Experiment No. 5
MEMORY DESIGN USING STATIC RANDOM ACCESS MEMORY (RAM)
ECE 441

Peter Chinetti

October 30, 2013

Date Performed:	October 10,17, & 24, 2013
Partners:	Zelin Wu
Instructor:	Professor Saniie

1 Introduction

1.1 Purpose

The purpose of this experiment is to familiarize the user with the following devices and concepts:

- Static Random Access Memory (SRAM)
- Open Collector Devices
- the SANPER-1 ELUS Block Select Lines

1.2 Background

1.3 MC68000 Asynchronous Bus Interface

The MC68000 has a 16-bit data bus, and byte accesses of memory locations are permitted from either an even or an odd address.

1.4 Open Collector Devices

A standard TTL type device has two output transistors that form the output driver for the logic gate. One transistor sits about the other one. That is, the emitter of the top transistor is tied to the collector of the bottom transistor. The top transistor function acts as an active pullup for the bottom transistor. For example, a 74LS04 logic device consists of six inverters. Each inverter consists

of two transistors that form the output driver. One of the transistors is used as an active pullup for the other transistor. On the other hand, a TTL logic device with open collector outputs has only one transistor acting as the output driver. The collector of the transistor is left floating or open. Hence the name open collector output. The IC manufacturer expects the design engineer to add whatever external pullup mechanism he/she deems appropriate for the circuit to function properly.
Since a standard TTL type logic gate has a pullup mechanism built in, these outputs cannot be tied together due to possible contention problems. For example, if one of the device is trying to drive its output low, while another device is trying to drive its output high, one of the two devices will win the contention battle, while the other device will have its output transistor destroyed.
On the other hand, since devices with open collector outputs do not have a built in pullup mechanism, several open collector outputs can be tied together without damaging any of the output drivers. When all the outputs are tied together, it forms a single bus line. A resistor can then be used as an external pullup mechanism for this bus line.
A good application for devices with open collector outputs is found in microprocessor systems. If several peripheral devices exist in the system, the Interrupt Request Lines (*IRQ) or each of these devices can be connected together and then connected to the *IRQ input of the microprocessor. A resistor is connected between this *IRQ bus line and the logic supply voltage (typically +5V DC). Thus one of the peripheral devices can assert the *IRQ line without damaging the other devices. Note this technique can only be employed if the device is equipped with an open collector output on its *IRQ line.

1.5 The SANPER-1 ELU'S Block Select Lines

There are four Block Select lines (*Q5, *Q6, *Q7, *Q8) provided on the System Expansion Board which can be used for decoding and selecting external devices. Each of the Block Select lines was obtained by decoding address A16 through A23 and Address Strobe (*AS). Thus one of these Block Select lines can be used as an input to any additional external address decoding circuitry.
The SANPER-1 ELU Memory Map indicates the address ranges associated with each of the Block Select lines.

2 Lab Procedure and Equipment List

2.1 Equipment

- SANPER System
- Computer with TUTOR software

2.2 Procedure

Design and implement memory, then test with a memory test program.

