PURDUE UNIVERSITY DEPARTMENT OF PHYSICS

Physics 53600

Electronics Techniques for Research

Now in PowerPoint!

Spring 2020 Semester

Prof. Matthew Jones

ANNOUNCEMENT

- Obvious changes to the course:
  - No in-person lectures: you'll have to read the lecture notes yourself
  - No more labs: don't worry about it – your grade will be based on work done so far
  - Remaining assignments will try to cover topics that would have been explored in the lab
  - Second mid-term: simplest to cancel it
  - Final exam: I'm not sure what to do about this yet, but I'll figure something out.
- Changes to grading scheme:
  - Old scheme: Assignments (30%) exams (40%) lab (30%)
  - New scheme: Assignments (50%) exams (25%) lab (25%)

ANNOUNCEMENT

- Because there won't be any in-person lectures, you will have to read the lecture notes yourself.
- To demonstrate that you have read them, you will be required to answer *one or two simple questions* before the next lecture is posted.
- The question will be somewhere (like maybe at the end?) and you just have to e-mail me the answer

  mjones@physics.purdue.edu
- To make this easy, please make your subject look like this:

  "PHYS53600 Lecture xx questions Your Name"
- These will be part of your assignment grade, maybe contributing 10% of your total grade.

LECTURE 19 QUESTION #1

What are four ways that can be used to increase channel capacity?

LECTURE 19 QUESTION #2

Although the SPI interface might not necessarily be very fast, what are some of its advantages?

Digital Data Transmission

- The speed of digital circuits is often limited by the error-free rate at which data can be transferred
- This includes transmission on all length scales:
  - Communication with Voyager 2 (10^{13} m)
  - Trans-oceanic cables (10^{6} m)
  - Commercial network infrastructure (10^{3} m)
  - Peripheral interfaces (1 m)
  - Computer backplanes (10^{-1} m)
  - Printed circuit board interconnects (10^{-2} m)
  - Within an integrated circuit (10^{-3} m)

Digital Data Transmission

- Fundamental limit is described by Shannon's Channel Capacity Theorem:

$$c = b \log_2(1 + s)$$

- Where
  - c is the error-free single-channel capacity (eg. in bits per second)
  - b is the bandwidth of the channel
  - s is the signal-to-noise ratio

Digital Data Transmission

- There are several obvious ways to increase the rate at which data can be transferred:
  - Increase the bandwidth
    - Low bandwidth example: smoke signals
    - High bandwidth example: shielded coaxial cable
  - Increase the signal-to-noise ratio
    - Either increase the amplitude of the signal or reduce the amplitude of the noise
  - Detect and correct errors in data transmission
  - Increase the number of channels

Examples

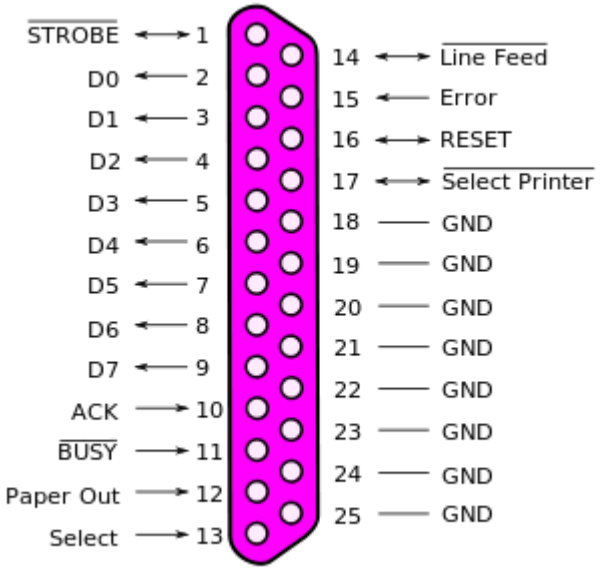
- There are many examples, but many of them are now obsolete because of their fundamental channel capacity limitations
- They still serve as useful examples because they are easy to understand and analyze

Parallel Printer Interface

- The parallel printer interface was developed in the 1970's and was only superseded by other interfaces (like USB) in the 1990's.
- It was more-or-less standardized which allowed many manufacturers to design compatible equipment.
- The standard specified both the electrical and physical interfaces.

