APPENDIX C

IC INTERFACING AND SYSTEM DESIGN ISSUES

OVERVIEW

This appendix provides an overview of IC technology and AVR interfacing. In addition, we look at the microcontroller-based system as a whole and examine some general issues in system design.

First, in Section C.1, we provide an overview of IC technology. Then, in Section C.2, the internal details of AVR I/O ports and interfacing are discussed. Section C.3 examines system design issues.

C.1: OVERVIEW OF IC TECHNOLOGY

In this section we examine IC technology and discuss some major developments in advanced logic families. Because this is an overview, it is assumed that the reader is familiar with logic families on the level presented in basic digital electronics books.

Transistors

The transistor was invented in 1947 by three scientists at Bell Laboratory. In the 1950s, transistors replaced vacuum tubes in many electronics systems, including computers. It was not until 1959 that the first integrated circuit was successfully fabricated and tested by Jack Kilby of Texas Instruments. Prior to the invention of the IC, the use of transistors, along with other discrete components such as capacitors and resistors, was common in computer design. Early transistors were made of germanium, which was later abandoned in favor of silicon. This was because the slightest rise in temperature resulted in massive current flows in germanium-based transistors. In semiconductor terms, it is because the band gap of germanium is much smaller than that of silicon, resulting in a massive flow of electrons from the valence band to the conduction band when the temperature rises even slightly. By the late 1960s and early 1970s, the use of the silicon-based IC was widespread in mainframes and minicomputers. Transistors and ICs at first were based on P-type materials. Later on, because the speed of electrons is much higher (about two-and-a-half times) than the speed of holes, N-type devices replaced P-type devices. By the mid-1970s, NPN and NMOS transistors had replaced the slower PNP and PMOS transistors in every sector of the electronics industry, including in the design of microprocessors and computers. Since the early 1980s, CMOS (complementary MOS) has become the dominant technology of IC design. Next we provide an overview of differences between MOS and bipolar transistors. See Figure C-1.

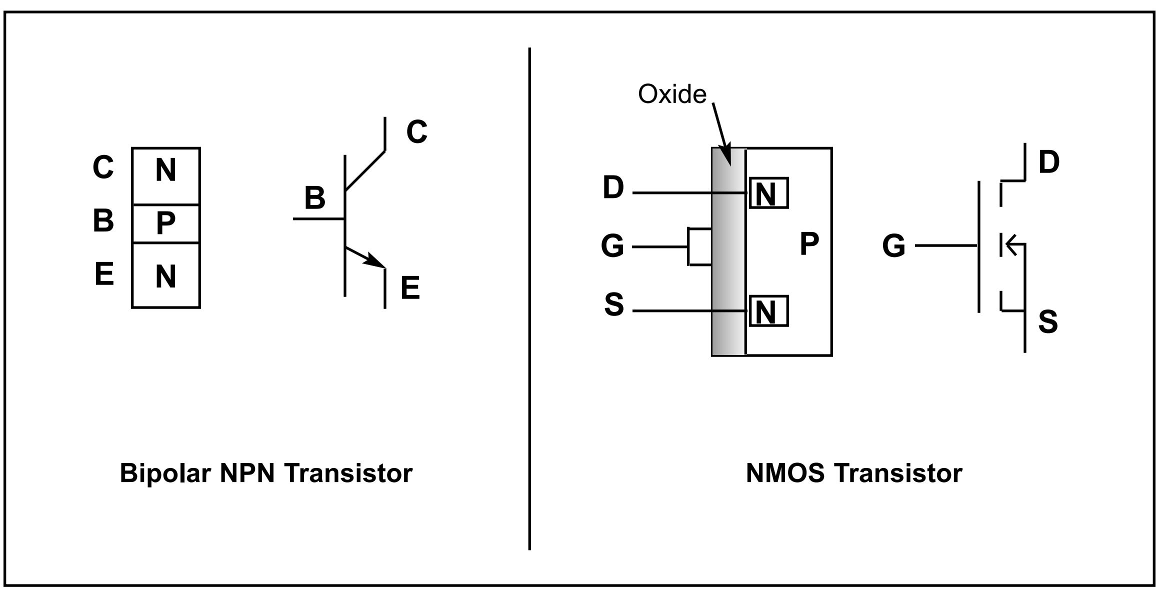


Figure C-1. Bipolar vs. MOS Transistors

MOS vs. bipolar transistors

There are two types of transistors: bipolar and MOS (metal-oxide semiconductor). Both have three leads. In bipolar transistors, the three leads are referred to as the *emitter*, *base*, and *collector*, while in MOS transistors they are named *source*, *gate*, and *drain*. In bipolar transistors, the carrier flows from the emitter to the collector, and the base is used as a flow controller. In MOS transistors, the carrier flows from the source to the drain, and the gate is used as a flow controller. In NPN-type bipolar transistors, the electron carrier leaving the emitter must overcome two voltage barriers before it reaches the collector (see Figure C-1). One is the N-P junction of the emitter-base and the other is the P-N junction of the base-collector. The voltage barrier of the base-collector is the most difficult one for the electrons to overcome (because it is reverse-biased) and it causes the most power dissipation. This led to the design of the unipolar type transistor called *MOS*. In N-channel MOS transistors, the electrons leave the source and reach the drain without going through any voltage barrier. The absence of any voltage barrier in the path of the carrier is one reason why MOS dissipates much less power than bipolar transistors. The low power dissipation of MOS allows millions of transistors to fit on a single IC chip. In today's technology, putting 10 million transistors into an IC is common, and it is all because of MOS technology. Without the MOS transistor, the advent of desktop personal computers would not have been possible, at least not so soon. The bipolar transistors in both the mainframes and minicomputers of the 1960s and 1970s were bulky and required expensive cooling systems and large rooms. MOS transistors do have one major drawback: They are slower than bipolar transistors. This is due partly to the gate capacitance of the MOS transistor. For a MOS to be turned on, the input capacitor of the gate takes time to charge up to the turn-on (threshold) voltage, leading to a longer propagation delay.

Overview of logic families

Logic families are judged according to (1) speed, (2) power dissipation, (3) noise immunity, (4) input/output interface compatibility, and (5) cost. Desirable qualities are high speed, low power dissipation, and high noise immunity (because it prevents the occurrence of false logic signals during switching transition). In interfacing logic families, the more inputs that can be driven by a single output, the better. This means that high-driving-capability outputs are desired. This, plus the fact that the input and output voltage levels of MOS and bipolar transistors are not compatible mean that one must be concerned with the ability of one logic family to drive the other one. In terms of the cost of a given logic family, it is high during the early years of its introduction but it declines as production and use rise.

The case of inverters

As an example of logic gates, we look at a simple inverter. In a one-transistor inverter, the transistor plays the role of a switch, and R is the pull-up resistor. See Figure C-2. For this inverter to work most effectively in digital circuits, however, the R value must be high when the transistor is "on" to limit the current flow from V_{CC} to ground in order to have low power dissipation (P = VI, where V

= 5 V). In other words, the lower the I, the lower the power dissipation. On the other hand, when the transistor is "off", R must be a small value to limit the voltage drop across R, thereby making sure that V_{OUT} is close to V_{CC}. This is a contradictory demand on R. This is one reason that logic gate designers use active components (transistors) instead of passive components (resistors) to implement the pull-up resistor R.

