Lab 9a Memory Interface and Memory Testing

This laboratory assignment accompanies the book, Embedded Microcomputer Systems: Real Time Interfacing, by Jonathan W. Valvano, published by Brooks-Cole, copyright © 2000.

Goals
- Interface a static RAM chip to the 6812,
- Implement a memory test program.

Review
- Valvano Chapter 9 on interfacing to the MC68HC812A4,
- Data sheets on the 128K by 8-bit static RAM (same as K6T1008C2E).

Starter files
- **`MEMTEST`** project

Background

This experiment considers the testing of a 128K by 8-bit RAM system. It will serve to demonstrate the general problems involved with static RAM memories. The memory system to be designed in this experiment will involve a K6T1008C2E RAM chip (128K addresses by 8 bits). This chip can store a total of 128K bytes and each byte has a unique address. This is called byte addressable. The memory interface design provides for the flexibility of allowing the RAM chip select (**`CS1`**) to be connected to either the 6812 **`CSD`** or the 6812 **`CSP0`**. To use the chip to store data the **`CSD`** chip select will be used. On the other hand, to use the chip to store programs the **`CSP0`** chip select will be used. The **`CSD`** can be used to place it from $0000 to $7FFF, while the **`CSP0`** can be used to place it from $8000 to $FFFF. 32K of the available 128K will be easily accessed using the expanded narrow mode of the 6812. Extended addressing mode will be required to access rest of your external RAM.

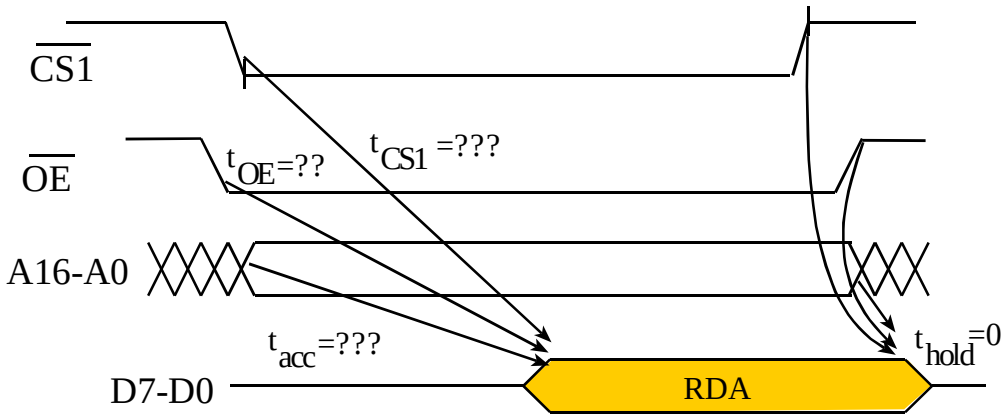

Figure 9.1. Read timing for the 128K RAM chip.

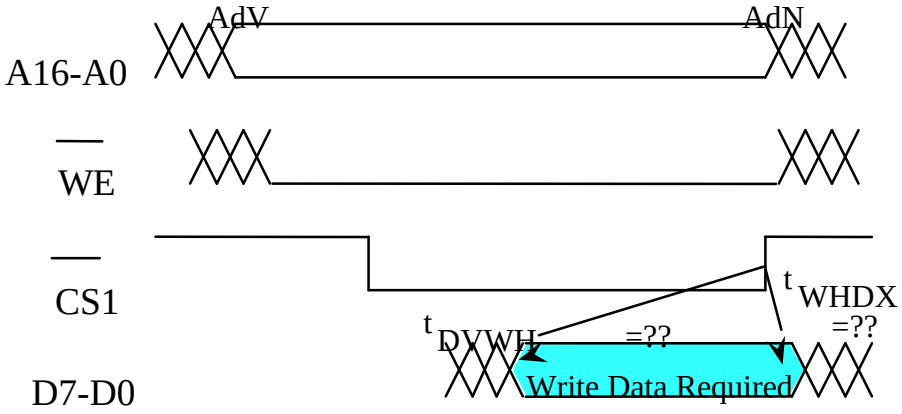

Figure 9.2. Write timing controlled by $\overline{\text{CS1}}$.

Preparation (do this before your lab period)

1: Draw a circuit diagram for the PC board created by your layout team. You should have completed this circuit diagram as part of the preparation for a previous lab, but make sure you have the up-to-date documentation. Assemble the components required to complete the interface, and solder them to the PC board.

2: Choose the correct number of cycle stretches so that Read Data Available overlaps Read Data Required and that Write Data Available overlaps Write Data Required. By using the built-in chip select, **CSD**, you will be synchronizing the memory control signals with the **E** clock. Draw the combined READ timing diagram and draw the combined WRITE timing diagram, as described in Chapter 9 of the book.

3: Next, write a program to test dynamic RAMs. You will run this program to test the your static RAM. Even though it is a static RAM, it should illustrate the differences between static and dynamic memory testing. You will be testing whether the DRAM can retain information over a few memory refresh cycles. In particular, you will write a known sequence of data values into the memory, wait 2 seconds, and verify if that sequence is still present.

4: Next, write an exhaustive test program to the entire 128K of RAM using extended address mode. You should test the ability to write every possible data into every possible address. The following list includes typical types of memory faults:

address pin is stuck high,
address pin is stuck low,
two address pins stuck together,
data pin is stuck high,
data pin is stuck low,
two data pins stuck together.