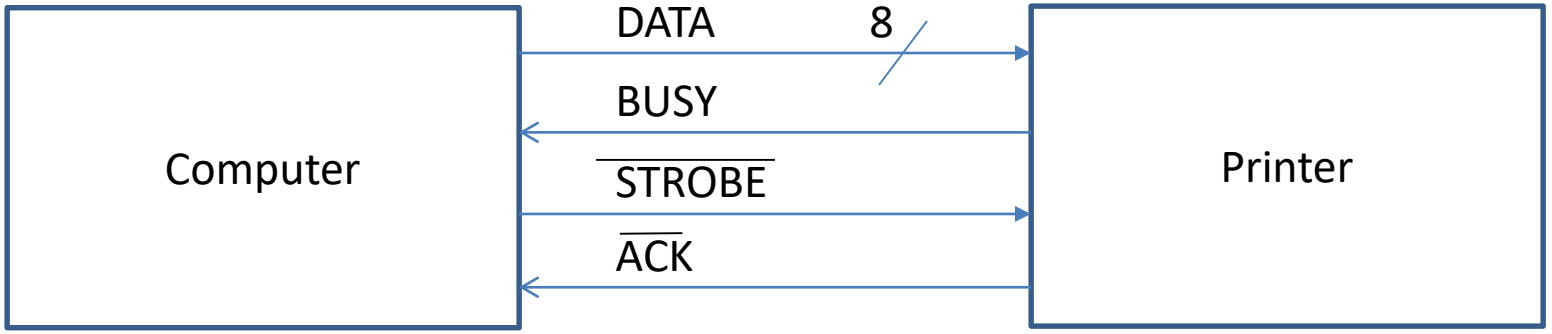
Parallel Printer Interface

- Physical interface:
  - 25-pin connector (DB-25)
  - 25-conductor cable
- Electrical interface:
  - 8 parallel data signals
  - Several control signals
  - Lots of ground signals
  - Also specifies voltages (5 volt TTL logic levels)

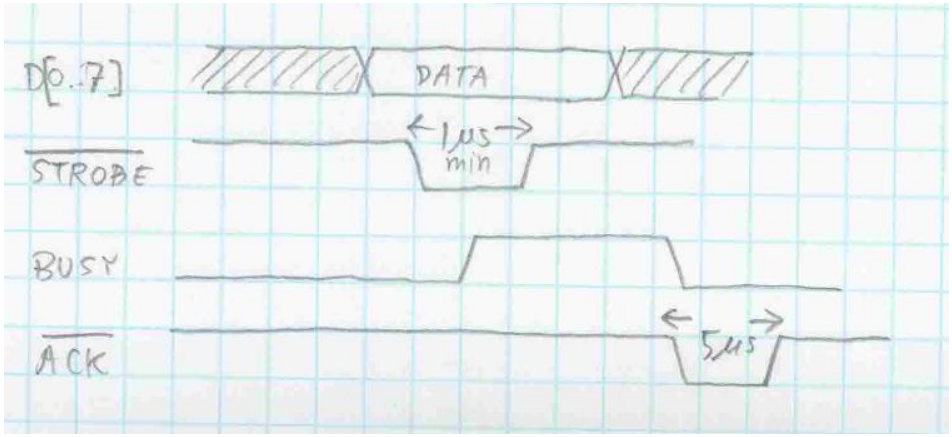


Parallel Printer Interface

- Most communications could be achieved using only three control signals:



- Timing diagram:



Parallel Printer Interface

- Maximum data transfer rate:
  - With this specification, the rate cannot be greater than about 150 kBytes per second.
  - Alternative specifications allowed rates of up to about 500 kBytes per second.
- Disadvantages:
  - Expensive cables (25 conductors)
  - 5 volt single-ended signals (generate RF noise)
  - Slow hand-shaking mechanism
- Fundamental data synchronization mechanism:
  - $\overline{\text{STROBE}}$ indicates when it is safe to sample DATA

Memory Interface

- Just because the parallel interface became obsolete for peripherals (like printers) doesn't mean that the concepts are not still widely used!
- In many cases this is still a very practical way to transfer data over short distances
  - Between integrated circuits on a printed circuit board
  - Between peripheral cards on a bus (like the older-generation PCI bus used in many PC's)

Memory Interface

- For example, consider the CY7C1020CV33 memory chip (32k x 16 bit static RAM)…
- Signal interface:
  - Input address: $A_{0..14}$
  - Data input/output: $D_{0..15}$
  - Chip enable input: $\overline{CE}$
  - Write enable input: $\overline{WE}$
  - Output enable: $\overline{OE}$
- Similarity with the parallel printer interface:
  - One signal (WE or OE) is used to indicate when it is safe to sample the address/data signals

Memory Interface

Features

- **Pin- and function-compatible with CY7C1020V33**
- **High speed**
  - **—t_{AA} = 10, 12, 15 ns**
- **CMOS for optimum speed/power**
- **Low active power**
  - **—360 mW (max.)**
- **Automatic power-down when deselected**
- **Independent control of upper and lower bits**
- **Available in 44-pin TSOP II**

Functional Description

The CY7C1020CV33 is a high-performance CMOS static RAM organized as 32,768 words by 16 bits. This device has an automatic power-down feature that significantly reduces power consumption when deselected.