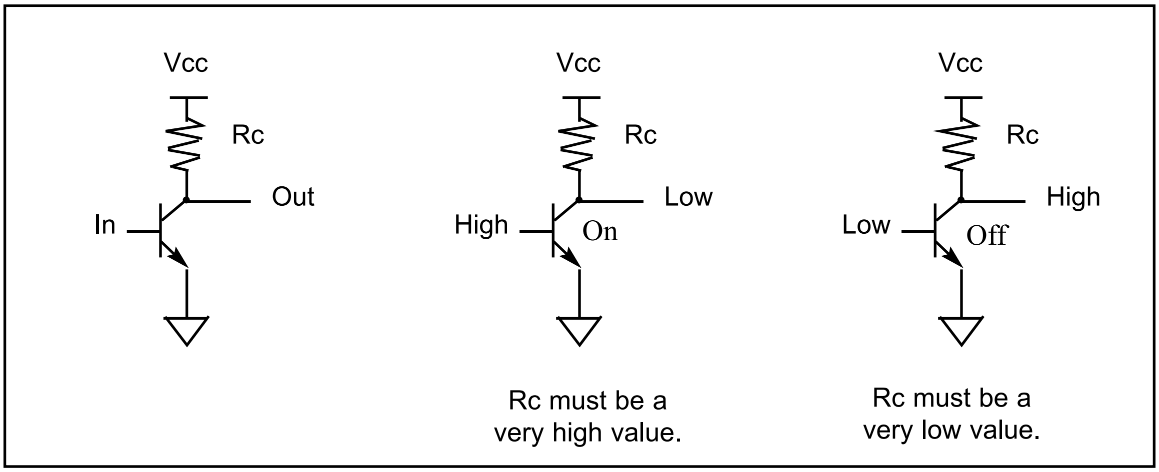


Figure C-2. One-Transistor Inverter with Pull-up Resistor

The case of a TTL inverter with totem-pole output is shown in Figure C-3. In Figure C-3, Q3 plays the role of a pull-up resistor.

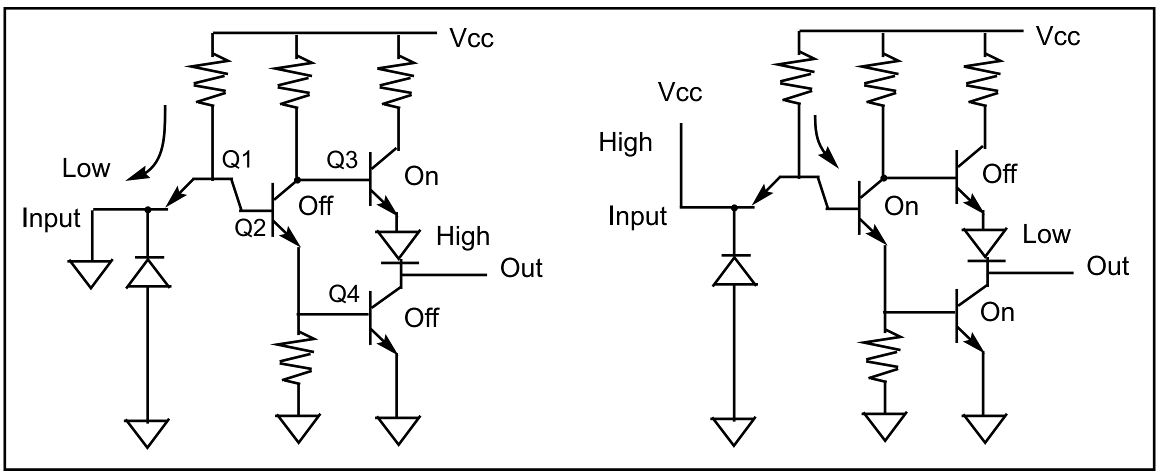


Figure C-3. TTL Inverter with Totem-Pole Output

CMOS inverter

In the case of CMOS-based logic gates, PMOS and NMOS are used to construct a CMOS (complementary MOS) inverter as shown in Figure C-4. In CMOS inverters, when the PMOS transistor is off, it provides a very high impedance path, making leakage current almost zero (about 10 nA); when the PMOS is on, it provides a low resistance on the path of V_{DD} to load. Because the speed of the hole is slower than that of the electron, the PMOS transistor is wider to compensate for this disparity; therefore, PMOS transistors take more space than NMOS transistors in the CMOS gates. At the end of this section we will see an open-collector gate in which the pull-up resistor is provided externally, thereby allowing system designers to choose the value of the pull-up resistor.

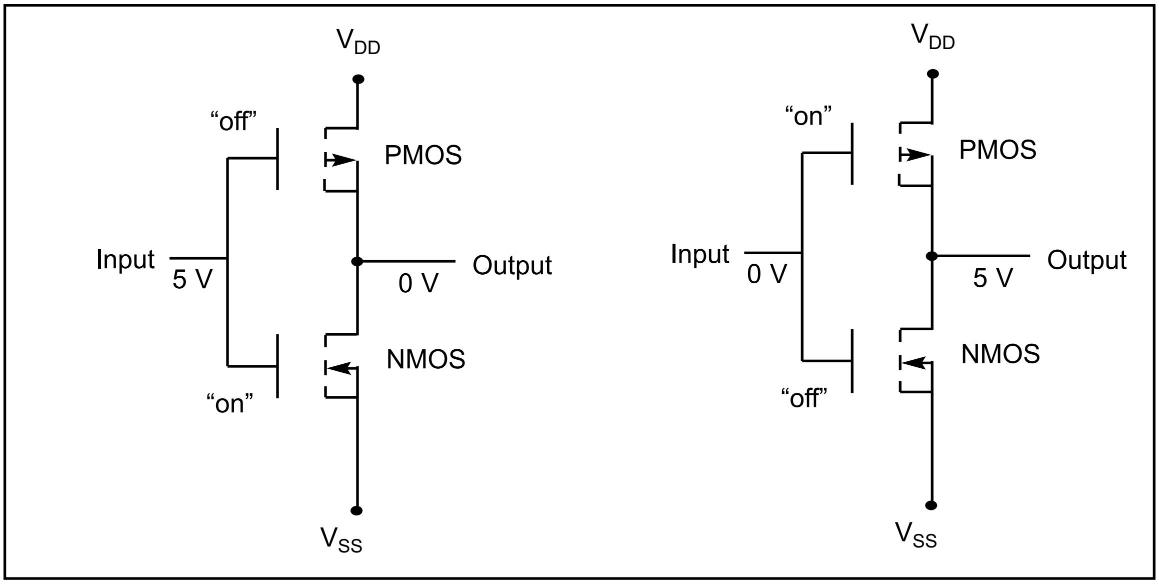


Figure C-4. CMOS Inverter

Input/output characteristics of some logic families

In 1968 the first logic family made of bipolar transistors was marketed. It was commonly referred to as the *standard TTL* (transistor-transistor logic) family. The first MOS-based logic family, the CD4000/74C series, was marketed in 1970. The addition of the Schottky diode to the base-collector of bipolar transistors in the early 1970s gave rise to the S family. The Schottky diode shortens the propagation delay of the TTL family by preventing the collector from going into what is called *deep saturation*. Table C-1 lists major characteristics of some logic families. In Table C-1, note that as the CMOS circuit's operating frequency rises, the power dissipation also increases. This is not the case for bipolar-based TTL.

