Lab 16: Tri-State Busses and Memory

U.C. Davis Physics 116B

Note: We may use a more modern RAM chip. Pinouts, etc. will be provided.

INTRODUCTION

In this lab, you will build a fairly large circuit that will allow the user to write data to any address in a RAM and read that data out sequentially or using a DIP switch to specify the address. Once again, this is a large circuit. Be sure to use good circuit-building technique. That is, build and debug each section of the circuit separately and then connect them all together.

Since this circuit will eventually comprise 6 ICs, I suggest using the chip layout shown in figure 2 so that all ICs fit on the board.

1. TRI-STATE OUTPUT CURRENT

First, we will measure the output current of a tristate device, one buffer of a 74LS241 circuit. Use the circuit in figure 1 to do this. For your lab report, make a table of the current readings for all combinations of switch settings. Also, answer these two questions: What property of a tristate output makes it useful for a bus? Which switch causes zero output current, the "high impedance state"?

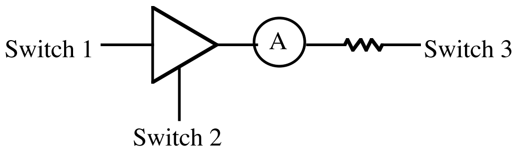


Figure 1: Circuit to measure tristate buffer output current.

2. THE DATA BUS

The goal of this section is to be able to write data to any address in RAM and to read it back. The block diagram in figure 3 shows how to do this. First (and for your lab report), draw a detailed circuit diagram for this circuit and write a description of how you will test it by writing data to several memory addresses and then by reading it back. Then, build the circuit and test it. In your lab report, note whether your circuit passed all your tests or whether any of the tests had to be altered.

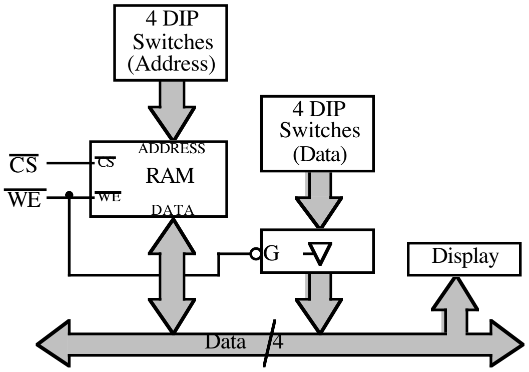


Figure 3: Block diagram for the circuit to read and write to RAM.

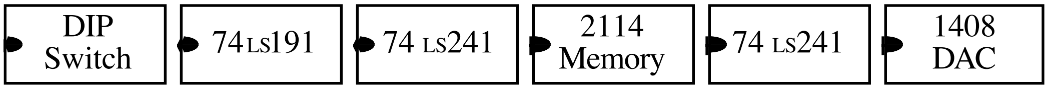


Figure 2: Recommended chip layout for the big memory circuit.

3. The Address Bus

Next, you will alter the circuit so that you can either give the address by hand or allow a counter to sequence through all the memory addresses. The block diagram for this modified circuit is shown in figure 4. As before, draw a detailed circuit diagram before you build the circuit. Write a test procedure which includes writing several 4-bit words to RAM and then reading all 16 using the counter. Then build and test the circuit, noting any difficulties during testing.

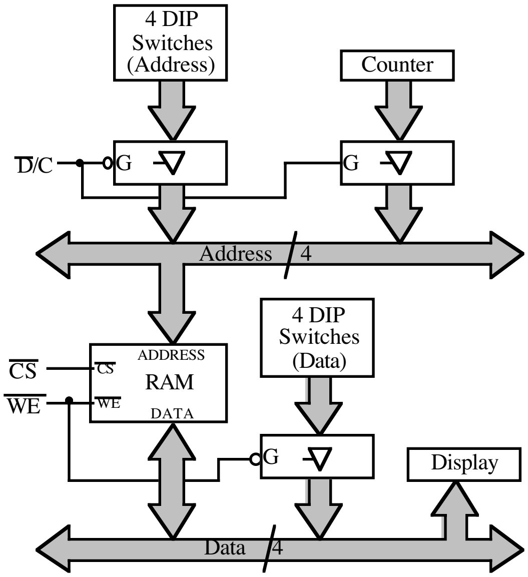


Figure 4: Block diagram for the complete big memory circuit.

