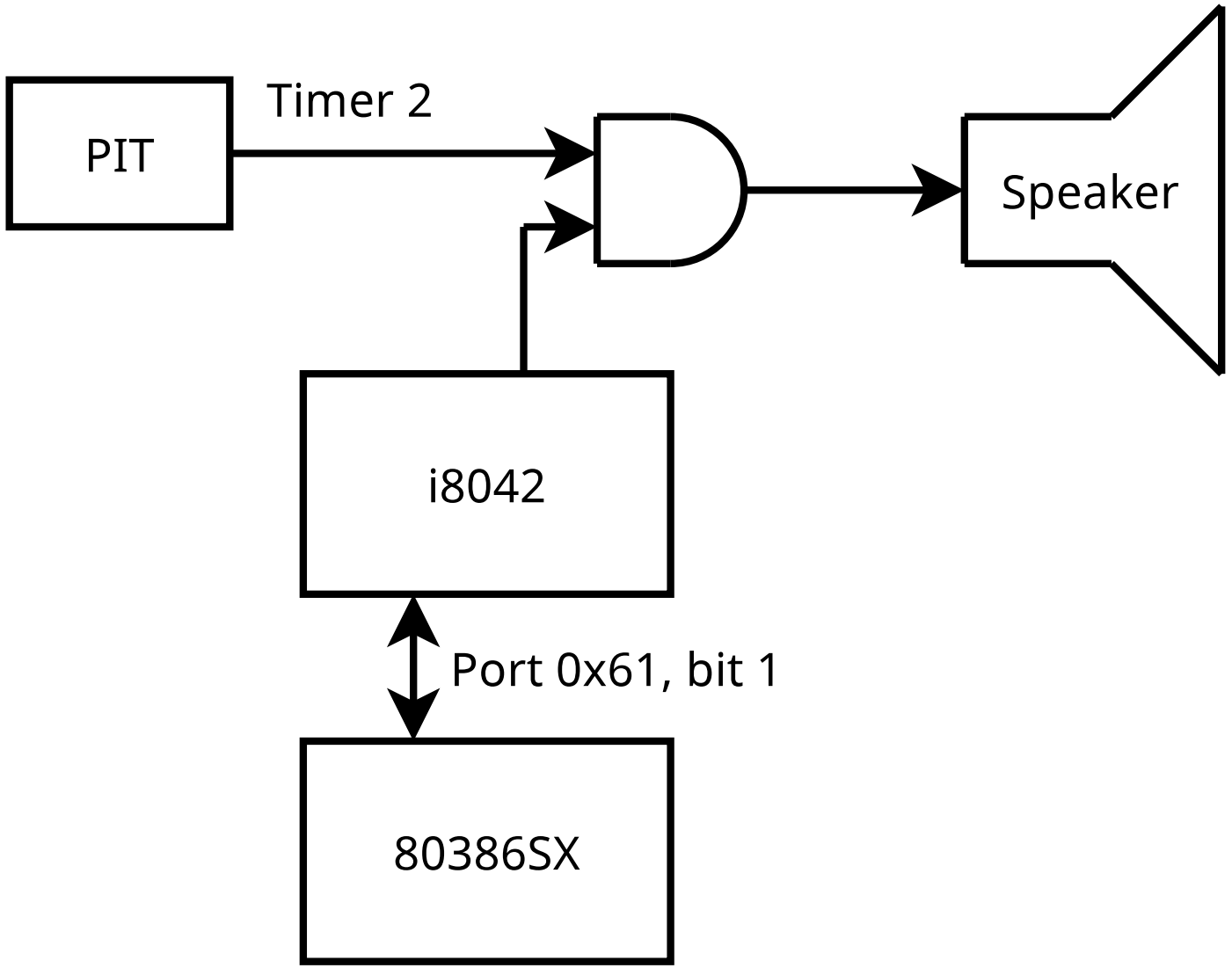
i8042: Keyboard interface



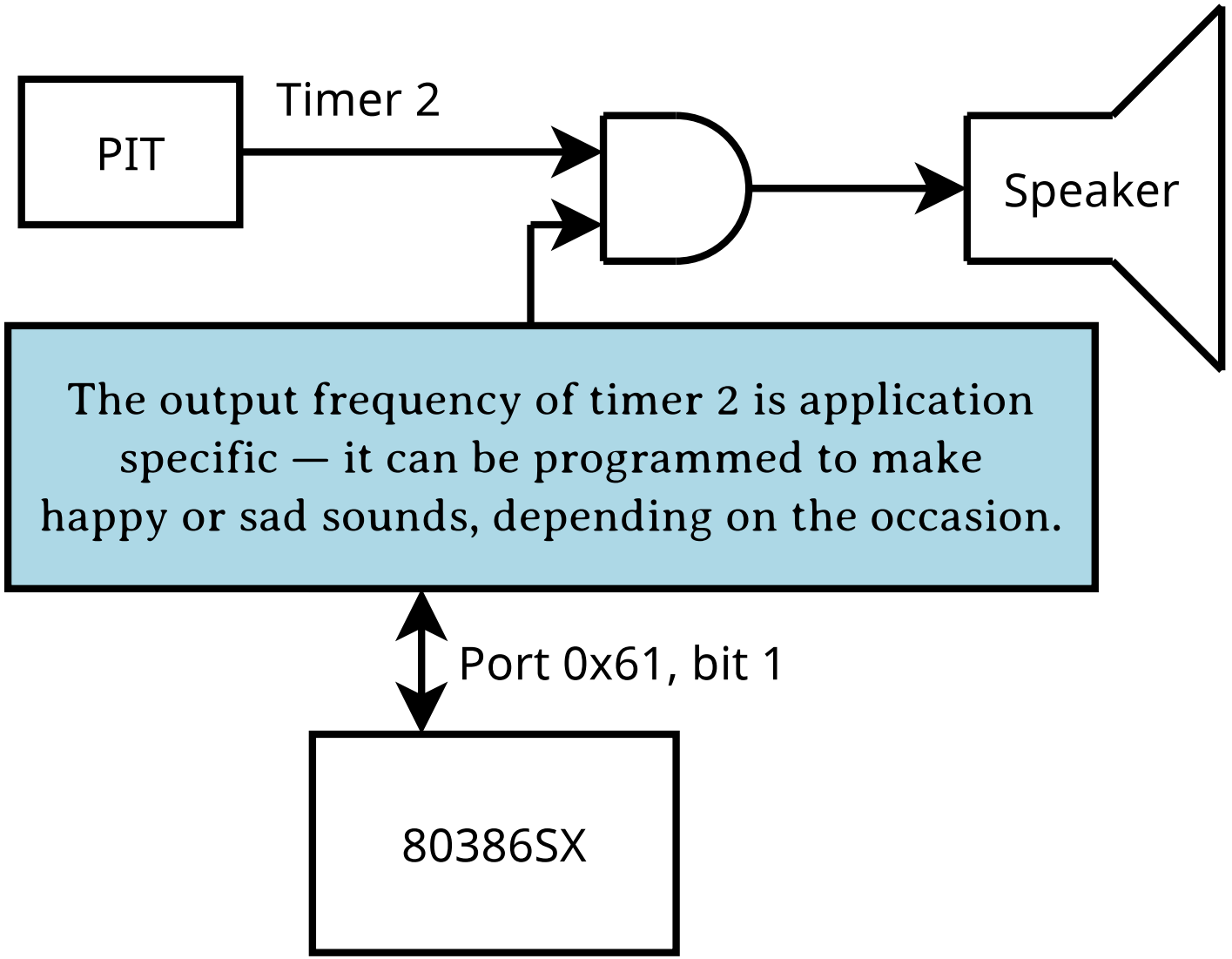
i8042: CPU Reset



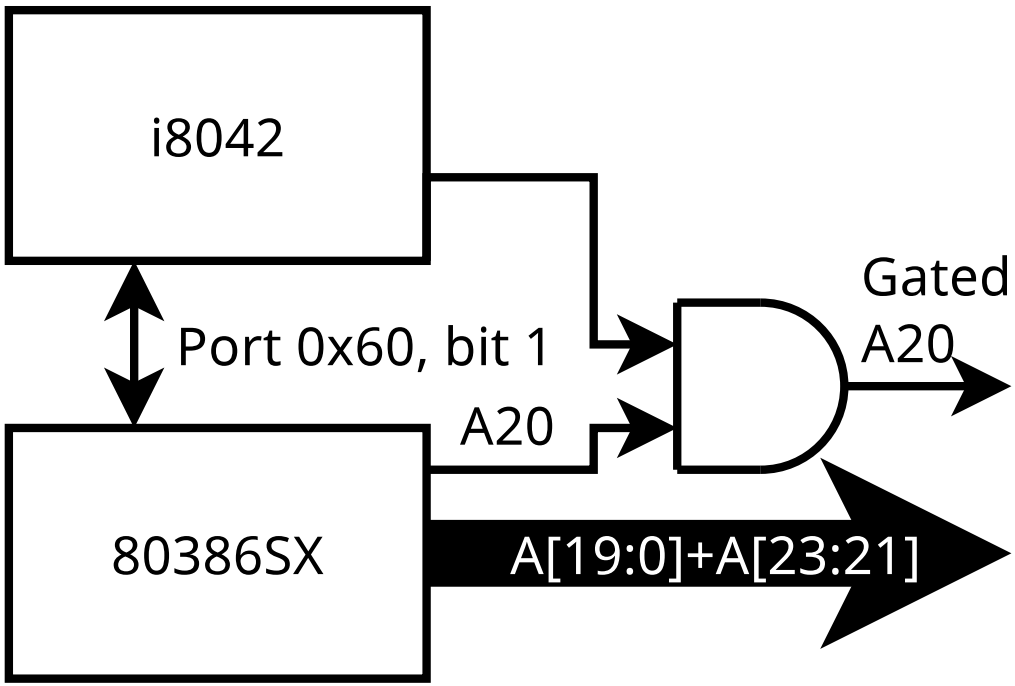
i8042: Speaker control



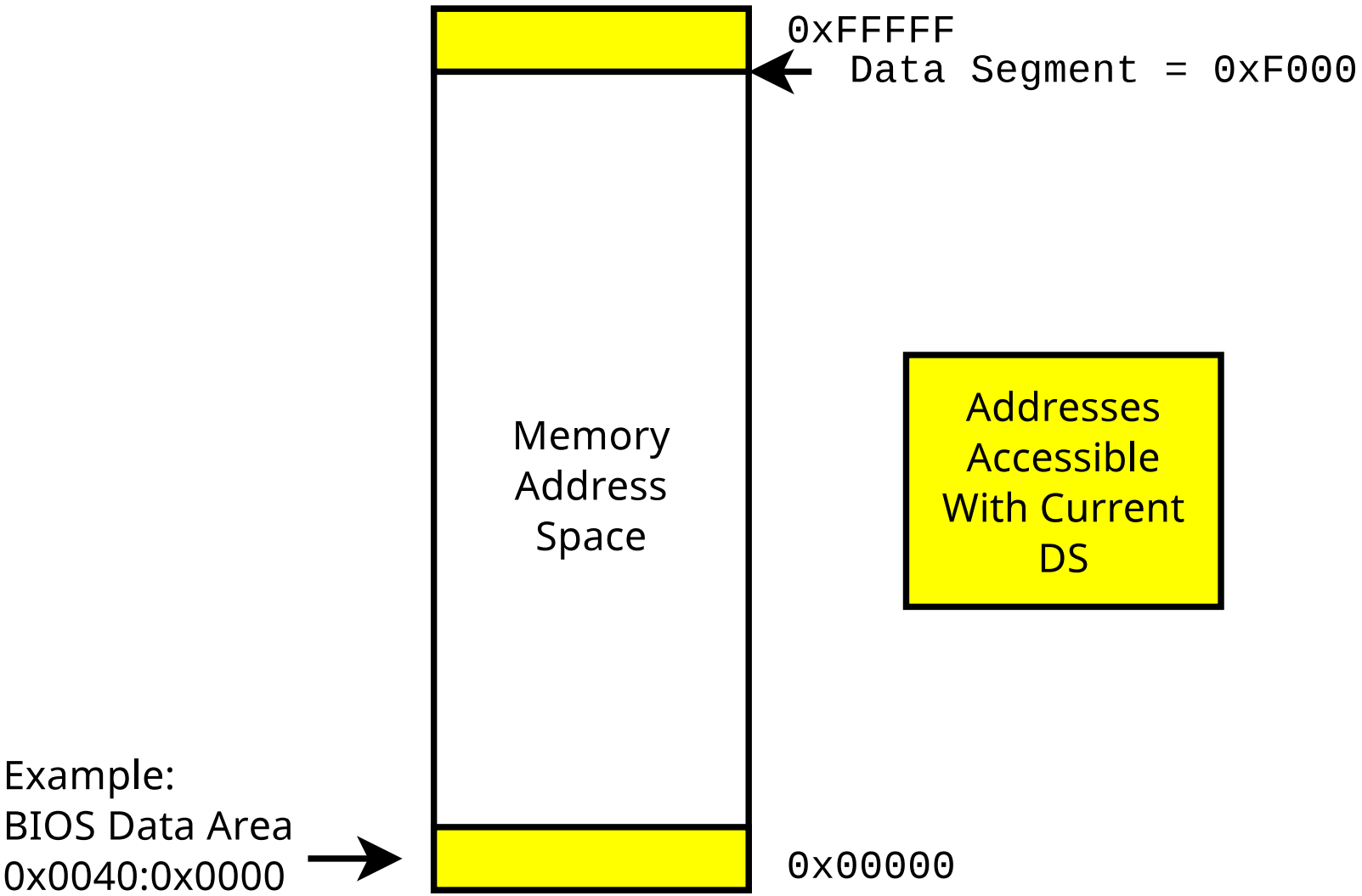
i8042: Speaker control



i8042: A20 Gate



Segment Wraparound

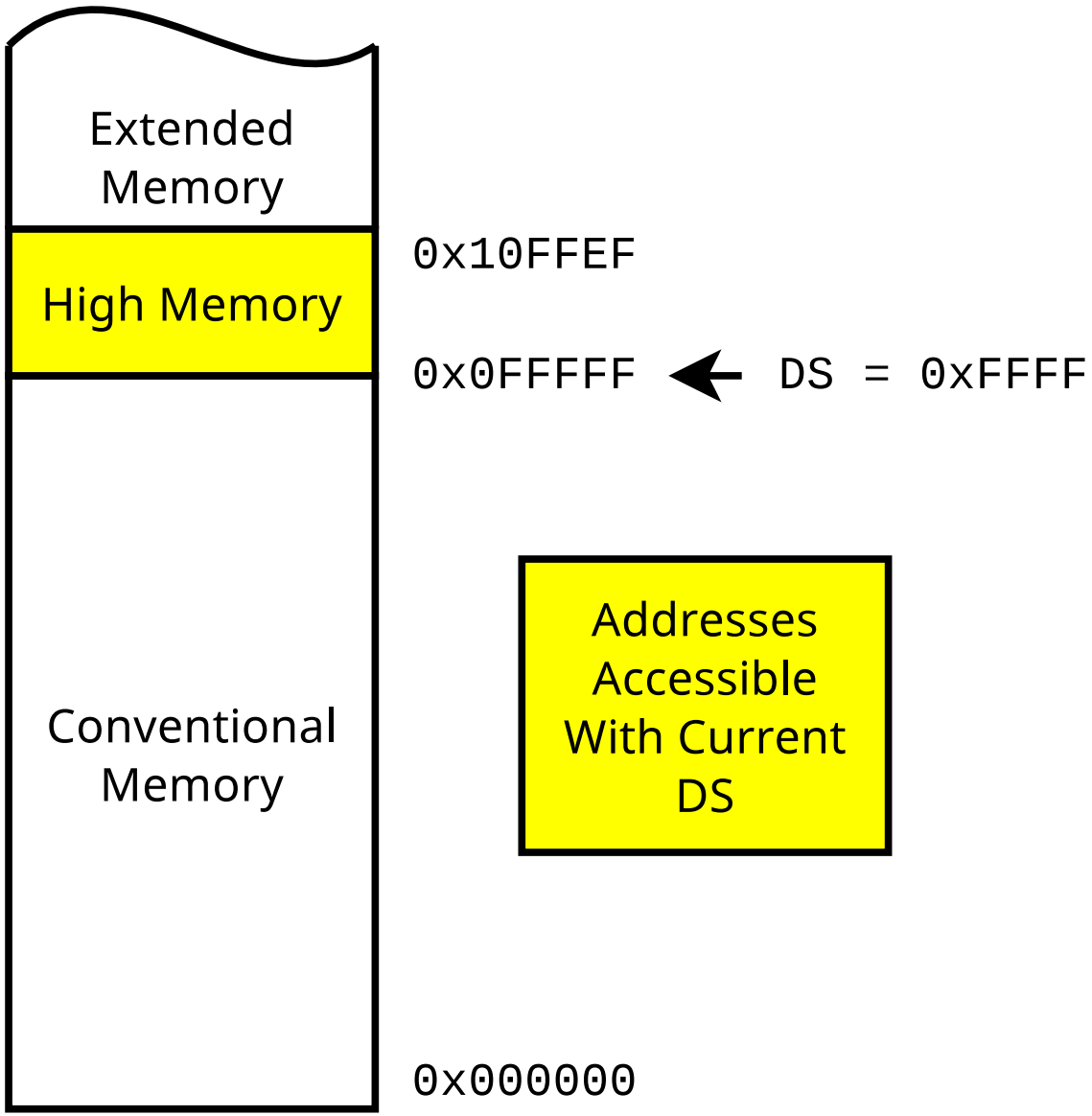


Real Mode Segmentation: A20 Masked

Segmentation trick: overflow to get low addresses from high segment.

```
; Setup Data Segment
mov ax, 0xFFFF    ; ax <- FFFFh
mov ds, ax        ; set data segment to FFFFh
; Read byte from memory address:
mov al, [0x0020] ;   DS << 4  ->     FFFF0h
                 ; + offset   ->   + 00010h
                 ;   -------        -------
                 ; effective addr = 100010h
```