Table C-1: Characteristics of Some Logic Families

Characteristic	STD TTL	LSTTL	ALSTTL	HCMOS
V_{CC}	5 V	5 V	5 V	5 V
V_{IH}	2.0 V	2.0 V	2.0 V	3.15 V
V_{IL}	0.8 V	0.8 V	0.8 V	1.1 V
V_{OH}	2.4 V	2.7 V	2.7 V	3.7 V
V_{OL}	0.4 V	0.5 V	0.4 V	0.4 V
I_{IL}	−1.6 mA	−0.36 mA	−0.2 mA	−1 μA
I_{IH}	40 μA	20 μA	20 μA	1 μA
I_{OL}	16 mA	8 mA	4 mA	4 mA
I_{OH}	−400 μA	−400 μA	−400 μA	4 mA
Propagation delay	10 ns	9.5 ns	4 ns	9 ns
Static power dissipation (f = 0)	10 mW	2 mW	1 mW	0.0025 nW
Dynamic power dissipation at f = 100 kHz	10 mW	2 mW	1 mW	0.17 mW

History of logic families

Early logic families and microprocessors required both positive and negative power voltages. In the mid-1970s, 5 V V_{CC} became standard. In the late 1970s, advances in IC technology allowed combining the speed and drive of the S family with the lower power of LS to form a new logic family called *FAST* (Fairchild Advanced Schottky TTL). In 1985, AC/ACT (Advanced CMOS Technology), a much higher speed version of HCMOS, was introduced. With the introduction of FCT (Fast CMOS Technology) in 1986, the speed gap between CMOS and TTL at last was closed. Because FCT is the CMOS version of FAST, it has the low power consumption of CMOS but the speed is comparable with TTL. Table C-2 provides an overview of logic families up to FCT.

Table C-2: Logic Family Overview

Product	Year Introduced	Speed (ns)	Static Supply Current (mA)	High/Low Family Drive (mA)
Std TTL	1968	40	30	−2/32
CD4K/74C	1970	70	0.3	−0.48/6.4
LS/S	1971	18	54	−15/24
HC/HCT	1977	25	0.08	−6/−6
FAST	1978	6.5	90	−15/64
AS	1980	6.2	90	−15/64
ALS	1980	10	27	−15/64
AC/ACT	1985	10	0.08	−24/24
FCT	1986	6.5	1.5	−15/64

Reprinted by permission of Electronic Design Magazine, c. 1991.

Recent advances in logic families

As the speed of high-performance microprocessors reached 25 MHz, it shortened the CPU's cycle time, leaving less time for the path delay. Designers normally allocate no more than 25% of a CPU's cycle time budget to path delay. Following this rule means that there must be a corresponding decline in the propagation delay of logic families used in the address and data path as the system frequency is increased. In recent years, many semiconductor manufacturers have responded to this need by providing logic families that have high speed, low noise, and high drive I/O. Table C-3 provides the characteristics of high-performance logic families introduced in recent years. ACQ/ACTQ are the second-generation advanced CMOS (ACMOS) with much lower noise. While ACQ has the CMOS input level, ACTQ is equipped with TTL-level input. The FCTx and FCTx-T are second-generation FCT with much higher speed. (The "x" in the FCTx and FCTx-T refers to various speed grades, such as A, B, and C, where A means low speed and C means high speed.) For designers who are well versed in using the FAST logic family, FASTr is an ideal choice because it is faster than FAST, has higher driving capability (I_{OL}, I_{OH}), and produces much lower noise than FAST. At the time of this writing, next to ECL and gallium arsenide logic gates, FASTr is the fastest logic family in the market (with the 5 V V_{CC}), but the power consumption is high relative to other logic families, as shown in Table C-3. The combining of high-speed bipolar TTL and the low power consumption of CMOS has given birth to what is called *BICMOS*. Although BICMOS seems to be the future trend in IC design, at this time it is expensive due to extra steps required in BICMOS IC fabrication, but in some cases there is no other choice. (For example, Intel's Pentium microprocessor, a BICMOS product, had to use high-speed bipolar transistors to speed up some of the internal functions.) Table C-3 provides advanced logic characteristics. The "x" is for different speeds designated as A, B, and C. A is the slowest one while C is the fastest one. The above data is for the 74244 buffer.

Table C-3: Advanced Logic General Characteristics

Family	Year	Number Suppliers	Tech Base	I/O Level	Speed (ns)	Static Current	I_{OH}/I_{OL}
ACQ	1989	2	CMOS	CMOS/CMOS	6.0	80 μA	−24/24 mA
ACTQ	1989	2	CMOS	TTL/CMOS	7.5	80 μA	−24/24 mA
FCTx	1987	3	CMOS	TTL/CMOS	4.1–4.8	1.5 mA	−15/64 mA
FCTxT	1990	2	CMOS	TTL/TTL	4.1–4.8	1.5 mA	−15/64 mA
FASTr	1990	1	Bipolar	TTL/TTL	3.9	50 mA	−15/64 mA
BCT	1987	2	BICMOS	TTL/TTL	5.5	10 mA	−15/64 mA

Since the late 1970s, the use of a +5 V power supply has become standard in all microprocessors and microcontrollers. To reduce power consumption, 3.3 V V_{CC} is being embraced by many designers. The lowering of V_{CC} to 3.3 V has two major advantages: (1) It lowers the power consumption, prolonging the life of the battery in systems using a battery, and (2) it allows a further reduction of line size (design rule) to submicron dimensions. This reduction results in putting more transistors in a given die size. As fabrication processes improve, the decline in the line size is reaching submicron level and transistor densities are approaching 1 billion transistors.

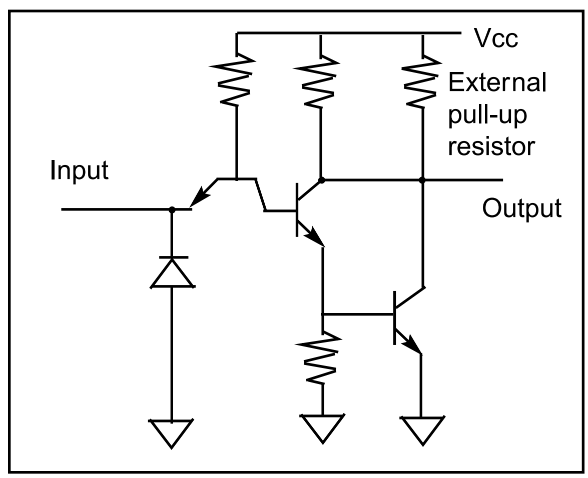


Figure C-5. Open Collector

Open-collector and open-drain gates

To allow multiple outputs to be connected together, we use open-collector logic gates. In such cases, an external resistor will serve as load. This is shown in Figures C-5 and C-6.

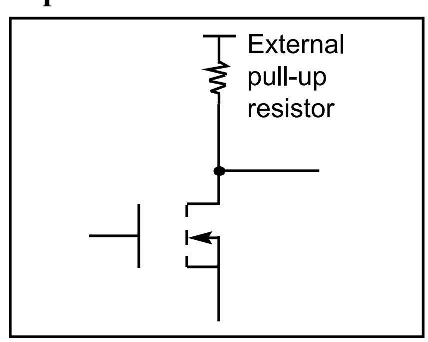


Figure C-6. Open Drain

SECTION C.2: AVR I/O PORT STRUCTURE AND INTERFACING

In interfacing the AVR microcontroller with other IC chips or devices, fan-out is the most important issue. To understand the AVR fan-out we must first understand the port structure of the AVR. This section provides a detailed discussion of the AVR port structure and its fan-out. It is very critical that we understand the I/O port structure of the AVR lest we damage it while trying to interface it with an external device.

IC fan-out

When connecting IC chips together, we need to find out how many input pins can be driven by a single output pin. This is a very important issue and involves the discussion of what is called *IC fan-out*. The IC fan-out must be addressed for both logic "0" and logic "1" outputs. See Example C-1. Fan-out for logic LOW and fan-out for logic HIGH are defined as follows:

$$\text{fan-out (of LOW)} = \frac{I_{OL}}{I_{IL}} \qquad \text{fan-out (of HIGH)} = \frac{I_{OH}}{I_{IH}}$$

Of the above two values, the lower number is used to ensure the proper noise margin. Figure C-7 shows the sinking and sourcing of current when ICs are connected together.