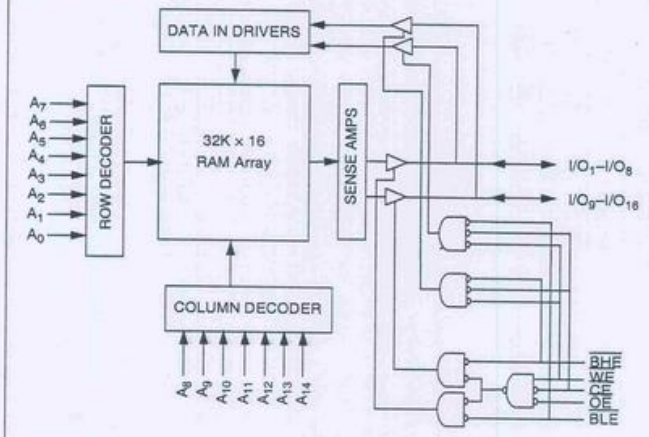
Writing to the device is accomplished by taking Chip Enable ($\overline{CE}$) and Write Enable ($\overline{WE}$) inputs LOW. If Byte Low Enable ($\overline{BLE}$) is LOW, then data from I/O pins (I/O_1 through I/O_8), is written into the location specified on the address pins (A_0 through A_{14}). If Byte High Enable ($\overline{BHE}$) is LOW, then data from I/O pins (I/O_9 through I/O_{16}) is written into the location specified on the address pins (A_0 through A_{14}).

Reading from the device is accomplished by taking Chip Enable ($\overline{CE}$) and Output Enable ($\overline{OE}$) LOW while forcing the Write Enable ($\overline{WE}$) HIGH. If Byte Low Enable ($\overline{BLE}$) is LOW, then data from the memory location specified by the address pins will appear on I/O_1 to I/O_8. If Byte High Enable ($\overline{BHE}$) is LOW, then data from memory will appear on I/O_9 to I/O_{16}. See the truth table at the back of this data sheet for a complete description of read and write modes.

The input/output pins (I/O_1 through I/O_{16}) are placed in a high-impedance state when the device is deselected ($\overline{CE}$ HIGH), the outputs are disabled ($\overline{OE}$ HIGH), the $\overline{BHE}$ and $\overline{BLE}$ are disabled ($\overline{BHE}$, $\overline{BLE}$ HIGH), or during a write operation ($\overline{CE}$ LOW, and $\overline{WE}$ LOW).

The CY7C1020CV33 is available in standard 44-pin TSOP Type II packages.