Extended Memory



Real Mode Segmentation: A20 Unmasked

Segmentation trick: overflow to get higher addresses than normal.

```
; Setup Data Segment
mov ax, 0xFFFF    ; ax <- FFFFh
mov ds, ax        ; set data segment to FFFFh
; Read byte from memory address:
mov al, [0xFFFF] ;   DS << 4  ->    FFFF0h
                 ; + offset   ->  + 0FFFFh
                 ;                --------
                 ;                = 10FFEFh
```

Other subsystems

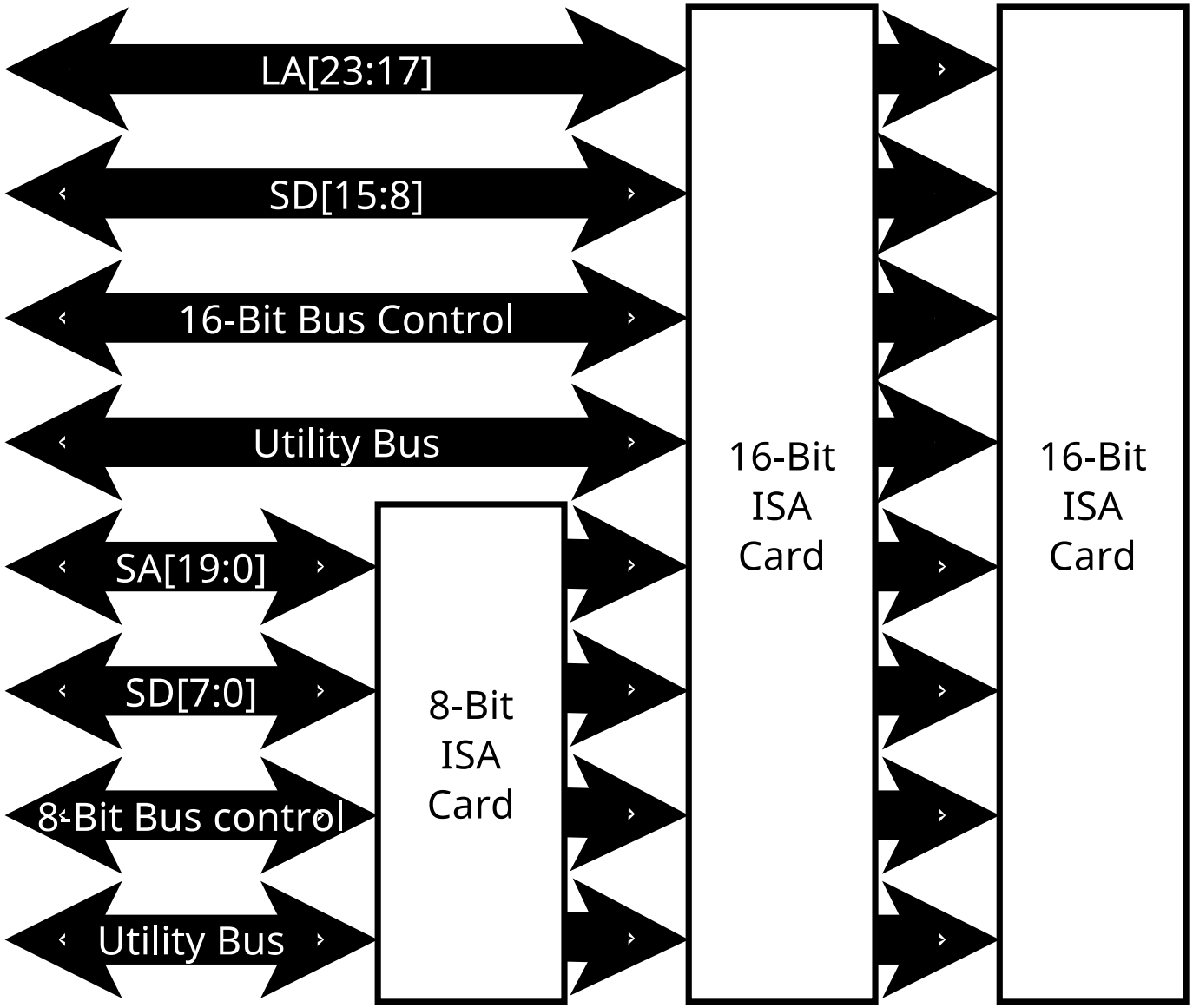
- Intel 8237: DMA Controllers
- Intel 8259: PIC
- Intel 8254: PIT
- RAM controllers

i8042: other features

- System Password Management
- Mouse controller
- RAM
- Display type (monochrome/color)
- ...

Industry Standard Architecture Bus

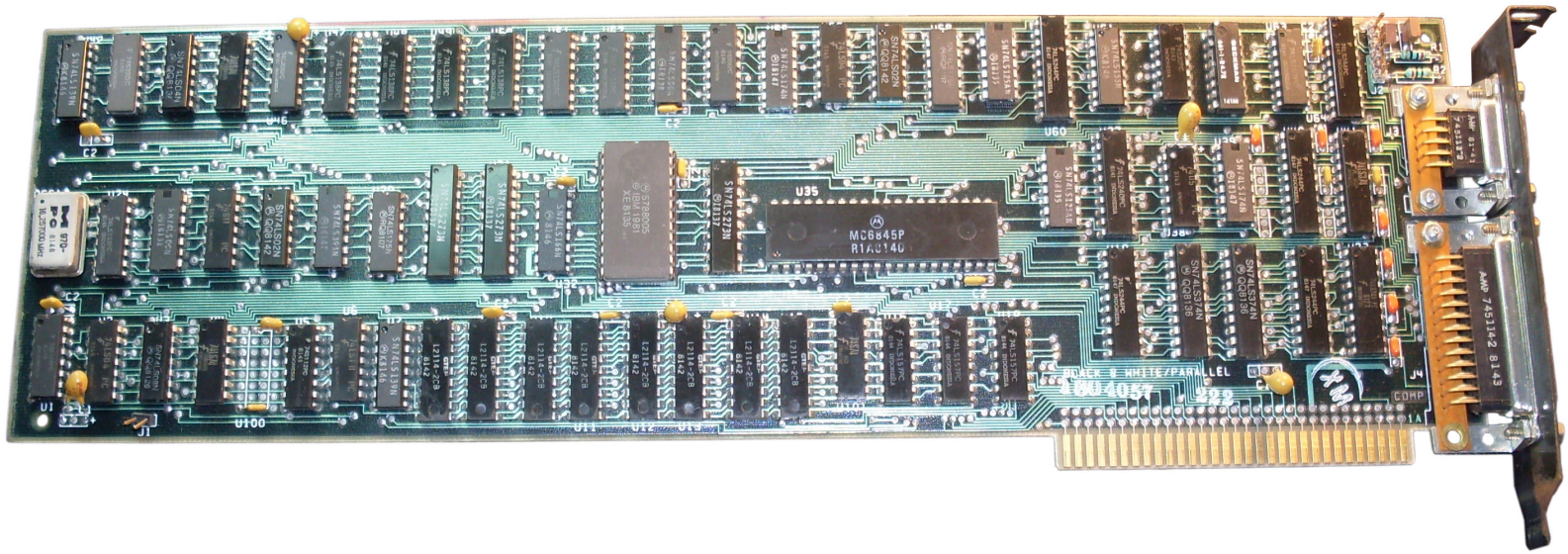
ISA: internal, parallel, clocked at 8MHz