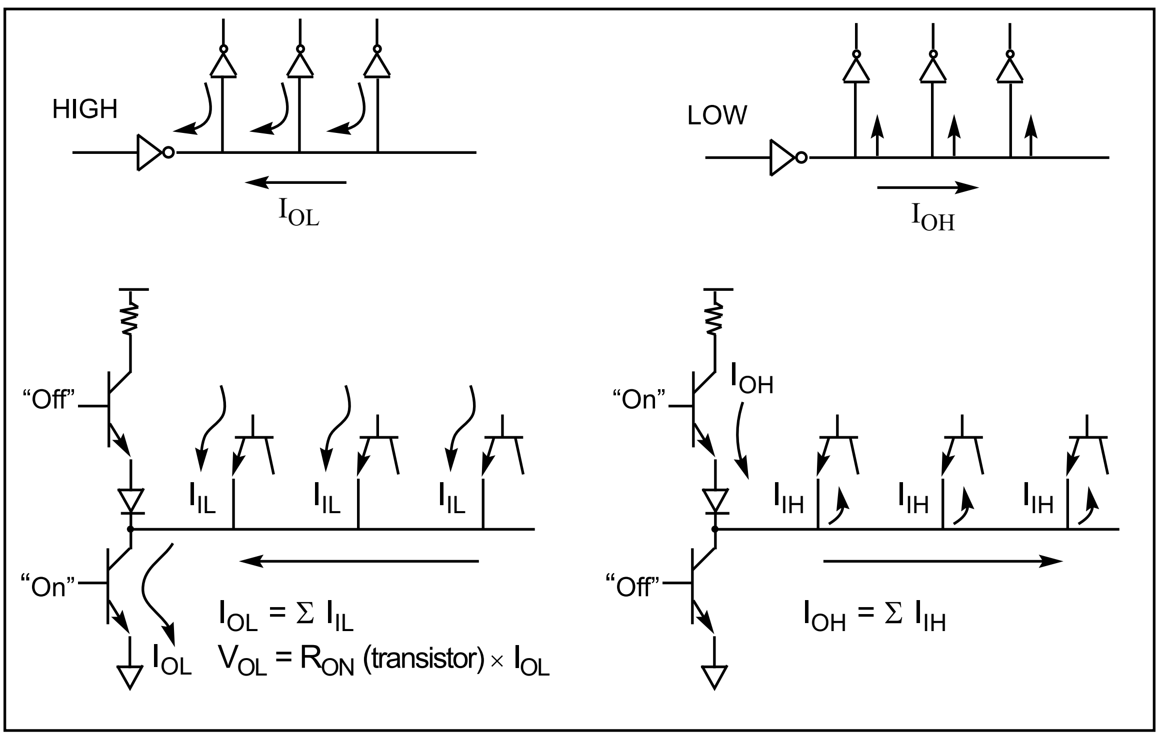


Figure C-7. Current Sinking and Sourcing in TTL

Notice that in Figure C-7, as the number of input pins connected to a single output increases, I_{OL} rises, which causes V_{OL} to rise. If this continues, the rise of V_{OL} makes the noise margin smaller, and this results in the occurrence of false logic due to the slightest noise.

Example C-1
Find how many unit loads (UL) can be driven by the output of the LS logic family.

Solution:

The unit load is defined as I_{IL} = 1.6 mA and I_{IH} = 40 μA. Table C-1 shows I_{OH} = 400 μA and I_{OL} = 8 mA for the LS family. Therefore, we have

$$\text{fan-out (LOW)} = \frac{I_{OL}}{I_{IL}} = \frac{8\ \text{mA}}{1.6\ \text{mA}} = 5$$

$$\text{fan-out (HIGH)} = \frac{I_{OH}}{I_{IH}} = \frac{400\ \mu\text{A}}{40\ \mu\text{A}} = 10$$

This means that the fan-out is 5. In other words, the LS output must not be connected to more than 5 inputs with unit load characteristics.

74LS244 and 74LS245 buffers/drivers

In cases where the receiver current requirements exceed the driver's capability, we must use buffers/drivers such as the 74LS245 and 74LS244. Figure C-8 shows the internal gates for the 74LS244 and 74LS245. The 74LS245 is used for bidirectional data buses, and the 74LS244 is used for unidirectional address buses.

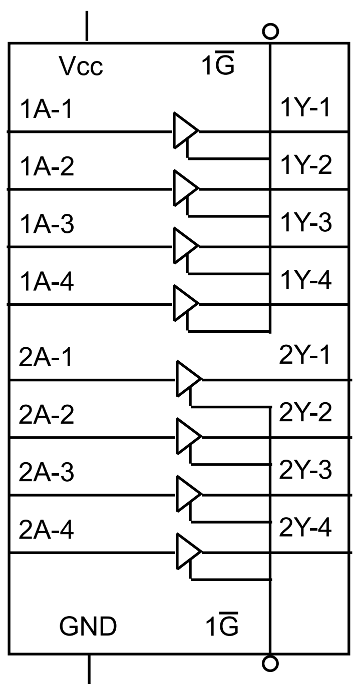
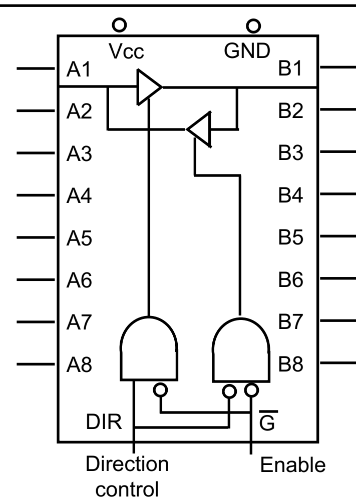


Figure C-8 (a). 74LS244 Octal Buffer
(Reprinted by permission of Texas Instruments, Copyright Texas Instruments, 1988)



Function Table

Enable G̅	Direction control DIR	Operation
L	L	B Data to A Bus
L	H	A Data to B Bus
H	X	Isolation

Figure C-8 (b). 74LS245 Bidirectional Buffer
(Reprinted by permission of Texas Instruments, Copyright Texas Instruments, 1988)

Tri-state buffer

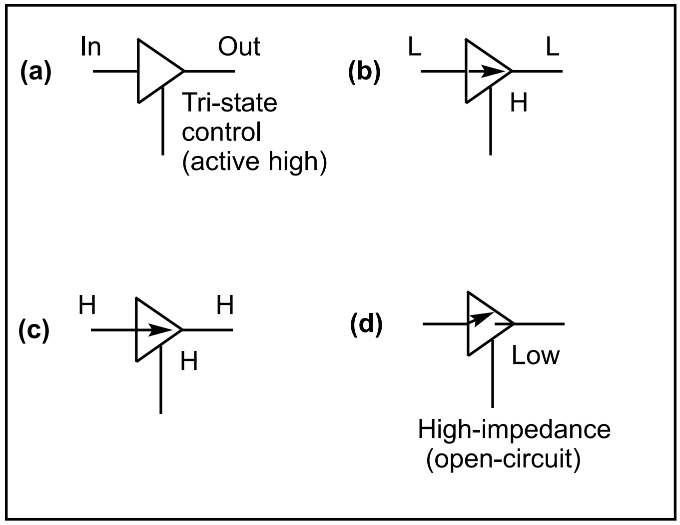


Figure C-9. Tri-State Buffer

Notice that the 74LS244 is simply 8 tri-state buffers in a single chip. As shown in Figure C-9 a tri-state buffer has a single input, a single output, and the enable control input. By activating the enable, data at the input is transferred to the output. The enable can be an active-LOW or an active-HIGH. Notice that the enable input for the 74LS244 is an active-LOW whereas the enable input pin for Figure C-9 is active-HIGH.