4. Digital Waveform Synthesizer

Now use your memory circuit to generate a voltage pattern. Add the 1408 DAC to the data bus as shown in figure 5. (That is, connect the 4 bit data bus to the DAC's 4 most significant input bits.) Using the procedure you wrote above, enter 16 numbers into memory then read them out sequentially into the DAC at a clock speed of about 10 kHz. Look at the output on the oscilloscope. If you are ambitious, try hooking this up to a speaker. Try to enter numbers that make the output look like a sine wave, a triangle wave, and any other shape you'd like to see. If you used the speaker, try to get a particular sound (a violin, trumpet, human voice, *etc*.). For your lab report, describe what is happening in your own words and include your favorite wave shape and the 16 numbers that made it.

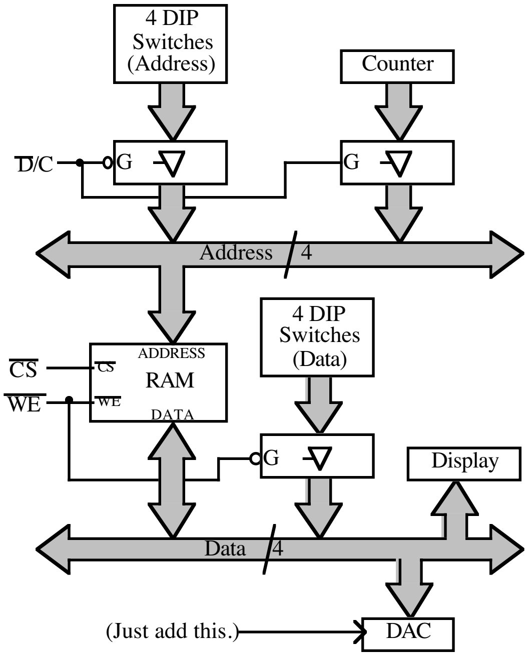


Figure 5: Block diagram for the digital waveform synthesizer.

Discussion: Tri-State Busses and Memory

U.C. Davis Physics 116B

INTRODUCTION

This is an introduction to tristate busses and memory. It is useful for lab 16. Note: we may use a more modern RAM chip (pinouts to be supplied in any case).

1. DATA BUSSES

Busses are fundamental to computer operation. Most of the internal parts of a computer communicate using them. The data bus is named after the regular commuter bus on the streets. The commuter bus will transport many people at once and the same vehicle and roads are used for many origins and destinations. Similarly, a data bus transports many bits of data simultaneously from one digital circuit to another. To be considered a bus, several circuits must be able to send data along the same wires (only one at a time, though), and several circuits must be able to receive data from those wires. The data bus idea is also rather analogous to a group of people talking in a room. Only one person at a time should talk, her voice (the data) fills the entire room (the bus), and many people at a time can listen.

For devices to communicate on a data bus, they must be able to "shut up" and let another device send data over the bus. Our logic gates so far are not very polite listeners; they are always outputting either a 1 or a 0. We need to give them a third state, a "shut up" state, if you will. This third state (hence the name *tri-state*) is called the high impedance (high Z) state. The circuit that allows these three output states is usually called a *tristate buffer* and is exemplified by the 74LS241 chip, an octal tristate buffer. In the first (short) part of lab 15, you will look at the output current of the 74LS241 tristate buffer in each of its three states, 0, 1, and high Z.

2. MEMORY

Recall that any sequential logic element has some sort of memory. A circuit designed to do nothing but store and recall data is called a *memory* circuit. A memory circuit basically contains very many data registers. Recall you built a data register from D flip flops in lab 13. In lab 15, you will use 16 data registers contained in the 2114 memory chip. As you might imagine, 16 four bit data registers would take very many pins on a chip if they were all available at once. So, a memory chip allows you to access only one at a time. Each data register is assigned a number, called its *address*. To select a particular data register, its address is applied to the address inputs of the memory chip. Then, data can be written to or read from that register. Because the addresses can be selected in any order, this type of memory is called *random access memory* or RAM. (In contrast, some types of memory, such as disk drives, must be accessed sequentially.)