Logic Block Diagram

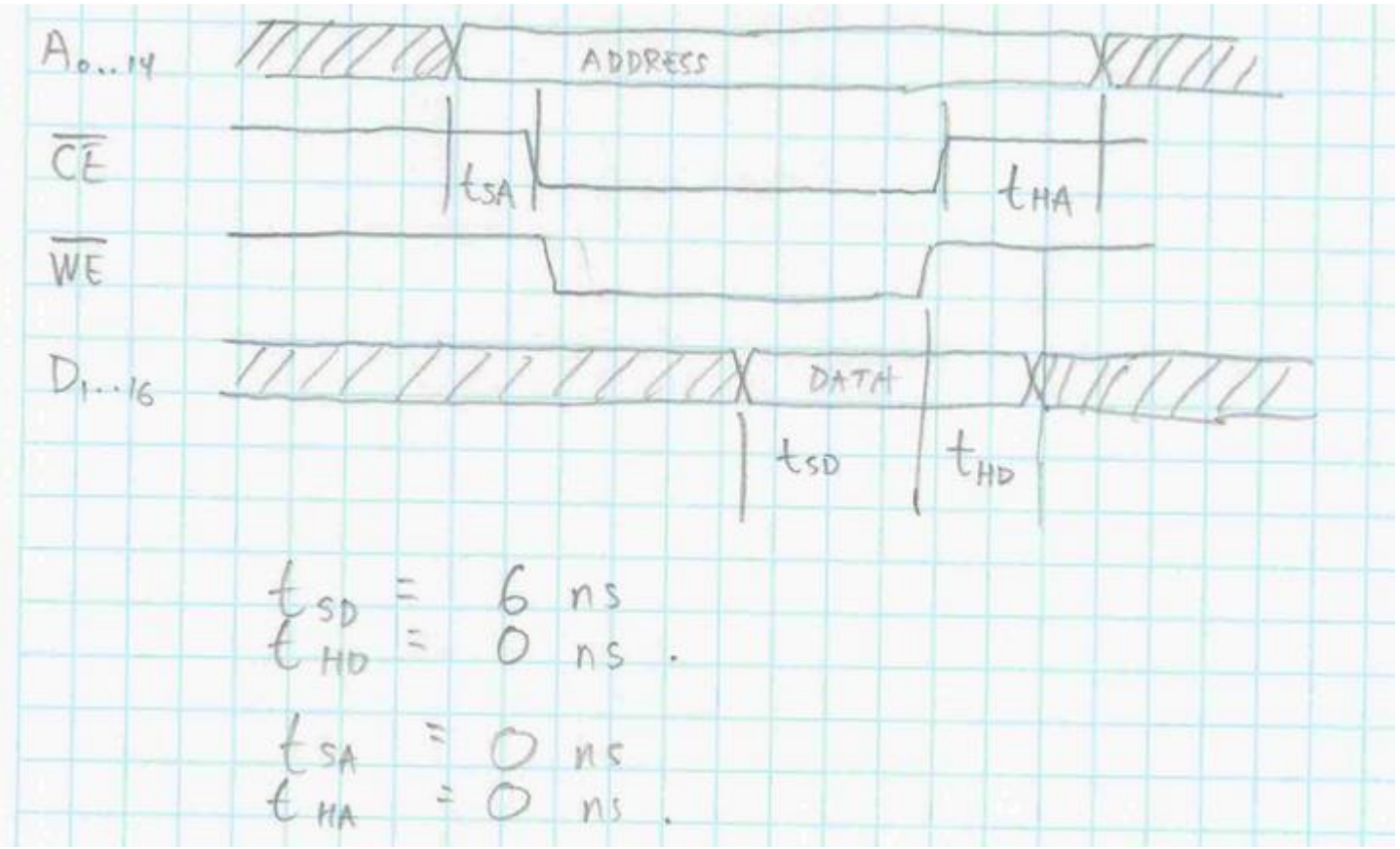


Pin Configuration

TSOP II
Top View

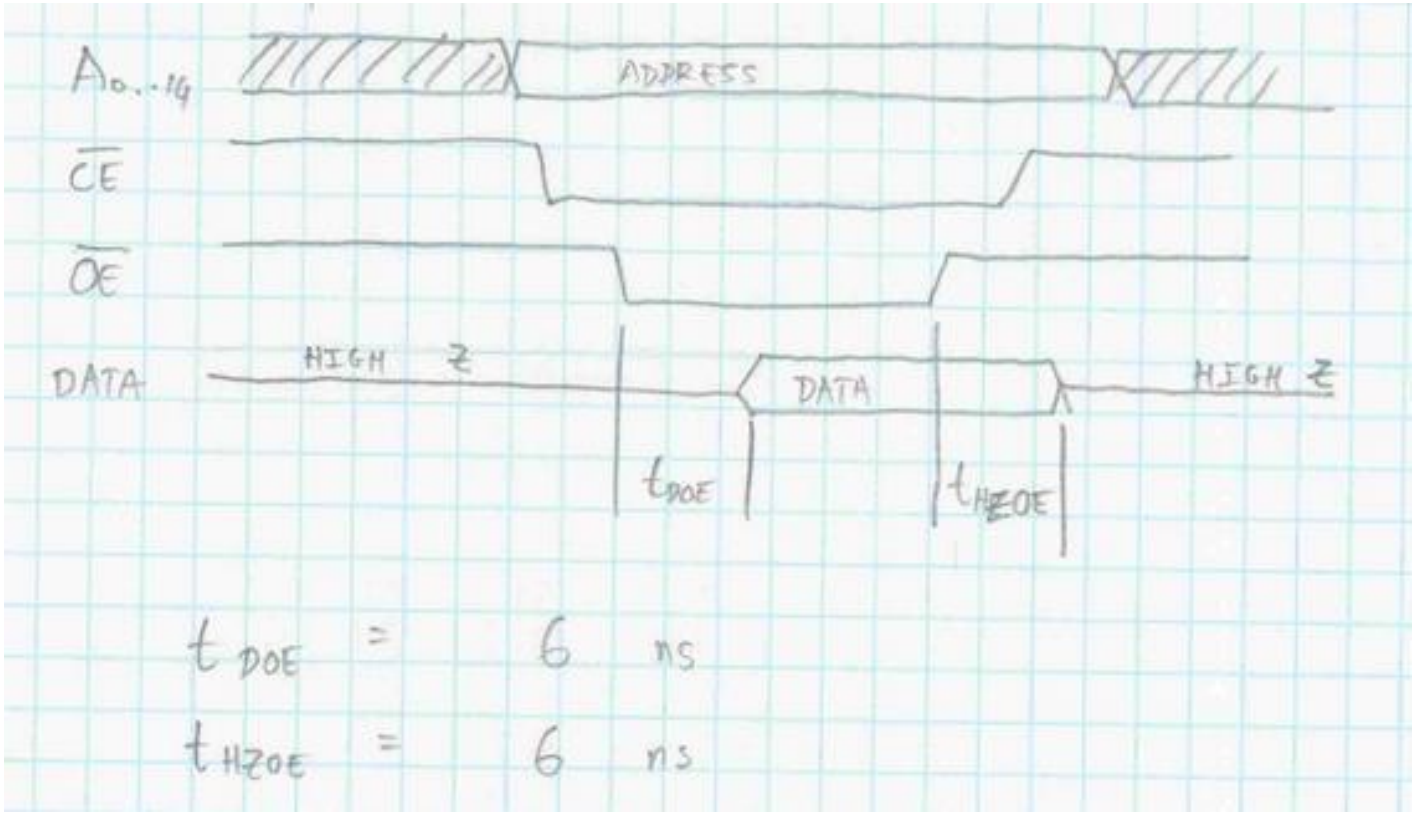
Memory Interface

- Write operation:



Memory Interface

- Read operation:

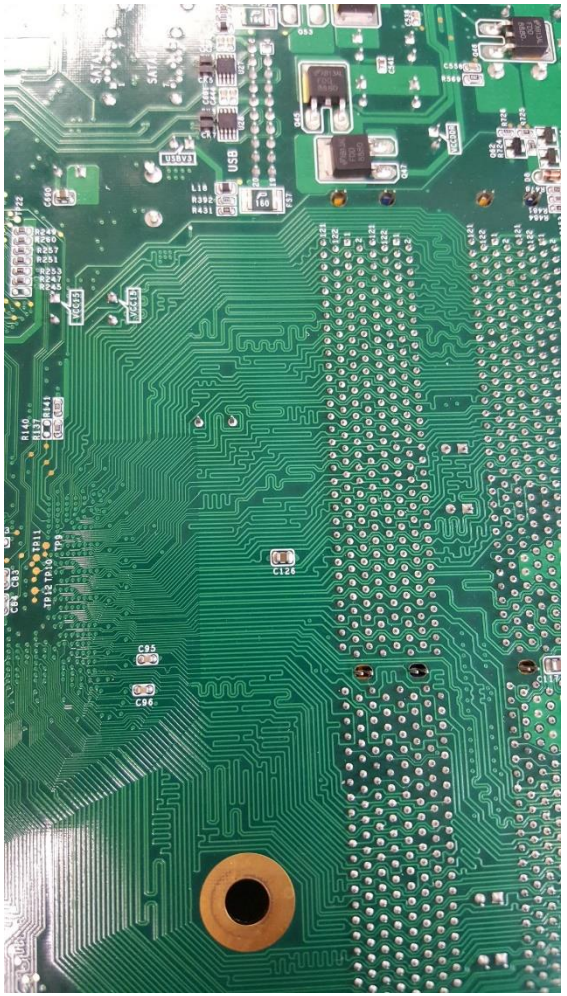


Memory Interfaces

- What is the maximum data transfer rate when writing?
- In principle, you could transfer 16 bits every 6 ns… this would be 2.667 Gbits per second.
- In practice it will be slower…
  - The signals on all the data lines might not all arrive at the same time (skew)
  - Propagation speed is about ½ c
  - Electrical connections that are physically close need extra delay (extra length) added so that all signals arrive at the same time

Correcting for Signal Skew

- You can see examples of this on most modern PCB layouts.
- The modelling of high-speed signals on printed circuit boards can be a very important part of the design process.



Serial Data Transfer

- Sometimes, high data transfer rates are not required and parallel interfaces present some disadvantages:
  - They require lots of physical resources (lots of pins on components, lots of wires in cables, lots of board space for routing)
- When data rates are not critical it can be convenient to transfer data one bit at a time.
  - This requires a data format specification

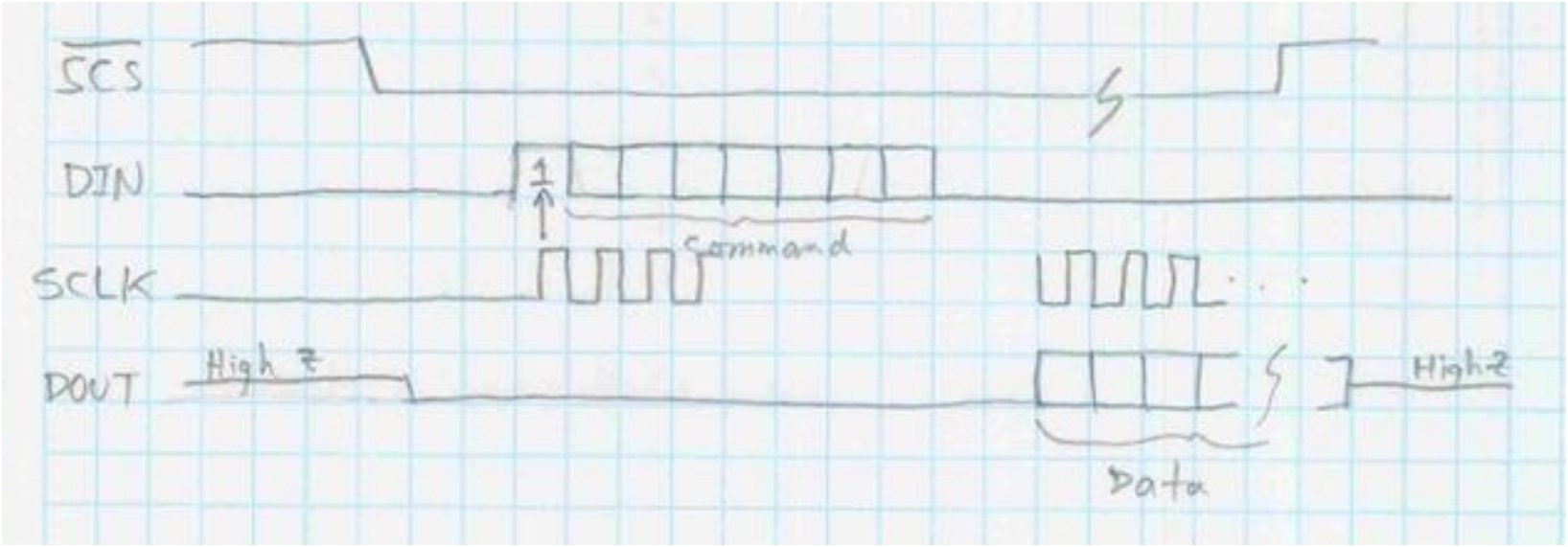
Serial Peripheral Interface

- The SPI interface is quite simple and is basically a standard for many integrated circuits that have internal registers that need to be programmed
  - Examples of actual devices that use the SPI interface will follow…
- The SPI interface (usually) has 4 signals:
  - SDIN (serial data in)
  - SDOUT (serial data out)
  - SCLK (serial clock)
  - $\overline{\text{SCS}}$ (serial chip select – active low)