74LS245 and 74LS244 fan-out

It must be noted that the output of the 74LS245 and 74LS244 can sink and source a much larger amount of current than that of other LS gates. See Table C-4. That is the reason we use these buffers for driver when a signal is travelling a long distance through a cable or it has to drive many inputs.

Table C-4: Electrical Specifications for Buffers/Drivers

	I_{OH} (mA)	I_{OL} (mA)
74LS244	3	12
74LS245	3	12

After this background on the fan-out, next we discuss the structure of AVR ports.

AVR port structure and operation

All the ports of the AVR are bidirectional. They all have three registers that can be accessed by IN and OUT instructions. We will descuss each register in detail.

PORTx register

As you can see in Figure C-10, the PORTx register can be accessed using read and write operations. When we want to write to PORTx, we use the "OUT PORTx, Rr" instruction. In this case, the WR-PORTx pin is set high and Rr is loaded into PORTx.

When we want to read from PORTx, we use "IN Rd, PORTx". In this case, the PRx pin is set to HIGH, which enables the buffer and makes it possible to read from PORTx.

The output of PORTx is either connected to the Px pin of the chip or con-

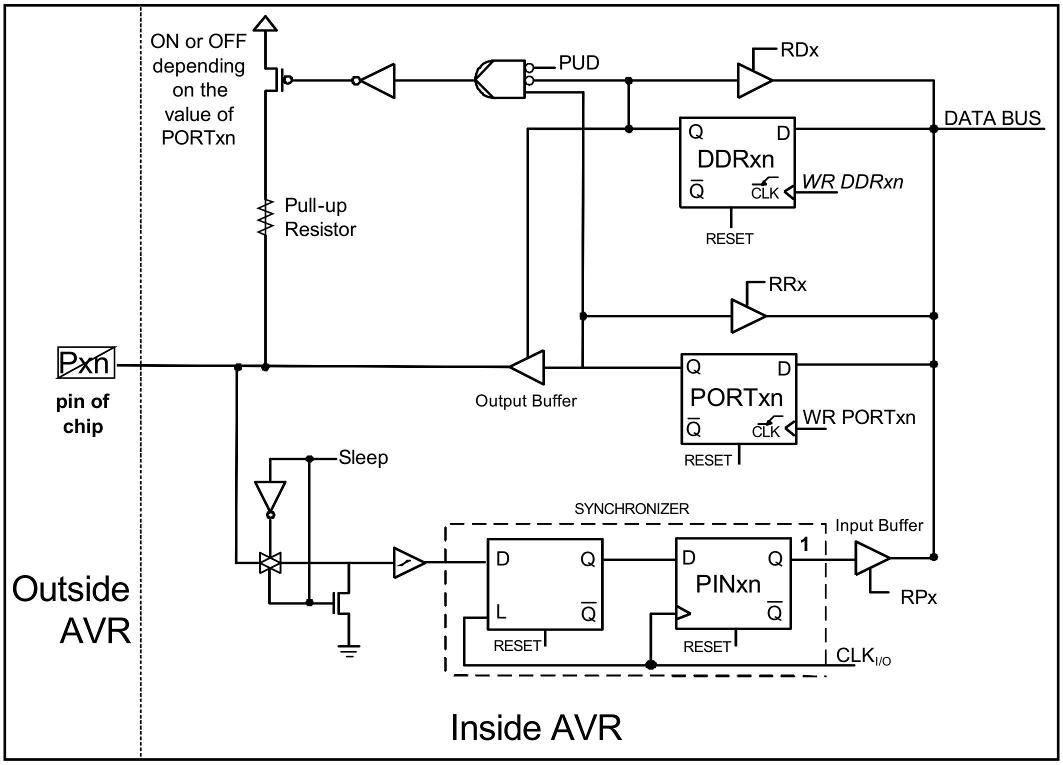


Figure C-10. The AVR Ports Structure

trols the pull-up resistor, as we will see next.

DDRx register

As shown in Figure C-10, the DDRx register can be accessed using read and write operations. When we want to write to DDRx, we use "OUT DDRx, Rr". In this case, the WR-DDRx pin is set to HIGH and enables writing to DDRx. When we want to read from DDRx, we use "IN Rd, DDRx". In this case, the RDx pin is set to LOW, which enables the buffer and makes it possible to read from DDRx.

The DDRx register controls the output buffer and the pull-up resistor. When the Q of DDRx is HIGH, it enables the output buffer and connects the Q of the PORTx register to the Px pin of the chip. In this case, the pin is configured as output. When the Q of DDRx is LOW, it disables the output buffer and configures the Px pin of the chip as input. In this case, assuming that the PUD bit is LOW, the Q of PORTx controls the pull-up resistor. When the Q of PORTx is HIGH, it enables the pull-up resistor, and when it is LOW, it disables the pull-up resistor.

PINx register

As you see in Figure C-10, when the AVR is not in sleep mode, the PINxn flip-flop is loaded with the value of the AVR pin on each machine cycle. Therefore, to read the current state of the Px pin of the chip, we should read the content of the PINx register. To do so, we use "IN Rd,PINx", which sets RPx high and enables the input buffer. In this case, the value of PINx passes through the internal data bus of AVR and will be loaded into the Rd register.

Reading the pin when DDRx.n = 0 (Input)

As we stated in Chapter 4, to make any bits of any port of the AVR an input port, we first must write a 0 (logic LOW) to the DDRx.n bit. Look at the following sequence of events to see why:

1. As can be seen from Figure C-11, if we write 0 to the DDRx.n, it will have "LOW" on its Q. This turns off the tri-state buffer.
2. When the tri-state buffer is off, it blocks the path from the Q of PORTx.n to the pin of chip, and the input signal is directed to the PINx.n buffer.
3. When reading the input port in instructions such as "`IN R16,PINB`" we are reading the data present at the pin. In other words, it is bringing into the CPU the status of the external pin. This instruction activates the read pin of the buffer and lets data at the pins flow into the CPU's internal bus. Figure C-11 shows how the input circuit works.