Once a given address on a memory chip has been selected, either read or write operations can be done on it using the same data pins. (In our case, we'll use 4 data bits. Most computers today use 16 or 32 data bits.) It may not be obvious how the same pins can be used as both input and output, but this can be done using the tristate bus idea above. The two inputs, write enable $\overline{\text{WE}}$ and chip select $\overline{\text{CS}}$, control the memory data pins according to the following truth table:

$\overline{\text{WE}}$	$\overline{\text{CS}}$	Data Pins are in...
1	1	High Z state.
0	1	High Z state.
1	0	Read mode. Data pins are outputting previously stored data.
0	0	Write mode. Data pins are accepting data, writing to register.

The names "write enable" and "chip select" are descriptive. When $\overline{\text{CS}}$=0 (which means $\overline{\text{CS}}$="on" since $\overline{\text{CS}}$ is active low as the bar indicates), the chip is selected and can read or write. When $\overline{\text{WE}}$=0 (which means $\overline{\text{WE}}$="on), the chip can write data to one of its registers.

Note also that write mode is "dangerous" in that data will be written to memory in write mode, even if it not the intended data or the intended address. The timing diagram in figure 1 shows how to write data to the memory chip safely. Figure 2 shows how to read data from the memory chip. You will be asked to construct similar timing diagrams for the entire memory circuit in lab 15.

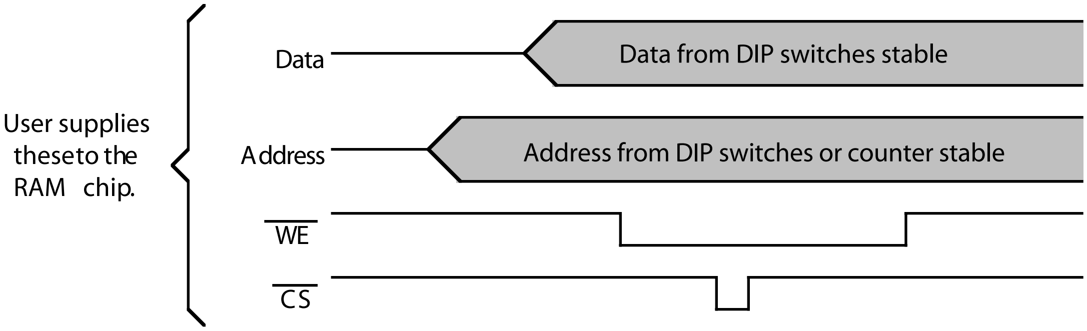

User supplies these to the RAM chip.
Data
Data from DIP switches stable
Address
Address from DIP switches or counter stable
$\overline{WE}$
$\overline{CS}$

Figure 1: WRITE cycle timing diagram.

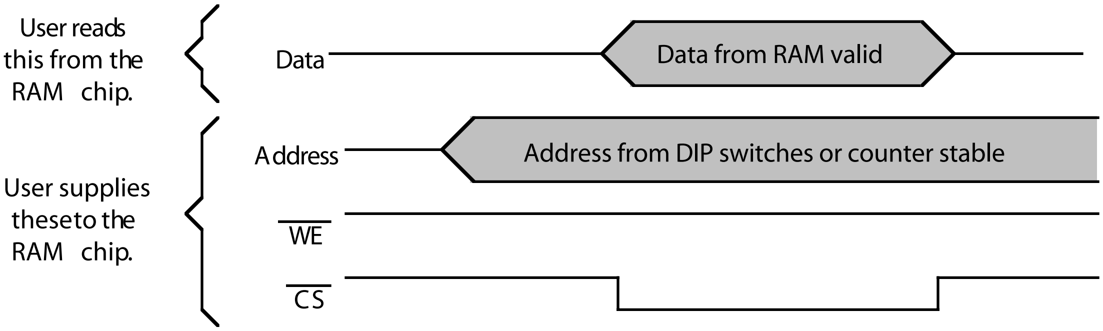

User reads this from the RAM chip.
Data
Data from RAM valid
User supplies these to the RAM chip.
Address
Address from DIP switches or counter stable
$\overline{WE}$
$\overline{CS}$

Figure 2: READ cycle timing diagram.

PRELIMINAR

Notice: This is not a final specification. Som parametric limits are subject to change.