SPI Interface

- Electrical specification:
  - When $\overline{SCS}$ is high (de-asserted), then SDOUT will be in a high impedance state.
  - This allows many devices to share the SDIN/SDOUT/SCLK signals provided ONLY ONE has $\overline{SCS}$ low at any given time.
  - Data is sampled on the rising edge of SCLK
  - SCLK doesn't have to have a fixed frequency
- This isn't really a formal standard, but it's simple enough that it is not hard to implement correctly.

Example SPI Interface



- SDIN is sampled on the rising edge of SCLK.
- The first ‘1’ bit after SCS goes low indicates the start of a “command” (for example - this is not always the case).
- If the command results in data being transferred out, it will appear on SDOUT.
- SDOUT is asserted on the falling clock edge so that the host can sample it on the rising clock edge.

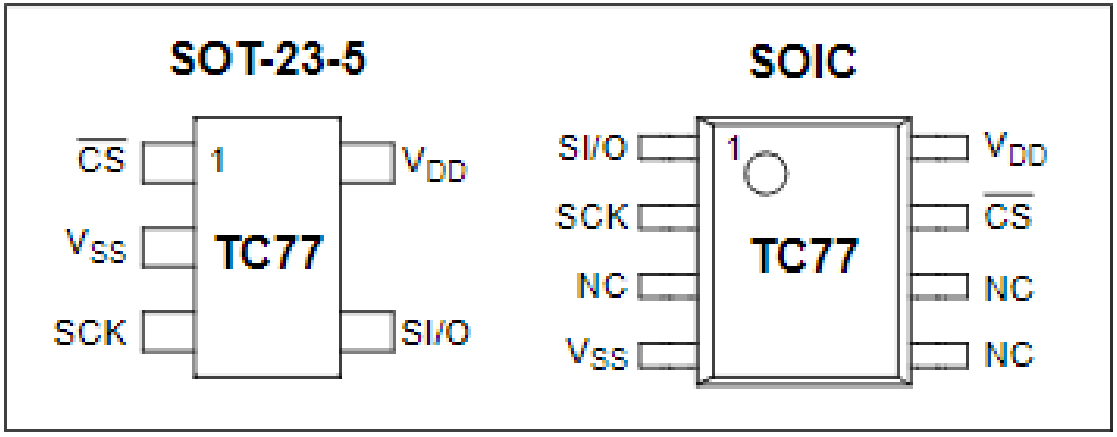
SPI Interface Example

MICROCHIP

TC77

Thermal Sensor with SPI Interface

Package Types



SPI Interface Example

- Reading the temperature:

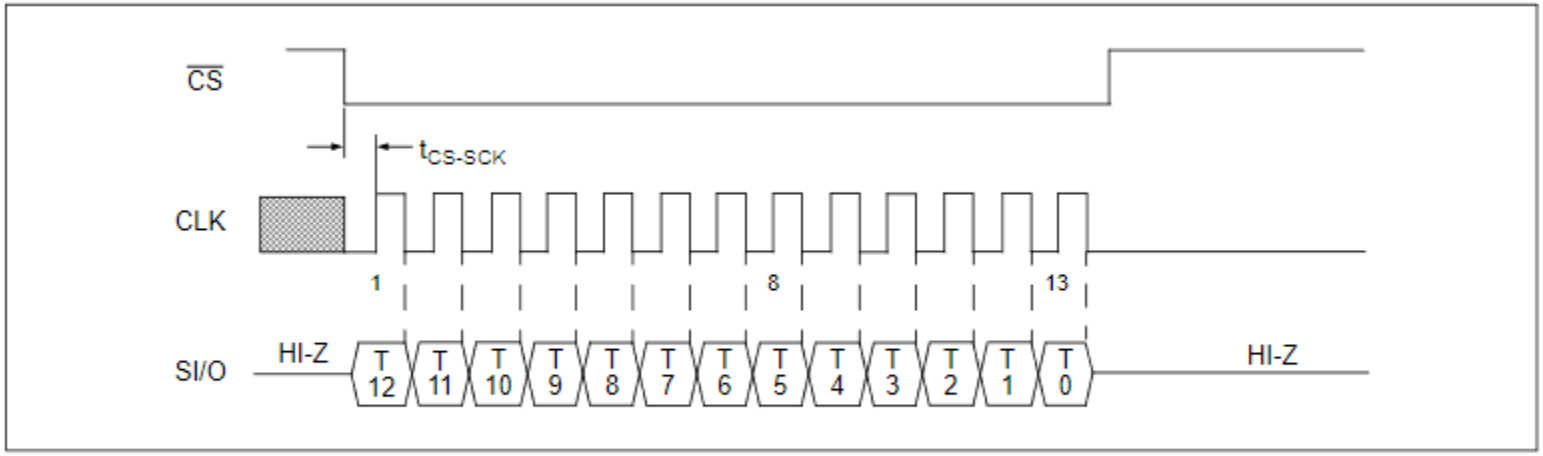
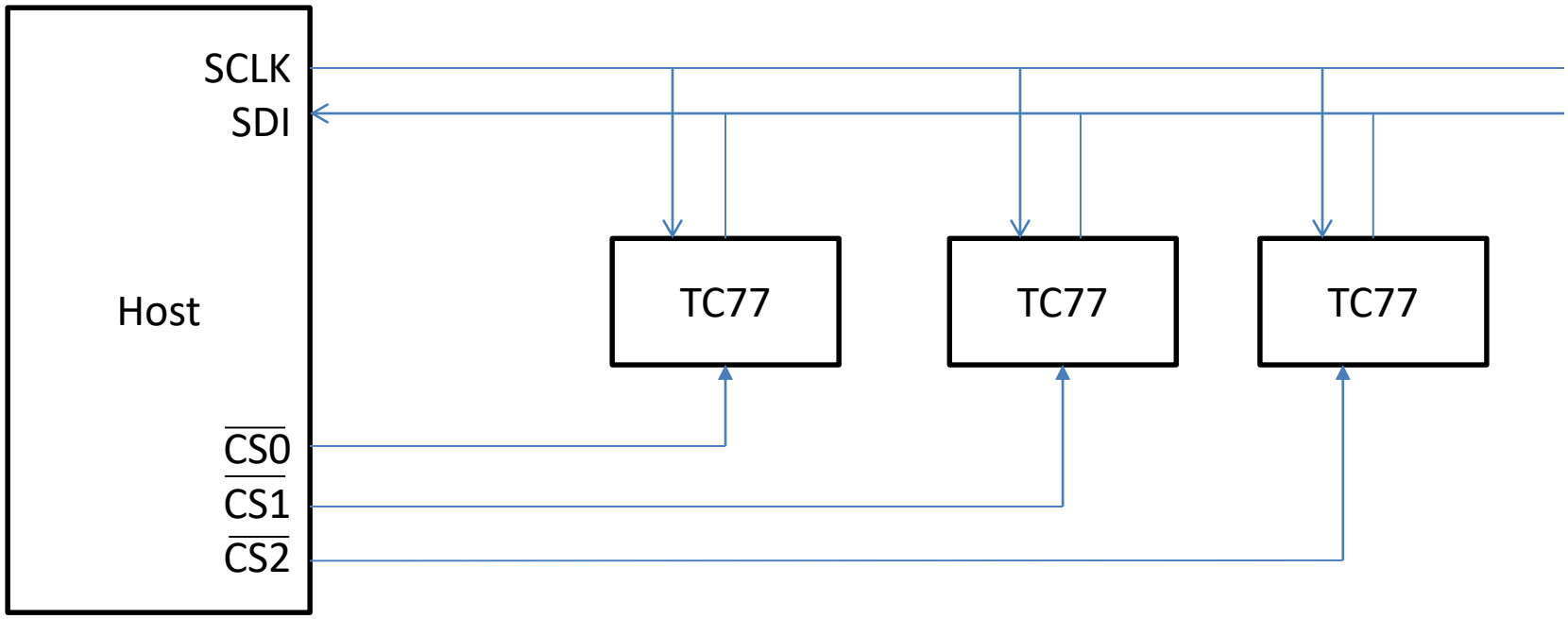


FIGURE 3-2: *Temperature Read Timing Diagram - (Reading only the first 13 Bits of the Temperature Register).*

SPI Interface Example

- You can connect many devices to the same bus and talk to them individually using different $\overline{CS}$ signals:

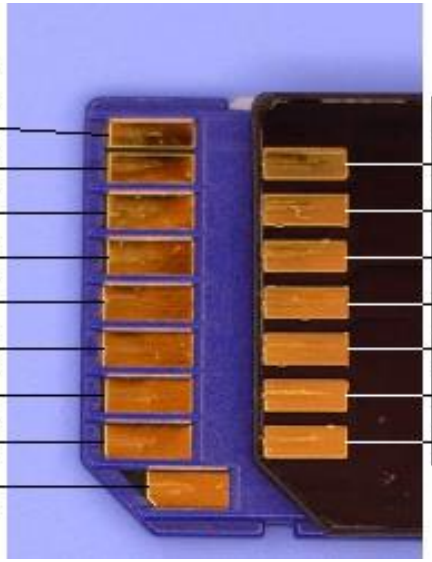


Example: SD memory interface

- Secure Digital Memory Cards (SD cards) can be used with an SPI interface:

SDC

No	SD	SPI
8	DAT1	
7	DAT0	DO
6	Vss2	
5	CLK	SCLK
4	Vcc	
3	Vss1	
2	CMD	DI
1	CAT3	CS
9	DAT2	



MMC

No	MMC	SPI
7	DAT	DO
6	Vss2	
5	CLK	SCLK
4	Vcc	
3	Vss1	
2	CMD	DI
1	RES	CS

- In this case we only care about the DO, SCLK, DI and CS signals.