In this test you will search for very specific types of memory faults. The TA has extra 50-pin sockets with 1, 2, 3, or 4 pins removed. For example, if the TA pulls the pin associated with A7, then the 6812 A7 will not be connected to the memory A7. Neither the 6812 or the memory will be permanently damaged if this faulty socket is placed between the Adapt812 board and the memory board. If address pin A7 is disconnected, then your software should report the error "**Pin A7 is faulty**". If address pins A15, A1 and data pin D5 are all disconnected, then your software should report all errors "**Pin A15 is faulty, Pin A1 is faulty, Pin D5 is faulty**".

Procedure (do this during your lab period)

1: First test your interface with the following program.

```
#define memory *(unsigned char *)(0x7000)
void main(void){
  COPCTL = 0x00; // disable COP
  DDRT = 0xFF;   // PortT is output
  MODE = 0x33;   // special exp narrow
  MODE = 0x33;   // special exp narrow
  PEAR = 0x2C;
  CSCTL0 = 0x10;  // enable CSD
  CSCTL1 = 0;     // $7000 to $7FFF
  CSSTR0 = 0x3F;  // stretch 3 E clocks
  while(1){
    memory = 0xC3;  // write data to external RAM
    PORTT = memory; // read from external RAM
  }
}
```

The jumpers for **MODA** and **MODB** on the Adapt812 will be left at their usual settings (0,0). To download and run a program in special expanded narrow mode, follow these steps:

1) Within Metrowerks edit, compile the program using the regular single chip segmentation
   - RAM at 0x0800-0x0BFF (globals start at 0x0800, SP initialized to 0x0C00)
   - EEPROM at 0xF000-0xFFFF
2) Power up Adapt812/BDM, and start BDM software
3) Execute **sync reset** until 6812 is halted (**r** command works, **status** returns **C0**)
4) Verify the processor is in special mode
   db 000b should give **1B** for *special single chip mode*
   (your program will change it to **33** *special expanded narrow mode*)
5) Verify the EEPROM is at $F000-$FFFF

db 0012 should be F1

6) Download your S19 record in the usual way

7) Start your program from the debugger (do not hit the reset button[1])

g f029 (look at the second to last line of your S19 record to see where it starts)

While running this program, the memory interface signals can be observed on the oscilloscope using the dual channel capability of the scope. In the tight loop, a pulse should be received from the **CS1** line during accesses to the memory. Observe **CS1** with the following signals in a pair-wise fashion:

- **CS1** and **E**
- **CS1** and **A14**
- **CS1** and **R/W**
- **CS1** and **D0**

Trigger on the falling edge of **CS1**. Using the dual channel capability, verify the proper relation between **E**, **A14**, **R/W** and **CS1**. In this way you should be able to verify the timing diagram created as part of the lab preparation. Draw the read and write timing diagrams as actually measured. Include both data available and data required. Which one did you measure and which one did you derive? Verify that all the 6812 and RAM timing restrictions are met. Repeat for several read and write cycles.

2: Next, run the **MEMTEST** project to verify that the RAM system is operating correctly.

3: Run your two memory test programs. To create errors purposefully run the interface with too few cycle stretches.

4: Change the interface so that it uses **CSP0**. The jumpers **MODA** and **MODB** are on the Adapt board (change **MODA** from off=0 to on=1). Leave the **MODB** at 0. For the Rev1 boards, to make **MODA**=1 you connect the JB2 jumper, and place the SW3 switch to EXP. For the Rev2 boards, to make **MODA**=1 you connect the JB1 jumper to the 1 position (not the 0 position). Remember to put **MODA** back before you turn the board in. Within the compiler, change the start of ROM from F000 to 8000. You now have 32K of program space. Try downloading and running a program out of your external RAM.

Deliverables (exact components of the lab report)

A) Objectives (1/2 page maximum)

B) Hardware Design

Detailed circuit diagram of the memory interface (preparation 1)

C) Software Design (no software printout in the report)

Briefly explain how your software works (1/2 page maximum)

D) Measurement Data

Draw the theoretical read and write timing diagrams (preparation 2)

Draw the measured read and write timing diagrams (procedure 1)

E) Analysis and Discussion (1 page maximum)

Checkout (show this to the TA)

You should run your two memory test programs. The TA has a plug that can be placed between your Adapt board and your RAM board that creates memory faults. You should be able to demonstrate how your interface can be used for extra program space using **CSP0**.

Your software files will be copied onto the TA's zip drive during checkout.

Hints:

1. If you need to download, and untether the system from the PC power, then power the Adapt board with batteries (6V into the power connector) or 5V directly into the 6812.
2. If you need help soldering, please ask the TA or one of the 2nd floor technicians.
3. The read access time (delay from address available to the start of read data available) can be determined by the number after the dash on the chip. For example, –7 means 70 ns, –8 means 80 ns, –1 means 100 ns, and –12 means 120 ns. The read access time is also the delay from the fall of CS1 to the start of read data available.
4. Don't try to create stuck high, stuck low, stuck together faults because it may damage the 6812 or memory.

[1] Hitting the reset button will place the processor in normal single chip mode. You need special mode.