2114
1024 X 4 BIT STATIC RAM

	2114-2	2114-3	2114	2114L3	2114L
Max. Access Time (ns)	200	300	450	300	450
Max. Power Dissipation (mw)	710mw	710mw	710mw	370mw	370mw

- **High Density 18 Pin Package**
- **Identical Cycle and Access Times**
- **Single +5V Supply**
- **No Clock or Timing Strobe Required**
- **Completely Static Memory**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Common Data Input and Output Using Three-State Outputs**
- **Pin-Out Compatible with 3605 and 3625 Bipolar PROMs**

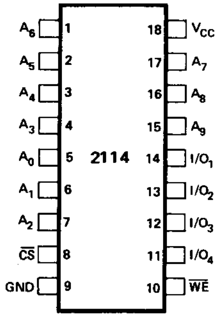
The Intel® 2114 is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using N-channel Silicon-Gate MOS technology. It uses fully DC stable (static) circuitry throughout — in both the array and the decoding — and therefore requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

The 2114 is designed for memory applications where high performance, low cost, large bit storage, and simple interfacing are important design objectives. The 2114 is placed in an 18-pin package for the highest possible density.

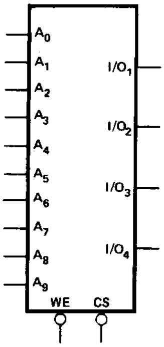
It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate Chip Select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are or-tied.

The 2114 is fabricated with Intel's N-channel Silicon-Gate technology — a technology providing excellent protection against contamination permitting the use of low cost plastic packaging.

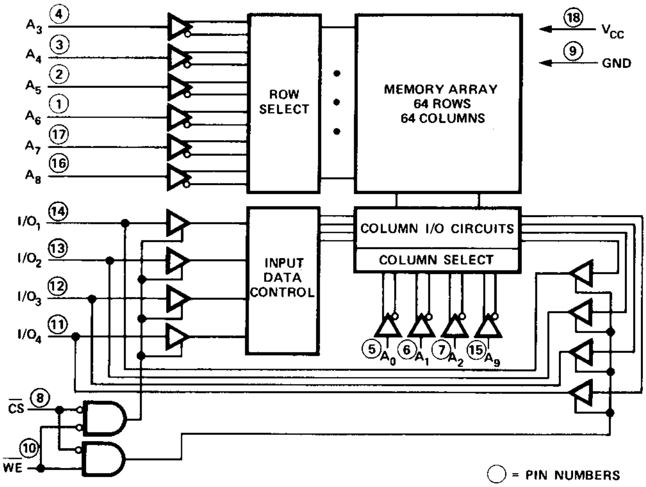
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

A_0–A_9	ADDRESS INPUTS	V_{CC} POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND GROUND
$\overline{CS}$	CHIP SELECT	
I/O_1–I/O_4	DATA INPUT/OUTPUT	

ELIMINAR

Notice: This is not a final specification. S
parametric limits are subject to change.

A.C. CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$, unless otherwise noted.

READ CYCLE [1]

SYMBOL	PARAMETER	2114-2		2114-3, 2114L3		2114, 2114L		UNIT
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{RC}	Read Cycle Time	200		300		450		ns
t_A	Access Time		200		300		450	ns
t_{CO}	Chip Selection to Output Valid		70		100		100	ns
t_{CX}	Chip Selection to Output Active	0		0		0		ns
t_{OTD}	Output 3-state from Deselection	0	40	0	80	0	100	ns
t_{OHA}	Output Hold from Address Change	10		10		10		ns

WRITE CYCLE [2]

SYMBOL	PARAMETER	2114-2		2114-3, 2114L3		2114, 2114L		UNIT
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{WC}	Write Cycle Time	200		300		450		ns
t_W	Write Time	100		150		200		ns
t_{WR}	Write Release Time	20		0		0		ns
t_{OTW}	Output 3-state from Write	0	40	0	80	0	100	ns
t_{DW}	Data to Write Time Overlap	100		150		200		ns
t_{DH}	Data Hold From Write Time	0		0		0		ns

NOTES: 1. A Read occurs during the overlap of a low $\overline{CS}$ and a high $\overline{WE}$.
2. A Write occurs during the overlap of a low $\overline{CS}$ and a low $\overline{WE}$.