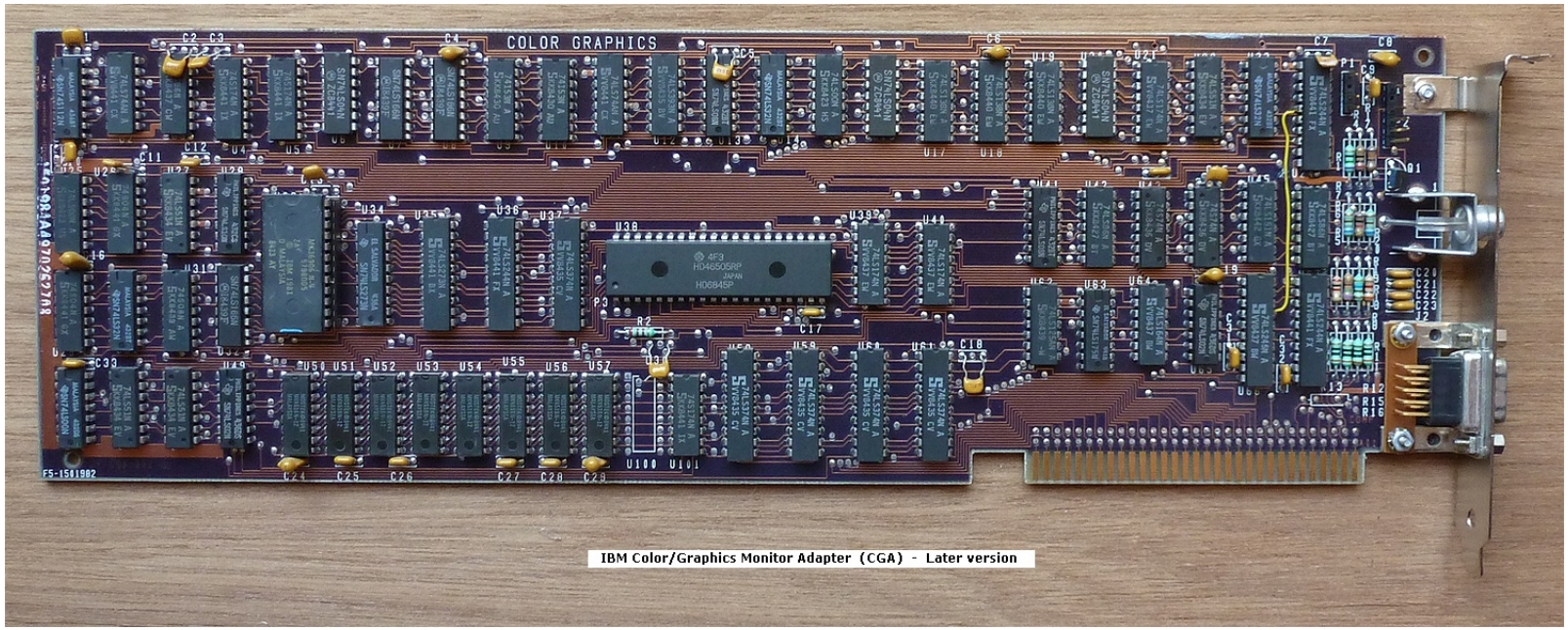
Typical ISA cards

- Additional RAM
- Sound card: Sound blaster
- Graphic adapters: Cirrus Logic, Matrox, 3dfx
- Mass storage: hard drive and floppy controllers
- Simple I/O: parallel and serial ports
- Industrial buses