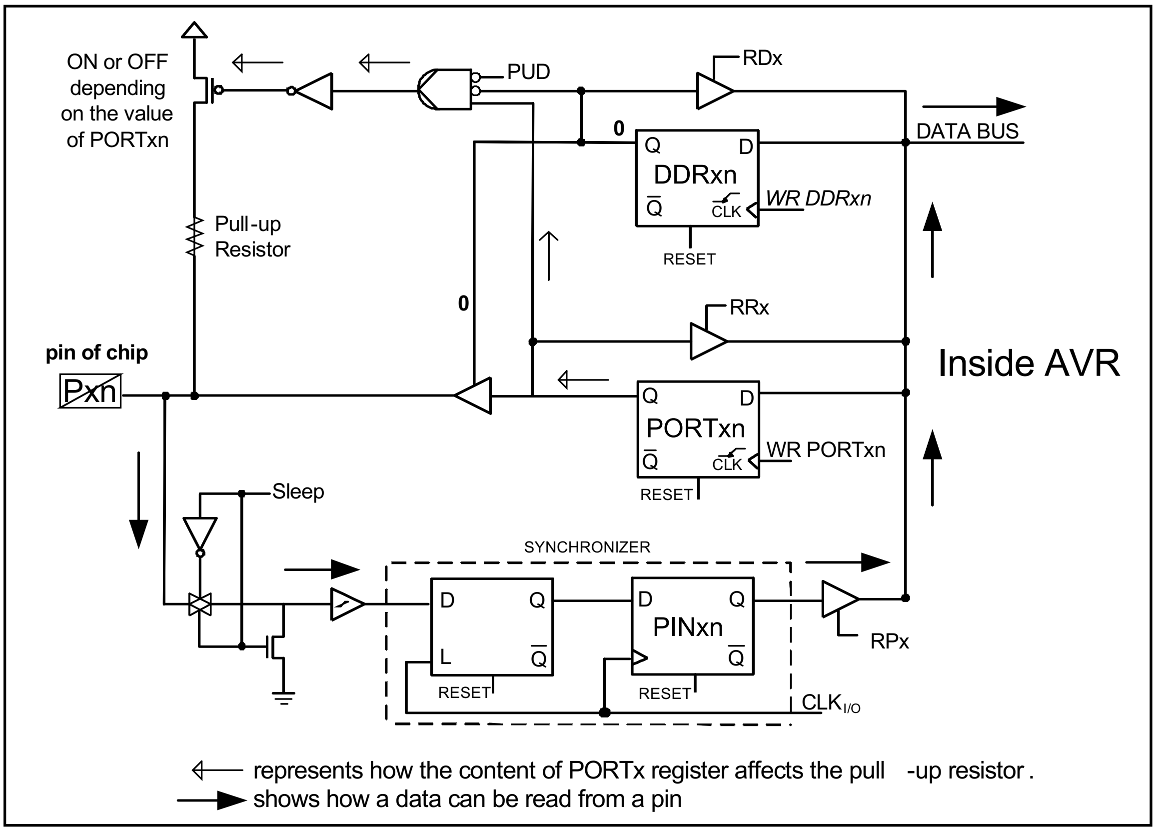


Figure C-11. Inputting (Reading) from a Pin via a PINx Register in the AVR

Writing to pin when DDRx.n = 1 (Output)

The above discussion showed why we must write a "LOW" to a port's DDRx.n bits in order to make it an input port. What happens if we write a "1" to DDRx.n that was configured as an input port? From Figure C-12 we see that when DDRx.n = 1, the DDRx.n latch has "HIGH" on its Q. This turns on the tri-state buffer, and the data of PORTx.n is transferred to the pin of chip.

From Figure C-12 we see that when DDRx.n = 1, if we write a 0 to the PORTx.n latch, then PORTx.n has "LOW" on its Q. This provides 0 to the pin of chip. Therefore, any attempt to read the input pin will always get the "LOW" ground signal. Figure C-13 shows what happens if we write “HIGH” to PORTx.n when DDRx.n = 1. Writing 1 to the PORTx.n makes Q = 1. As a result, a 1 is provided to the pin of the chip. Therefore, any attempt to read the input pin will always get the “HIGH” signal.