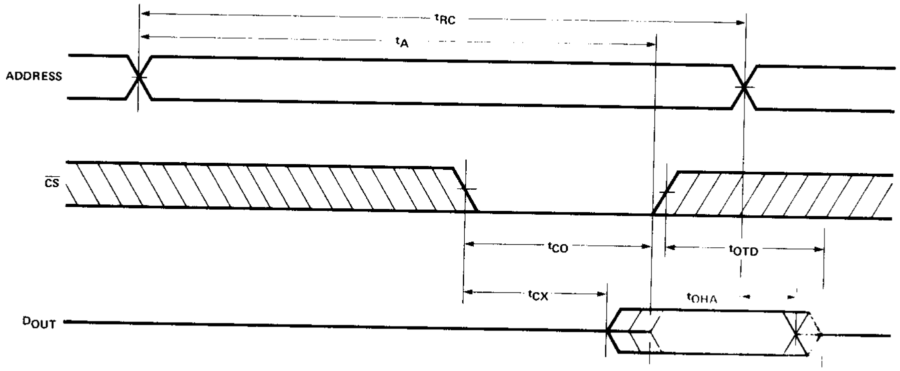
A.C. CONDITIONS OF TEST

Input Pulse Levels	0.8 Volt to 2.4 Volt
Input Rise and Fall Times	10 nsec
Input and Output Timing Levels	1.5 Volts
Output Load	1 TTL Gate and C_L = 50 pF

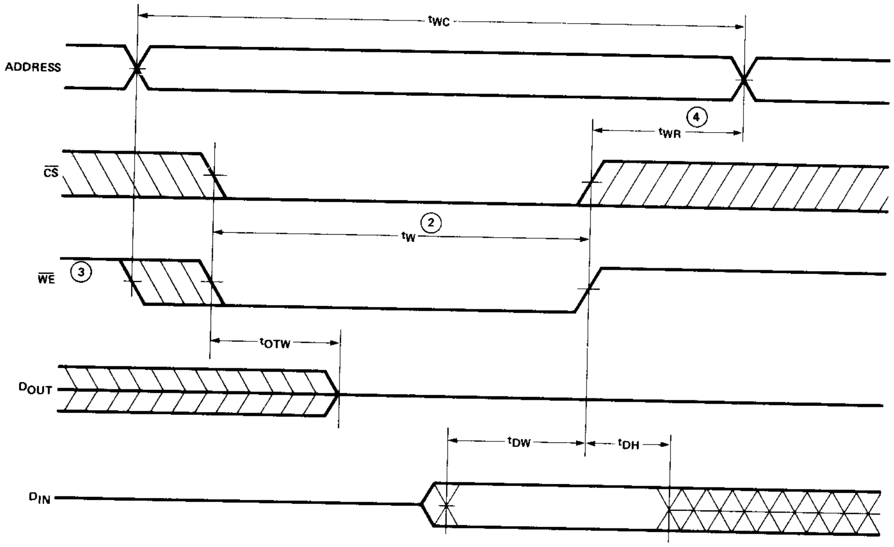
PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

WAVEFORMS

READ CYCLE ①



WRITE CYCLE



NOTES:
① $\overline{WE}$ is high for a Read Cycle.
② t_W is measured from the latter of $\overline{CS}$ or $\overline{WE}$ going low to $\overline{WE}$ going high.
③ $\overline{WE}$ must be high during all address transitions.
④ t_{WR} is referenced to the high transition of $\overline{WE}$.

DATA STORAGE

When $\overline{WE}$ is high, the data input buffers are inhibited to prevent erroneous data from getting into the array. As long as $\overline{WE}$ remains high, the data stored cannot be affected by the address, Chip Select, or data I/O voltage levels and timing transitions. The block diagram also shows data storage cannot be affected by $\overline{WE}$, the addresses, nor the I/O ports as long as $\overline{CS}$ is high. Either $\overline{CS}$ or $\overline{WE}$ by itself — or in conjunction with the other — can prevent extraneous writing due to signal transitions.

Data within the array can only be changed during a Write time — defined as the overlap of $\overline{CS}$ low and $\overline{WE}$ low. To prevent the loss of data, the addresses must be properly established during the entire Write time plus t_{WR}.

Internal delays on the 2114 are established such that address decoding propagates ahead of data inputs (keyed by the Write time). Therefore, it is permissable to establish the addresses coincident to the selection of a Write time, but no later. If the Write time precedes the addresses, the data in the previously addressed locations, or some other location, may be inadvertently changed.

While it is important that the addresses remain stable for the entire Write cycle, the data inputs are not required to remain stable. Appropriate voltage levels will be written into the cells as long as the data is stable for t_{DW} at the end of the Write time.