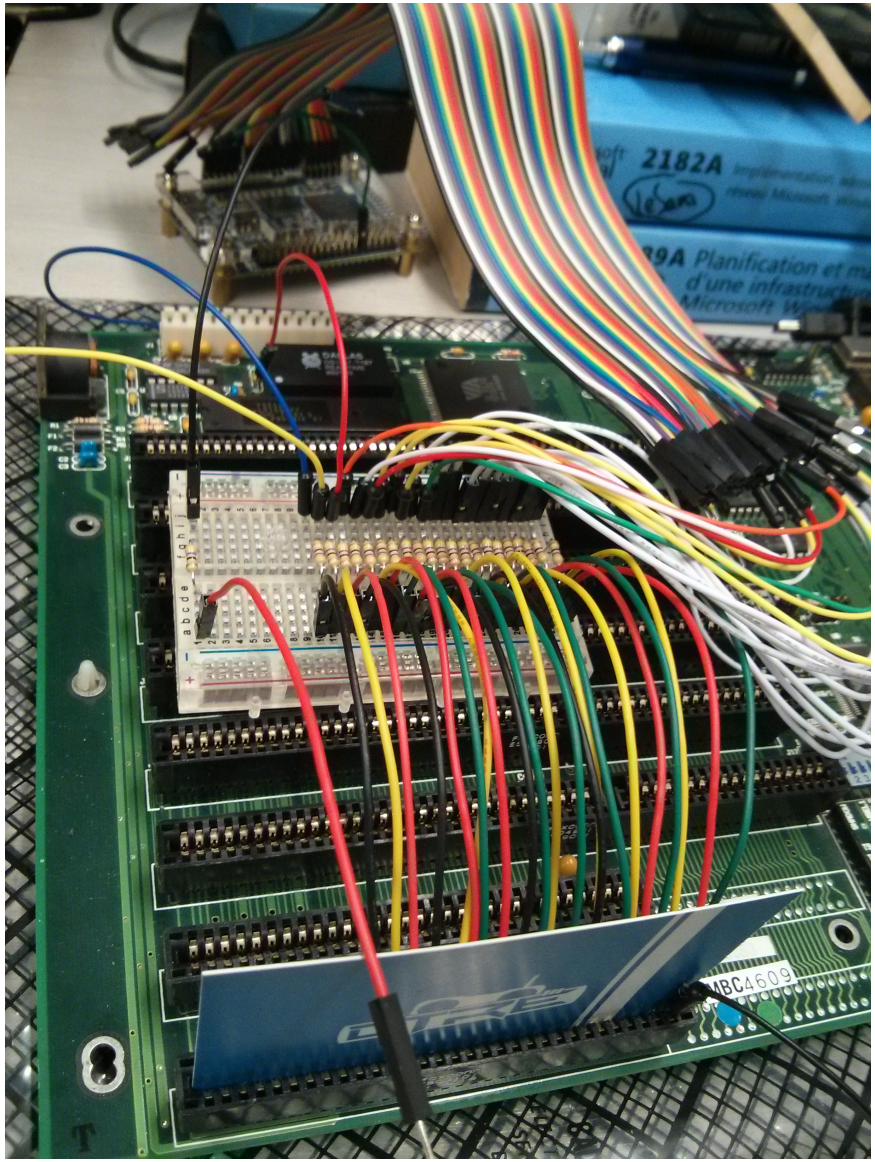
Monochrome Display Adapter



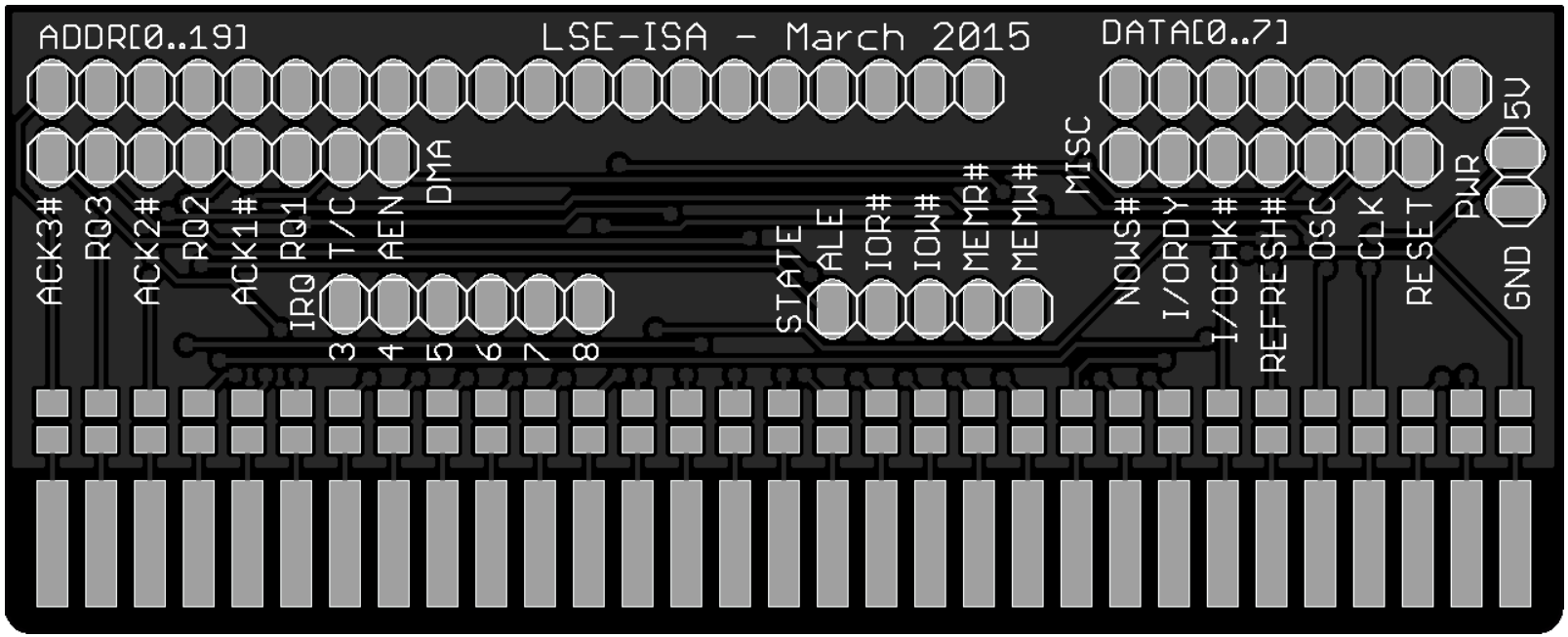
Color Display Adapter



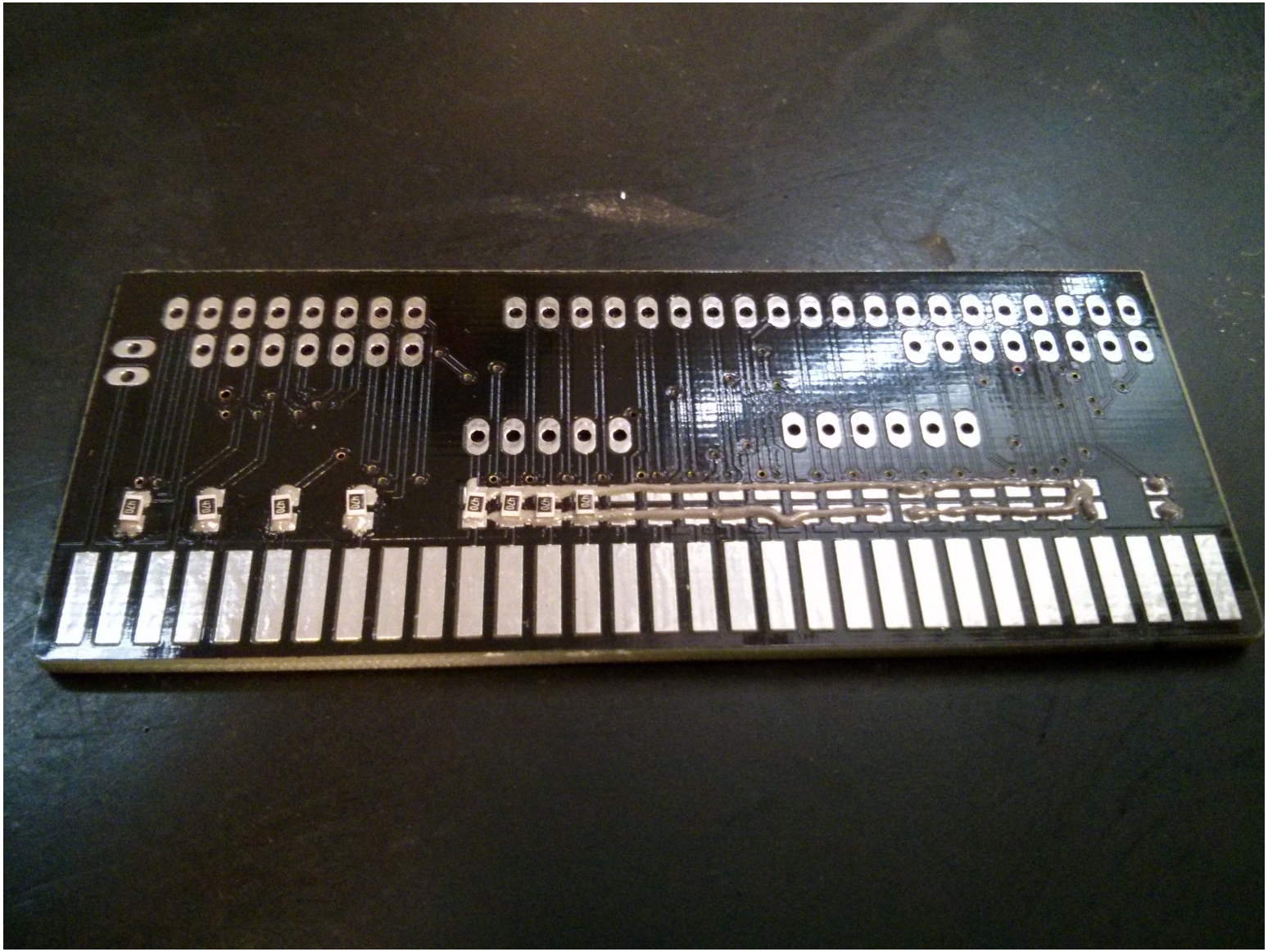
Probing the ISA bus



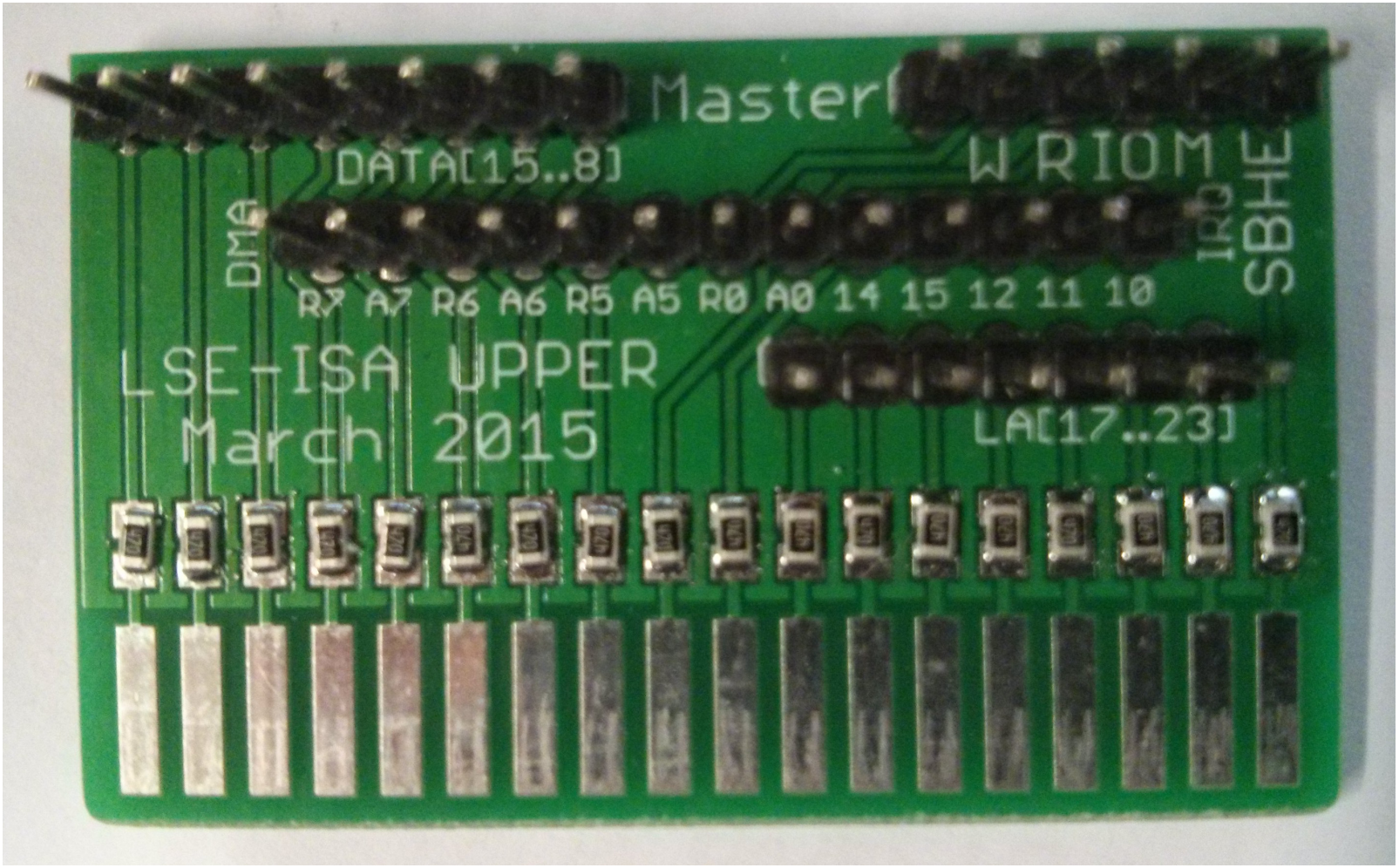
LSE-ISA: 8-Bit ISA breakout



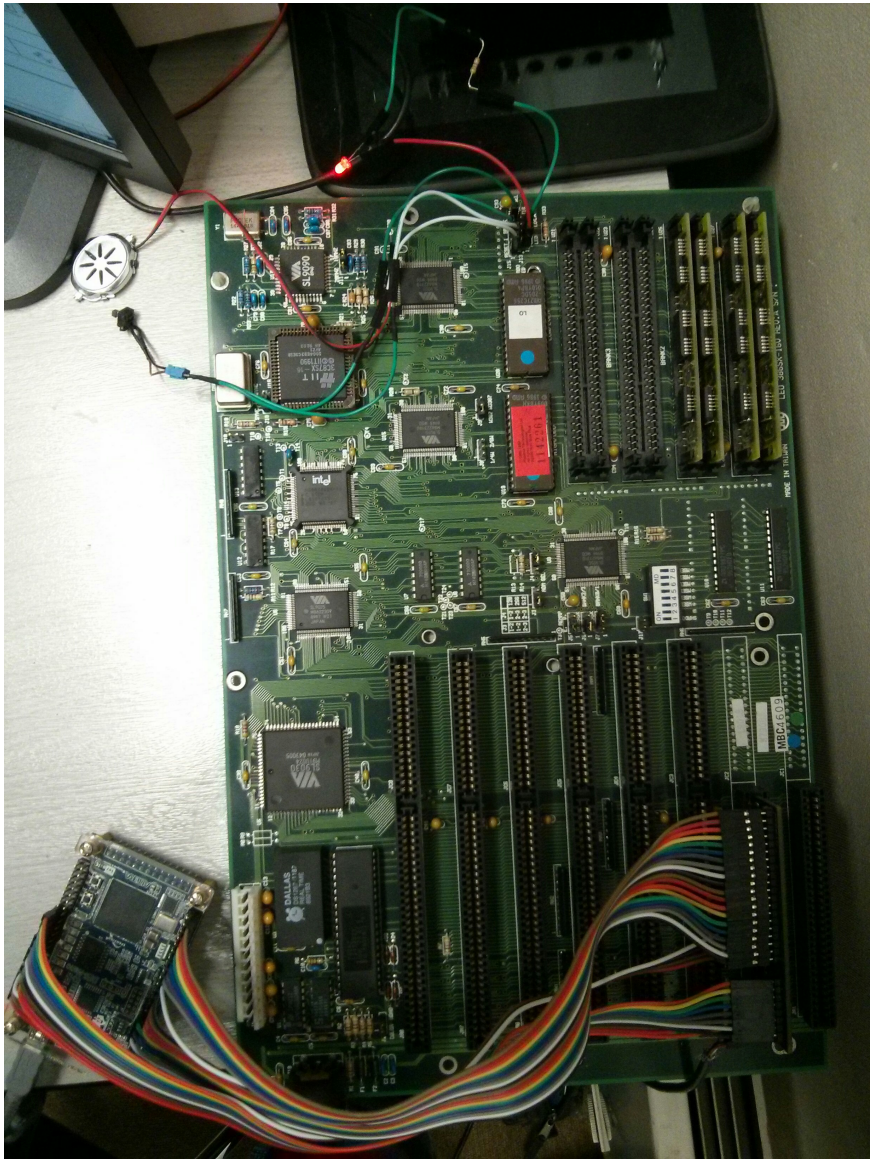
LSE-ISA: soldering



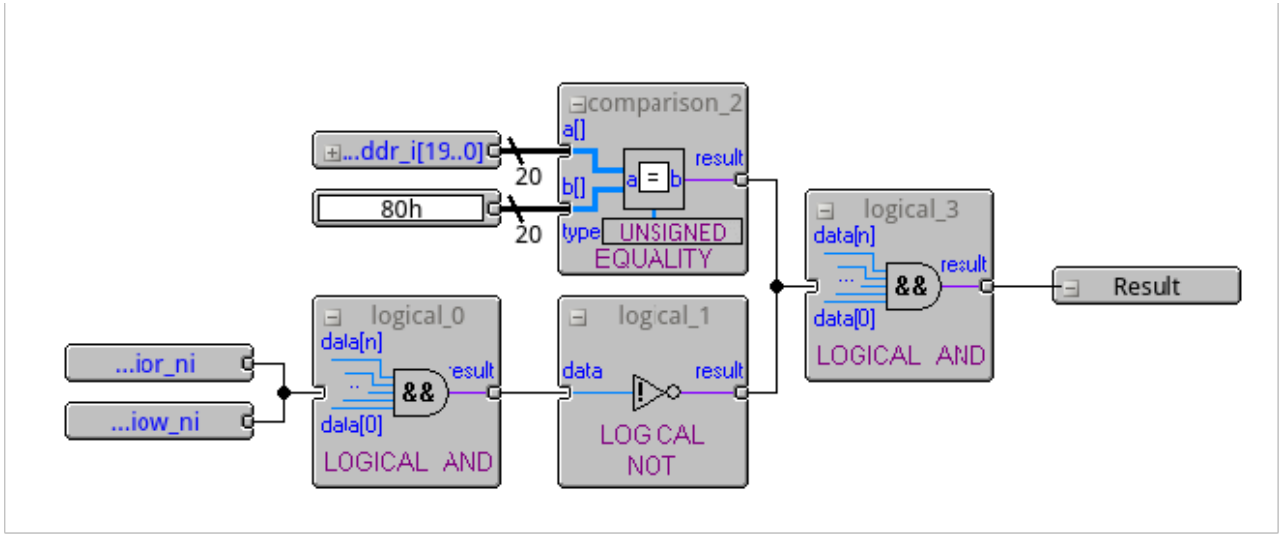
LSE-ISA: 16-Bit ISA breakout

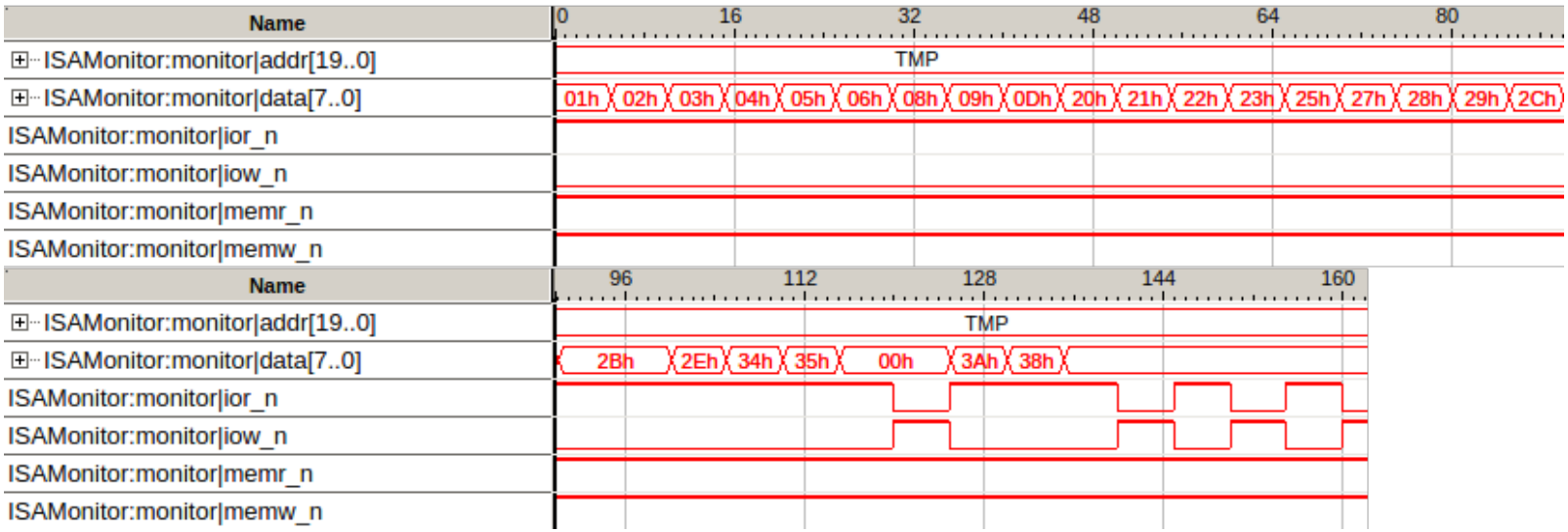


Probing the ISA bus

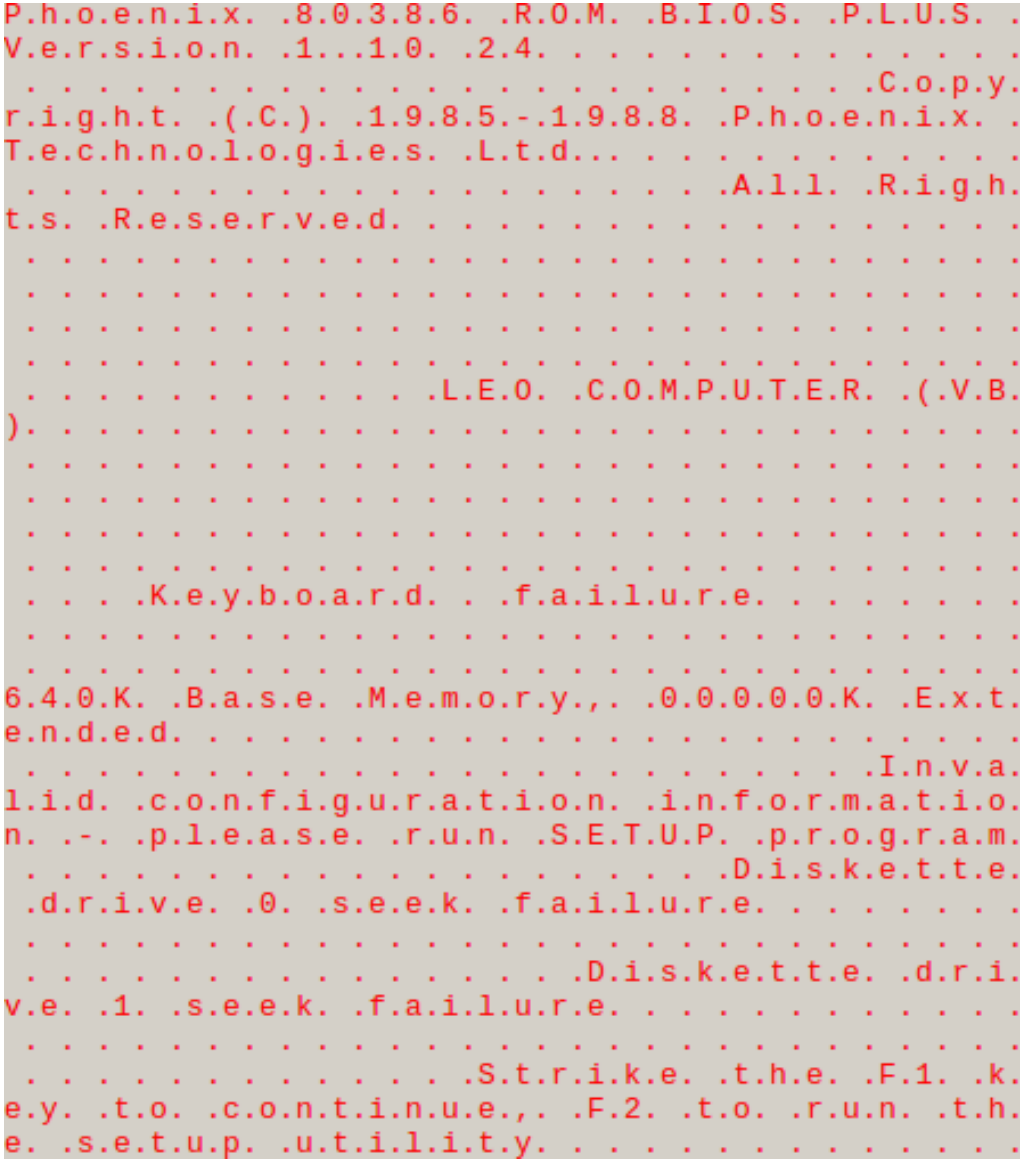


Listening to the bus

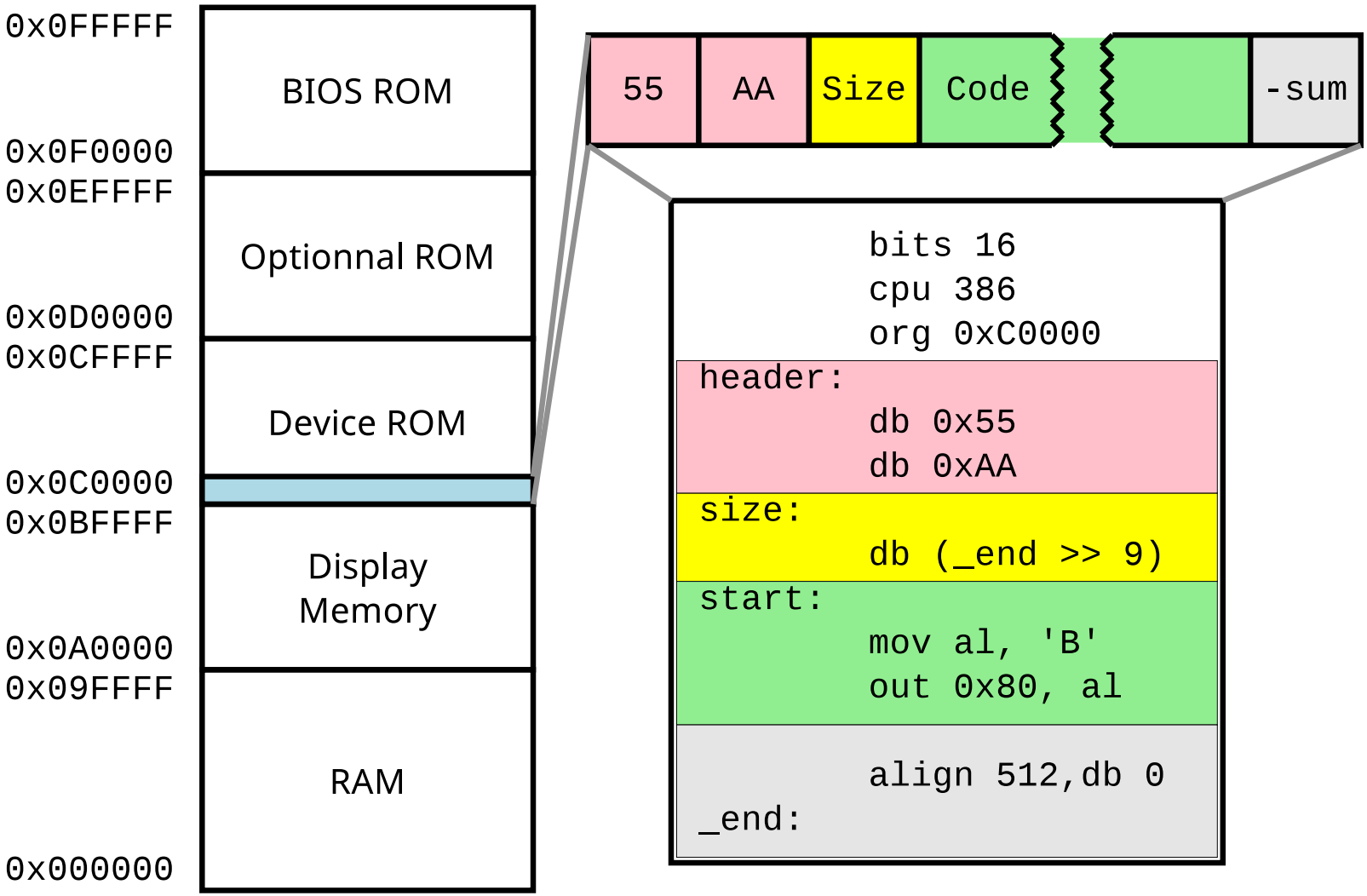




Emulating the MDA Buffer



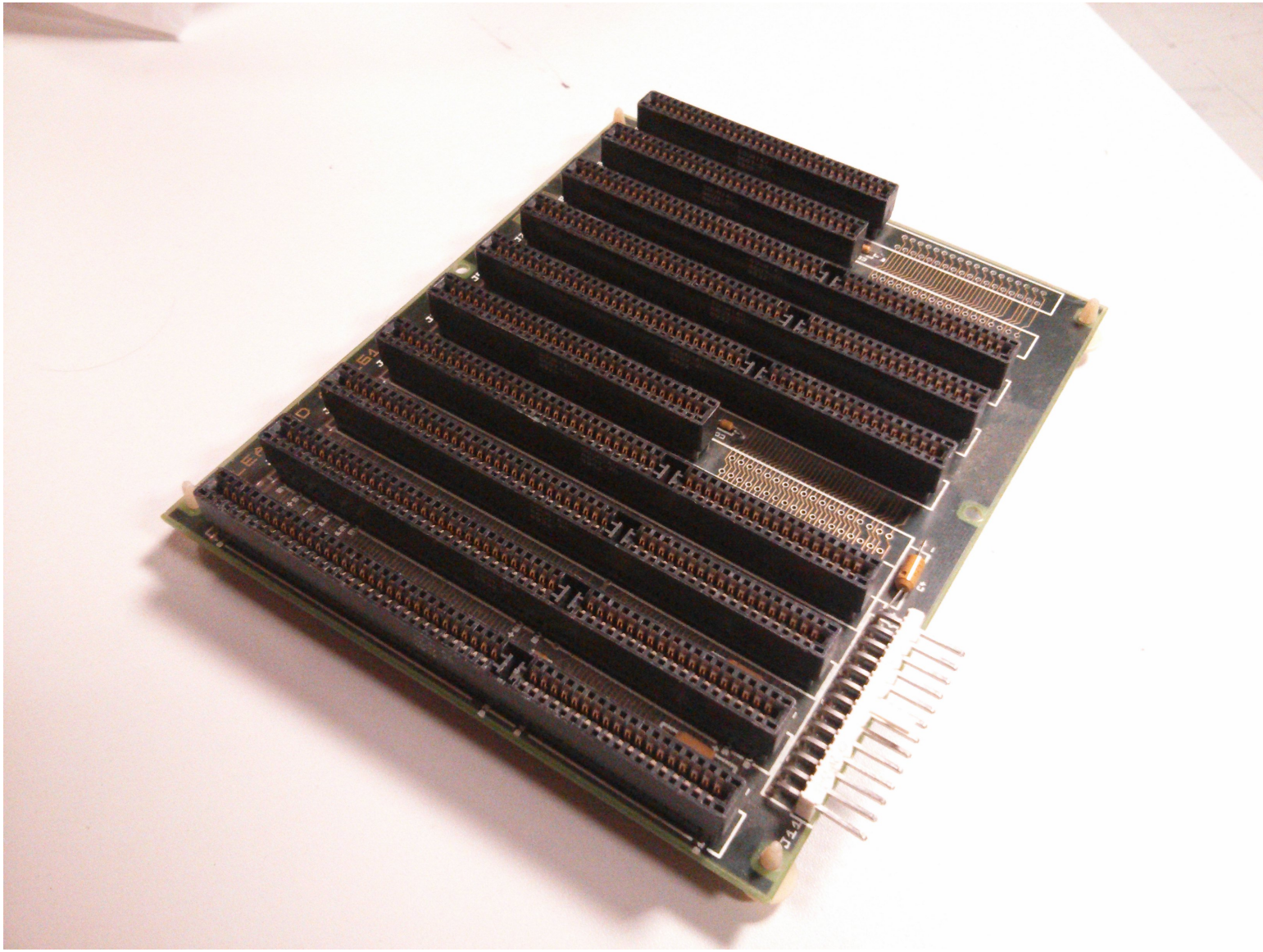
Device ROM


0x0FFFFF
BIOS ROM
0x0F0000
0x0EFFFF
Optionnal ROM
0x0D0000
0x0CFFFF
Device ROM