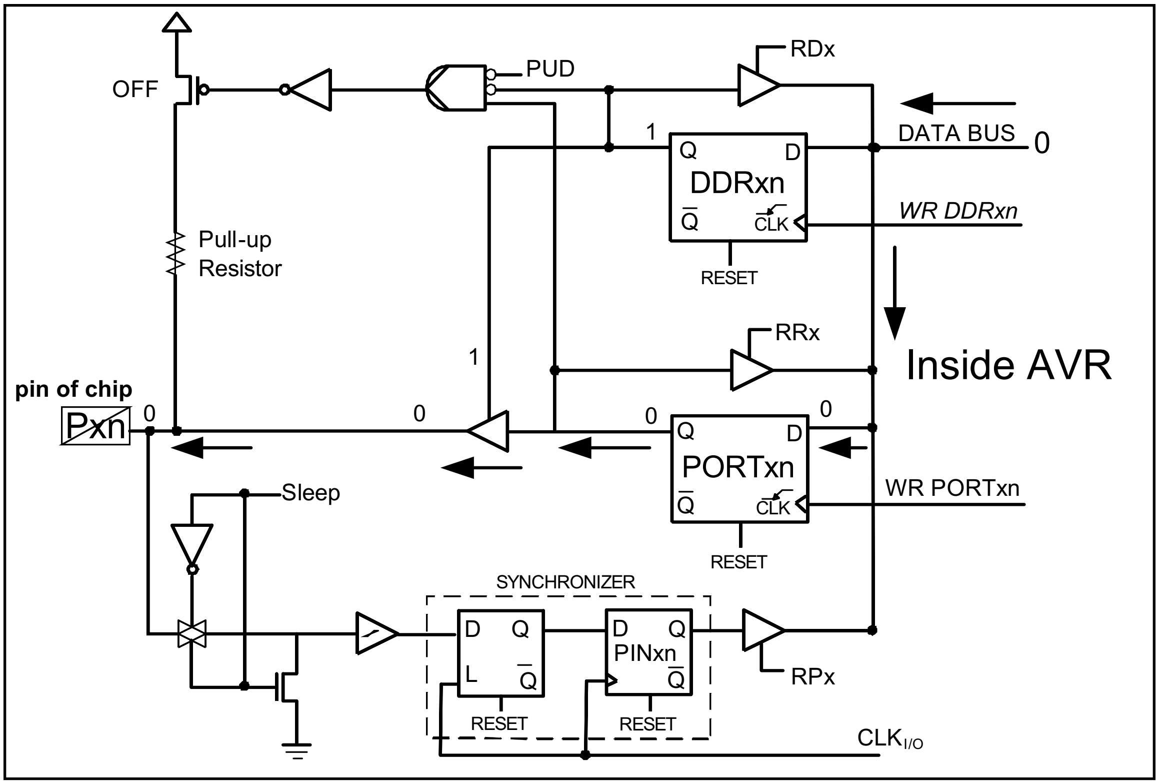


Figure C-12. Outputting (Writing) 0 to a Pin in the AVR

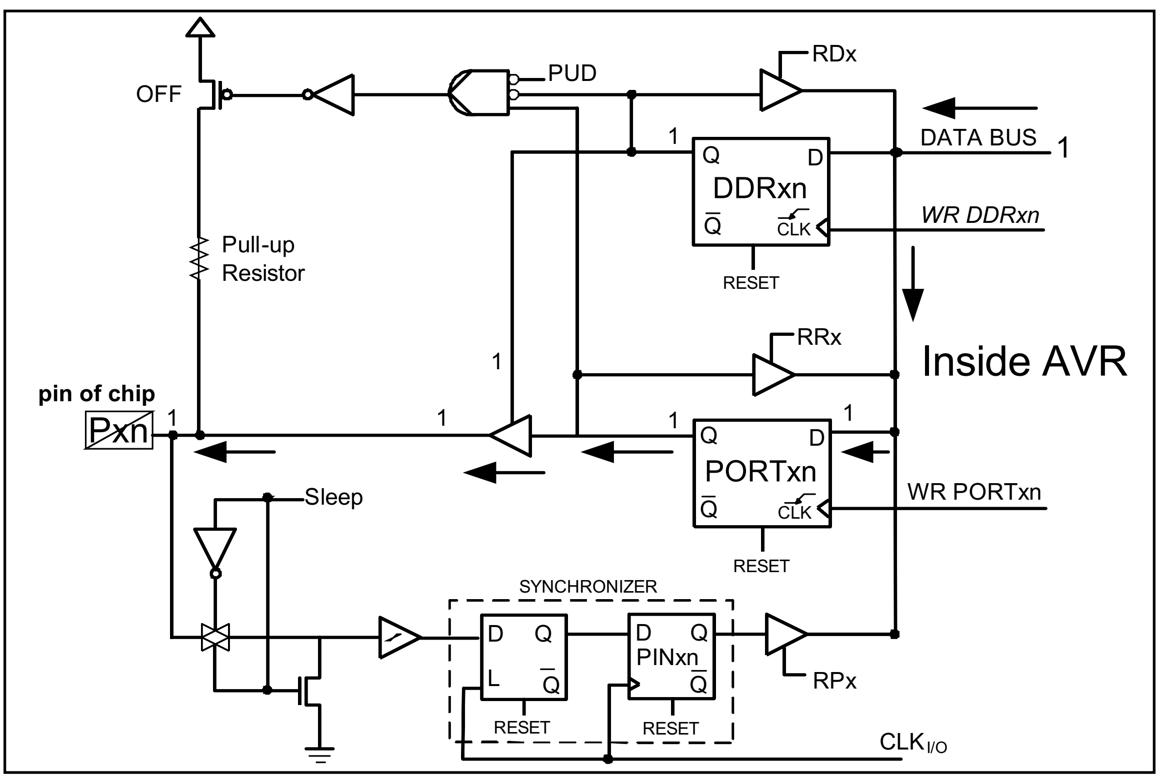


Figure C-13. Outputting (Writing) 1 to a Pin in the AVR

Notice that we should not make an I/O port output while it is externally connected to a voltage; otherwise, we might damage the ports.

For example, see Figure C-14. In this program, the PORTB.3 is mistakenly set as output. When the key is closed, the pin will be directly connected to ground while the AVR is trying to send out high. As a result, the AVR will be damaged when the key is closed. Also, the program will not work properly, as it will always read high while trying to read the pin.

The above points are extremely important and must be emphasized because many people damage their ports and afterwards wonder how it happened. We must also use the right instruction when we want to read the status of an input pin.

```
      SBI   DDRB, 3  ;PB3 as output
;Note: Since PB3 is connected to a
;switch it cannot be configured as
;output
      SBI   PORTB, 3 ;PB3 = high
HERE: SBIC  PINB, 3
      RJMP  HERE  ;stay in the loop
      ...
```

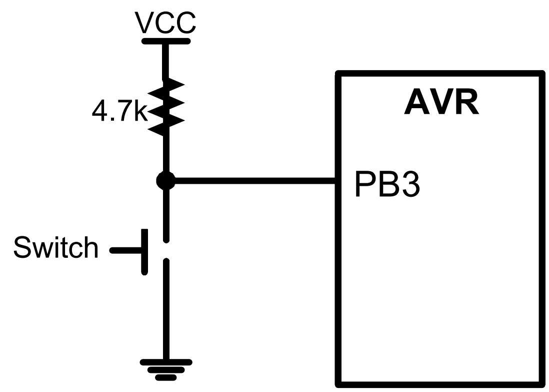


Figure C-14. A Common Mistake, Which Damages I/O Ports

AVR port fan-out

Now that we are familiar with the port structure of the AVR, we need to examine the fan-out for the AVR microcontroller. AVR microcontrollers are all based on CMOS technology. Note, however, that while the core of the AVR microcontroller is CMOS, the circuitry driving its pins is all TTL compatible. That is, the AVR is a CMOS-based product with TTL-compatible pins. Table C-5 provides the I/O characteristics of AVR ports.

Table C-5: Fan-out for AVR Ports

Pin	Fan-out
IOL	20 mA
IOH	−20 mA
IIL	−1 μA
IIH	1 μA

Note: Negative current is defined as current sourced by the pin.

SECTION C.3: SYSTEM DESIGN ISSUES

In addition to fan-out, the other issues related to system design are power dissipation, ground bounce, V_{CC} bounce, crosstalk, and transmission lines. In this section we provide an overview of these topics.

Power dissipation considerations

Power dissipation is a major concern of system designers, especially for laptop and hand-held systems in which batteries provide the power. Power dissipation is a function of frequency and voltage as shown below:

$$Q = CV$$

$$\frac{Q}{T} = \frac{CV}{T}$$

$$\text{since} \quad F = \frac{1}{T} \qquad \text{and} \quad I = \frac{Q}{T}$$

$$I = CVF$$

$$\text{now} \quad P = VI = CV^2F$$

In the above equations, the effects of frequency and V_{CC} voltage should be noted. While the power dissipation goes up linearly with frequency, the impact of the power supply voltage is much more pronounced (squared). See Example C-2.

Example C-2

Compare the power consumption of two microcontroller-based systems. One uses 5 V and the other uses 3 V for V_{CC}.

Solution:
Because P = VI, by substituting I = V/R we have $P = V^2/R$. Assuming that R = 1, we have $P = 5^2 = 25$ W and $P = 3^2 = 9$ W. This results in using 16 W less power, which means power saving of 64% ($16/25 \times 100$) for systems using a 3 V power source.

Dynamic and static currents

Two major types of currents flow through an IC: dynamic and static. A dynamic current is I = CVF. It is a function of the frequency under which the component is working. This means that as the frequency goes up, the dynamic current and power dissipation go up. The static current, also called DC, is the current consumption of the component when it is inactive (not selected). The dynamic current dissipation is much higher than the static current consumption. To reduce power consumption, many microcontrollers, including the AVR, have power-saving modes. In the AVR, the power saving mode is called *sleep mode.* We describe the sleep mode next.

Sleep mode

In sleep mode the clocks of the CPU and some peripheral functions, such as serial ports, interrupts, and timers, are cut off. This brings power consumption down to an absolute minimum, while the contents of RAM and the SFR registers are saved and remain unchanged. The AVR provides six different sleeping modes, which enable you to choose which units will sleep. For more information see the AVR datasheets.

Ground bounce

One of the major issues that designers of high-frequency systems must grapple with is ground bounce. Before we define ground bounce, we will discuss lead inductance of IC pins. There is a certain amount of capacitance, resistance, and inductance associated with each pin of the IC. The size of these elements varies depending on many factors such as length, area, and so on.

The inductance of the pins is commonly referred to as *self-inductance* because there is also what is called *mutual inductance*, as we will show below. Of the three components of capacitor, resistor, and inductor, the property of self-inductance is the one that causes the most problems in high-frequency system design because it can result in ground bounce. Ground bounce occurs when a massive amount of current flows through the ground pin caused by many outputs changing from HIGH to LOW all at the same time. See Figure C-15 (a). The voltage is related to the inductance of the ground lead as follows:

$$V = L\frac{di}{dt}$$

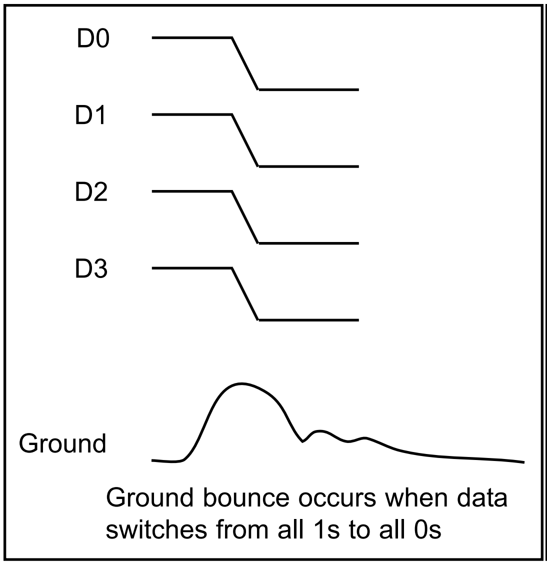
As we increase the system frequency, the rate of dynamic current, di/dt, is also increased, resulting in an increase in the inductance voltage L (di/dt) of the ground pin. Because the LOW state (ground) has a small noise margin, any extra voltage due to the inductance can cause a false signal. To reduce the effect of ground bounce, the following steps must be taken where possible:

1. The V_{CC} and ground pins of the chip must be located in the middle rather than at opposite ends of the IC chip (the 14-pin TTL logic IC uses pins 14 and 7 for ground and V_{CC}). This is exactly what we see in high-performance logic gates such as Texas Instruments' advanced logic AC11000 and ACT11000 families. For example, the ACT11013 is a 14-pin DIP chip in which pin numbers 4 and 11 are used for the ground and V_{CC}, instead of 7 and 14 as in the traditional TTL family. We can also use the SOIC packages instead of DIP.
2. Another solution is to use as many pins for ground and V_{CC} as possible to reduce the lead length. This is exactly why all high-performance microprocessors and logic families use many pins for V_{CC} and ground instead of the traditional single pin for V_{CC} and single pin for GND. For example, in the case of Intel's Pentium processor there are over 50 pins for ground, and another 50 pins for V_{CC}.

The above discussion of ground bounce is also applicable to V_{CC} when a large number of outputs changes from the LOW to the HIGH state; this is referred to as *V_{CC} bounce*. However, the effect of V_{CC} bounce is not as severe as ground bounce because the HIGH ("1") state has a wider noise margin than the LOW ("0") state.

Filtering the transient currents using decoupling capacitors

In the TTL family, the change of the output from LOW to HIGH can cause what is called *transient current*. In a totem-pole output in which the output is LOW, Q4 is on and saturated, whereas Q3 is off. By changing the output from the



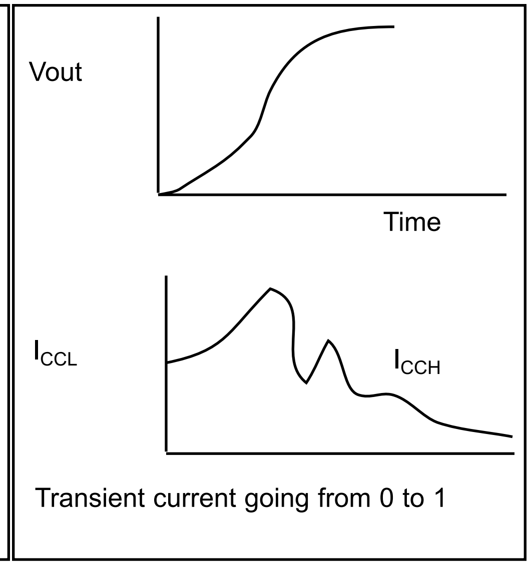


Figure C-15. (a) Ground Bounce **Figure C-15. (b) Transient Current**

LOW to the HIGH state, Q3 turns on and Q4 turns off. This means that there is a time when both transistors are on and drawing current from V_{CC}. The amount of current depends on the R_{ON} values of the two transistors, which in turn depend on the internal parameters of the transistors. The net effect of this, however, is a large amount of current in the form of a spike for the output current, as shown in Figure C-15 (b). To filter the transient current, a 0.01 µF or 0.1 µF ceramic disk capacitor can be placed between the V_{CC} and ground for each TTL IC. The lead for this capacitor, however, should be as small as possible because a long lead results in a large self-inductance, and that results in a spike on the V_{CC} line [V = L (di/dt)]. This spike is called *V_{CC} bounce*. The ceramic capacitor for each IC is referred to as a *decoupling capacitor*. There is also a bulk decoupling capacitor, as described next.

Bulk decoupling capacitor

If many IC chips change state at the same time, the combined currents drawn from the board's V_{CC} power supply can be massive and may cause a fluctuation of V_{CC} on the board where all the ICs are mounted. To eliminate this, a relatively large decoupling tantalum capacitor is placed between the V_{CC} and ground lines. The size and location of this tantalum capacitor vary depending on the number of ICs on the board and the amount of current drawn by each IC, but it is common to have a single 22 µF to 47 µF capacitor for each of the 16 devices, placed between the V_{CC} and ground lines.

Crosstalk

Crosstalk is due to mutual inductance. See Figure C-16. Previously, we discussed self-inductance, which is inherent in a piece of conductor. *Mutual inductance* is caused by two electric lines running parallel to each other. The mutual inductance is a function of l, the length

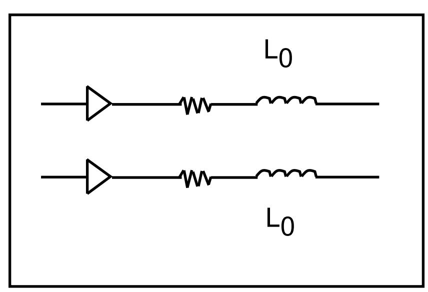


Figure C-16. Crosstalk (EMI)

of two conductors running in parallel; d, the distance between them; and the medium material placed between them. The effect of crosstalk can be reduced by increasing the distance between the parallel or adjacent lines (in printed circuit boards, they will be traces). In many cases, such as printer and disk drive cables, there is a dedicated ground for each signal. Placing ground lines (traces) between signal lines reduces the effect of crosstalk. (This method is used even in some ACT logic families where a V_{CC} and a GND pin are next to each other.) Crosstalk is also called *EMI* (electromagnetic interference). This is in contrast to *ESI* (electrostatic interference), which is caused by capacitive coupling between two adjacent conductors.

Transmission line ringing

The square wave used in digital circuits is in reality made of a single fundamental pulse and many harmonics of various amplitudes. When this signal travels on the line, not all the harmonics respond in the same way to the capacitance, inductance, and resistance of the line. This causes what is called *ringing*, which depends on the thickness and the length of the line driver, among other factors. To reduce the effect of ringing, the line drivers are terminated by putting a resistor at the end of the line. See Figure C-17. There are three major methods of line driver termination: parallel, serial, and Thevenin.

In serial termination, resistors of 30–50 ohms are used to terminate the line. The parallel and Thevenin methods are used in cases where there is a need to match the impedance of the line with the load impedance. This requires a detailed analysis of the signal traces and load impedance, which is beyond the scope of this book. In high-frequency systems, wire traces on the printed circuit board (PCB) behave like transmission lines, causing ringing. The severity of this ringing depends on the speed and the logic family used. Table C-6 provides the trace length, beyond which the traces must be looked at as transmission lines.

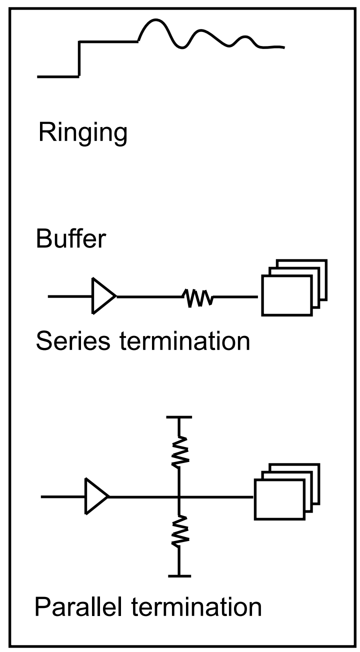


Figure C-17. Reducing Transmission Line Ringing

Table C-6: Line Length Beyond Which Traces Behave Like Transmission Lines

Logic Family	Line Length (in.)
LS	25
S, AS	11
F, ACT	8
AS, ECL	6
FCT, FCTA	5

(Reprinted by permission of Integrated Device Technology, copyright IDT 1991)