SD Memory Interface

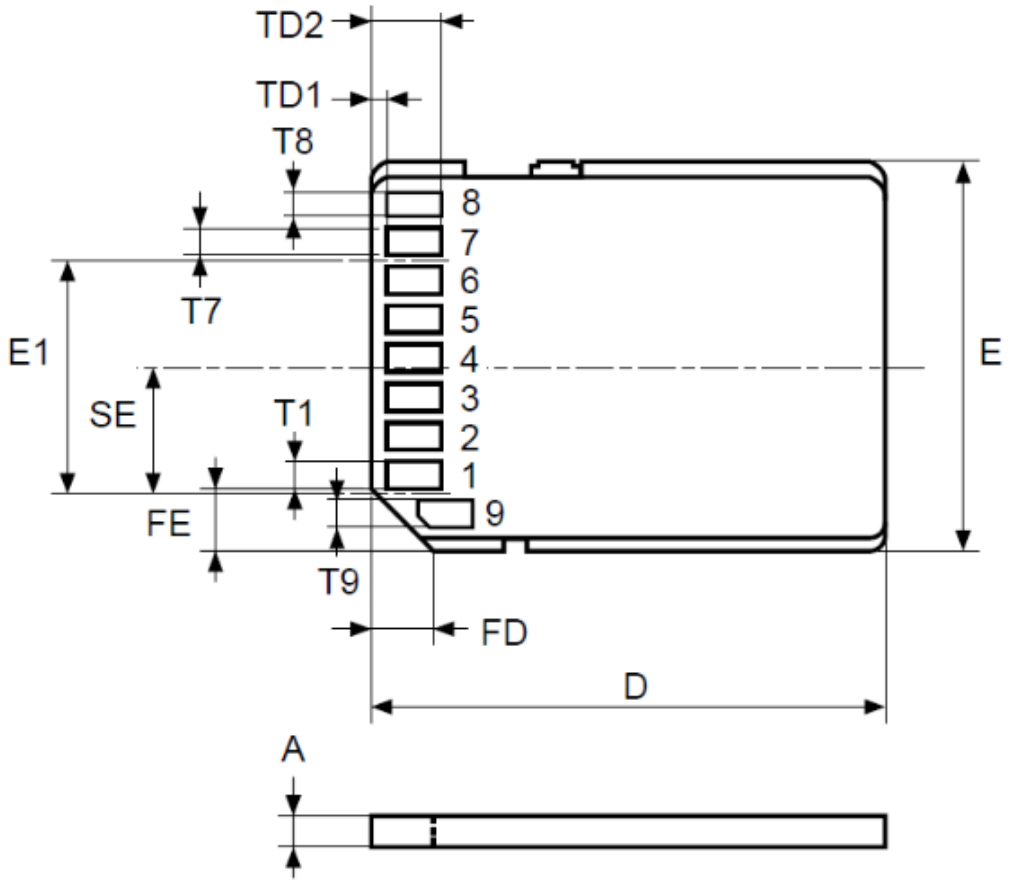
- The interface must be a standard to ensure interoperability of all SD cards and devices.
- One example of a complete specification can be found here:

https://media.digikey.com/pdf/Data%20Sheets/Viking%20PDFs/PSFSD3yyyyQyyyy.pdf

- These devices have two operating modes (SD mode and SPI mode) but we will only focus on the SPI mode.

SD Card Mechanical Interface

Figure 5-1: 2.5” SD Case Dimensions: 24mm (W) x 32mm (L) x 2.1mm (H)



Note: Drawing is not to scale

SD Card Physical Interface

4.1.1 SD Bus Pin Assignment

Table 4-1: SD Bus Pin Assignment

Pin #	SD Mode			SPI Mode		
	Name	Type[1]	Description	Name	Type[1]	Description
1	CD/DAT3[2]	I/O/PP[3]	Card Detect/ Data Line [Bit 3]	CS	I[3]	Chip Select (neg true)
2	CMD	I/O/PP	Command/Response	DI	I	Data In
3	VSS1	S	Supply voltage ground	VSS	S	Supply voltage ground
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V_{SS2}	S	Supply voltage ground	V_{SS2}	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO	O/PP	Data Out
8	DAT1	I/O/PP	Data Line [Bit 1]	RSV		
9	DAT2	I/O/PP	Data Line [Bit 2]	RSV		

SD Card SPI Interface

- Data transfer is performed as previously described.
  - there is a command sent to the DI input…
  - …possibly followed by additional data
  - Data is read back from the DO output.
- All that remains to be specified is how to send commands and interpret the data that is sent back.
- You can find examples of the standard set of commands online
  - They consist of things like, “initialize”, “block read”, and “block write”