0x0C0000
0x0BFFFF
Display Memory
0x0A0000
0x09FFFF
RAM
0x000000
55
AA
Size
Code
-sum
bits 16
cpu 386
org 0xC0000
header:
db 0x55
db 0xAA
size:
db (_end >> 9)
start:
mov al, 'B'
out 0x80, al
align 512,db 0
_end:

Dumping the BIOS using the Device ROM

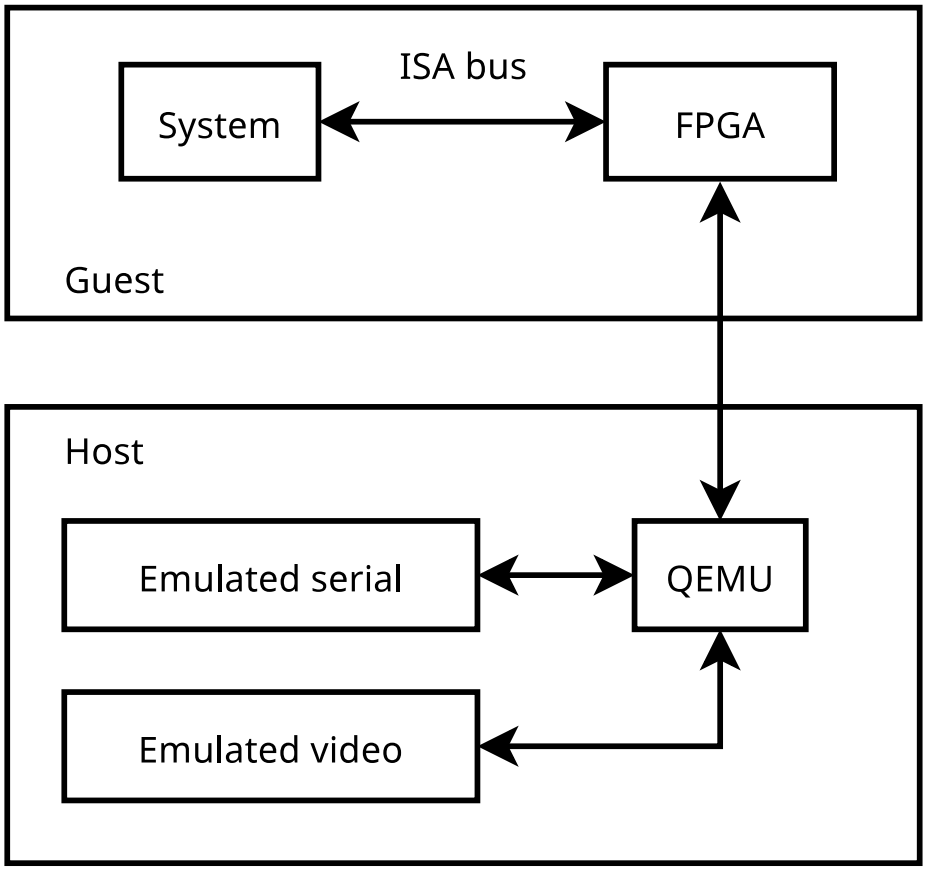

```
; Set source segment to start of BIOS data
mov ax, 0xF000
mov ds, ax
; Set destination segment to scratch memory
mov ax, 0xA000
mov es, ax

; Set counter to number of bytes to be copied
; FFF is the max (4096d == FFFh)
mov cx, 0x0FFF
; Source
mov si, 0xF000
; Destination
xor di, di

rep movsb
```

Going further with the ISA bus: the backplane



Conclusion

Current work: semihosting for the LEO-PC




```
mov al, 'B'
mov dx, 0x3F8