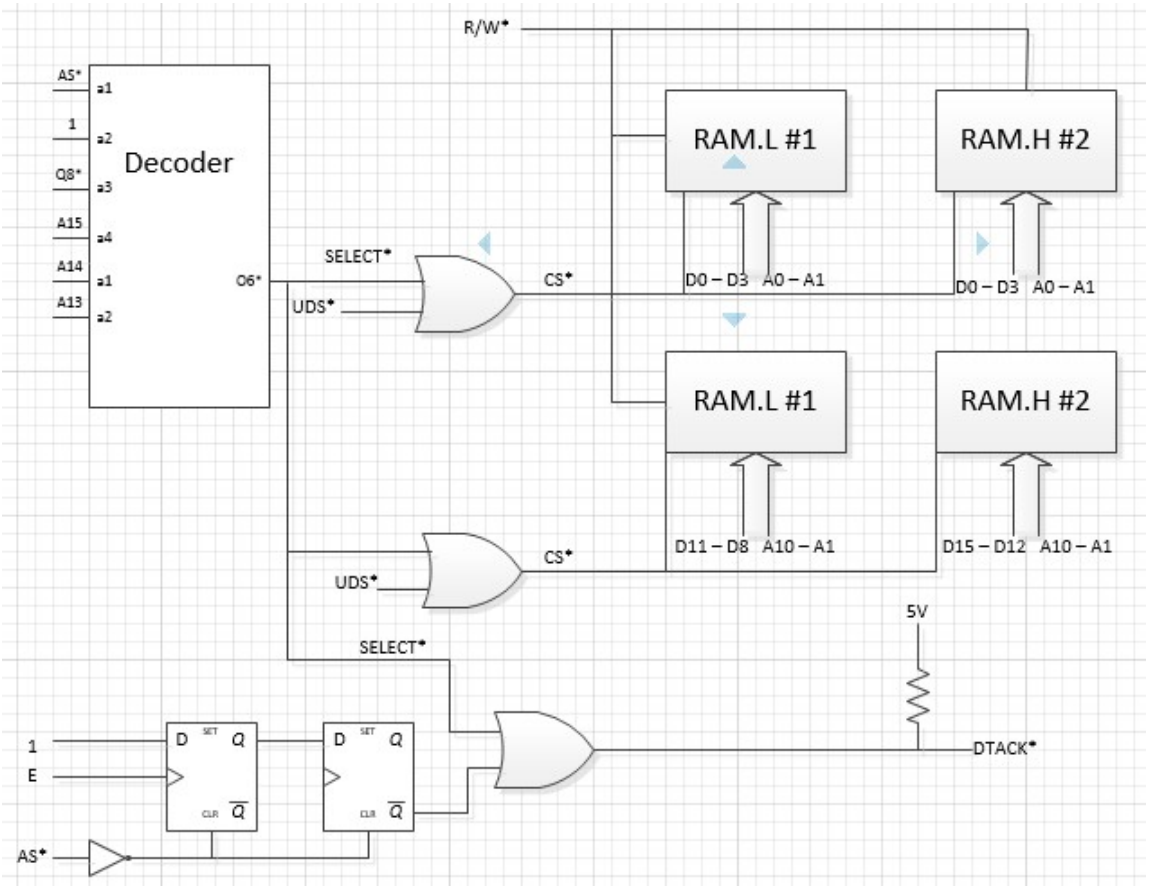
3 Results, Analysis and Discussion

3.1 Memory Test Program

```
                    ORG $900
MSGPAS              DC.B    'MEMORY TEST PASSED'
MSGENP              DS.B    1
START:              ORG $1000
                    MOVE.L  #$082000,A5     ;Limits of memory are:
                    MOVE.L  #$087FFF,A6     ;82000 87FFF
LOOP:               MOVE.B  #$AA,(A5)+      ;Write out 0xAA to memory
                            CMPA.L  A5,A6   ;Check if we are done with range
                    BNE     LOOP            ;Loop back
                    MOVE.L  #$082000,A5     ;Go back to start
CHECK_AA:           CMP.B   #$AA,(A5)       ;Make sure AA got written out
                    BNE     FAIL_AA         ;If it didn't fail
                    MOVE.B  #$55,(A5)+      ;Write 0x55 out
                    CMP.L   A5,A6           ;until done with range
                    BNE     CHECK_AA        ;loop back
                    MOVE.L  #$082000,A5     ;Go back to start
CHECK_55:           CMP.B   #$55,(A5)       ;Check 0x55 can be read
                    BNE     FAIL_55         ;fail
                    MOVE.B  #$AA,(A5)+      ;write AA out
                    CMPA.L  A5,A6           ;until done with range
                    BNE     CHECK_55        ;loop
                    LEA     MSGPAS,A5       ;load string addr into A5
                    LEA     MSGENP,A6       ;load A6
                    JMP     OUTPUT          ;jump to success
FAIL_AA:            MOVE.L  #$2000,A6       ;load A6
                    MOVE.L  A5,D0
                    MOVE.B  #231,D7         ;PNT6HX
                    TRAP    #14
                    MOVE.B  #$20,(A6)+      ;space
                    MOVE.B  #$41,(A6)+      ;A
                    MOVE.B  #$41,(A6)+      ;A
                    MOVE.B  #$20,(A6)+      ;space
                    MOVE.B  (A5),D0
                    MOVE.B  #233,D7         ;PNT2HX
                    TRAP    #14
                    MOVE.L  #$2000,A5       ;set start of output
                    JMP     OUTPUT          ;jump to output w/ string 'AA'
```

```
FAIL_55:    MOVE.L   #$2000,A6
            MOVE.L   A5,D0
            MOVE.B   #231,D7          ;PNT6HX
            TRAP     #14
            MOVE.B   #$20,(A6)+       ;space
            MOVE.B   #$35,(A6)+       ;5
            MOVE.B   #$35,(A6)+       ;5
            MOVE.B   #$20,(A6)+       ;space
            MOVE.B   (A5),D0
            MOVE.B   #233,D7
            TRAP     #14
            MOVE.L   #$2000,A5       ;set start
OUTPUT:     MOVE.B   #243,D7          ;OUTPUT
            TRAP     #14
            MOVE.B   #228,D7          ;TUTOR
            TRAP     #14
            END START
```

3.2 Diagram



3.3 Memory Location

Memory was placed at the $082000 address, beacuse it had a simple address to decode.

3.4 Memory Test Design

The memory test program tests that 0xAA (10101010b) can be written and read, then that 0x55 (01010101b) can be written and read. Another useful pattern to write would be 0x00FF and 0xFF00 so upper and lower byte order can be checked.

3.5 Why SRAM

SRAM is easy to use, as it does not requre refreshes like DRAM does, and the high density possible with DRAM is not required for a 1K block.

4 Conclusions

This experiment was accomplished. A memory bank was built and tested.