outb dx, al ; CPU outputs
```

The FPGA is chip selected.
It sends a write request to the host.

QEMU receives the request and calls the matching emulator function.

Tools and References

Tools

- Icarus Verilog
- gtkwave
- Altera Quartus
- Wavedrom
- yasm
- sigrok

References

- ISA System Architecture, by Tom Shanley and Don Anderson
- Intel 80386SX reference
- IBM PC AT Technical and Hardware references
- http://ibm-pc.org/ for the PDF of the IBM documentation
- http://www.minuszerodegrees.net/ IBM 51xx reference
- http://www.karbosguide.com/books/pcarchitecture/start.htm PC Architecture

Questions?

Contact

- Rémi Audebert
  - IRC: `halfr@irc.rezosup.org`
  - Mail: `halfr@lse.epita.fr`

I

Appendix

Shadow RAM

Issue: ROMs have a slow access time.

- In order to compensate, the BIOS will copy the content of BIOS and VIDEO ROM to RAM during POST. This is achieved by:
  1. Reading and writing the ROMs content in place,each write being redirected to RAM.
  2. Disable the ROM and select the RAM for read operations.

- This is not the only way to implement "Shadow RAM":
  - Use the 386 MMU to map the virtual addresses of the ROM to a copy in RAM.
  - Use a Position Independent BIOS, copy it somewhere and jump there.

After ISA

- ISA Plug'n'Play
- MCA
- Extended ISA (EISA)
- VESA Local Bus
- PCI
- AGP
- PCI
- PCI-X
- PCI Express

Math coprocessor: 80387SX

387 instructions
executed by the 386
trigger computation offloads to
the 387 using IO Ports

IO Ports
0x8000F8 - 0x8000FF